Imperial College London

INSTITUTE OF GLOBAL HEALTH INNOVATION

3D Vision for Improved Scene Understanding in Robotic Surgery

Author Samyakh Tukra
Submitted May 5th 2023

Supervisors Dr. S. Giannarou[1,2]

[1]Department of Surgery & Cancer, Imperial College London, UK.
[2]Hamlyn Centre of Surgical Innovation, Imperial College London, UK.

A thesis submitted in fulfilment of the requirements for the degree of *Philosophiae Doctor* in Imperial College London

Hamlyn Centre of Surgical Innovation, Department of Surgery & Cancer,
Imperial College London

Hamlyn Centre of Surgical Innovation,
Department of Surgery and Cancer,
Imperial College London,
180 Queens Gate,
South Kensington,
SW7 2AZ

Samyakh Tukra © 2023

One should never impose limits on their capabilities.
Therefore, despite the odds, relentlessly push forward,
persevere, and refuse to settle for mediocrity!

Samyakh Tukra (myself)

These 4.5 years have shaped the future years of my life. It has been an educational journey as a whole.

This thesis is dedicated to my paternal and maternal grandfathers Mr. Shuban Lal Tukra and Mr. Pran Nath Koul respectively. Additionally, my parents Mr Sudhir and Ms Rekha Tukra, to whom I owe all my successes and joyous moments in life. They are my ultimate superheroes.

Declaration

This thesis presents original ideas that were developed through collaboration between the author and their supervisors. The author affirms that to the best of their knowledge, the work is original and proper attribution has been provided where necessary. The contributions made by others have been acknowledged through co-authorship of the relevant papers described in this thesis.

The work in chapter 3 was published in the IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) in July 2020 and has obtained a patent for being a radical new surgical vision technology to improve surgeon's visibility during operation. The work in chapter 4 was published in the journal of Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization in 2021. It also won the best paper award in the 15th Augmented Environments for Computer Assisted Interventions (AE-CAI) MICCAI 2021 workshop. The work from Chapter 5 was published in the conference proceedings of the 25th international conference on Medical Image Computing and Computer Assisted Intervention (MICCAI) in 2022. The work from Chapter 5 was further extended in chapter 6 and was submitted to the conference proceedings of the 26th international conference on Medical Image Computing and Computer Assisted Intervention (MICCAI) in 2023. The work in chapter 7 will be submitted to the journal of IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) in April 2023. The research presented in this thesis has been showcased as an abstract or in poster formats at numerous conferences and events. Furthermore the work shown in this thesis has also been exhibited in the book of Artificial Intelligence in Medicine (Springer Nature).

The author is the exclusive writer of this document, unless stated otherwise. All tables and figures were produced by the author, including the formatting and styling of the document. This thesis has been submitted in fulfilment of the requirements for the degree of Philosophiae Doctor at Imperial College London only. It has not been submitted for any other degree or professional qualification.

The formatting template for this PhD thesis, is adapted from the open-source repository [178] which can be found: https://github.com/mlnotebook.

Samyakh Tukra,

April 2023

Copyright

I declare that this thesis is my own research work except where explicitly stated, and that I have acknowledged the work of others.

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-Non Commercial 4.0 International Licence (CC BY-NC). Under this licence, you may copy and redistribute the material in any medium or format. You may also create and distribute modified versions of the work. This is on the condition that: you credit the author and do not use it, or any derivative works, for a commercial purpose. When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes. Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

Acknowledgements

I would like to express my sincerest gratitude to the many individuals whom I have had the privilege of meeting before and during my doctoral studies.

I would like to thank my supervisor Dr Stamatia Giannarou, for giving me the opportunity to work on this exceptional project, for guiding me and believing in my capability to improve, evolve and deliver. Her mentorship throughout the course of my thesis, have been instrumental in shaping my academic growth. I'd also like to thank Prof. Anil Bharath, who's supported me throughout my time at Imperial College and expanded my knowledge in AI. I'm grateful to have worked with him during my masters, which catalysed my interest in research.

My heartfelt thanks to Ms. Meng Gao for her constant support, affection, and care during this challenging journey and beyond. Her encouragement and understanding have been crucial in my perseverance, and I'm deeply grateful for her presence in my life and for her impact on my personal and academic pursuits. I aspire to reciprocate her influence someday.

I am grateful to Mr. Callum Adamson for his invaluable mentorship in organisation, professional communication, and interpersonal skills, which have shaped my professional and entrepreneurial development. My sincere thanks to Ms. Ashley Duqué Kienzle for her guidance in product design, delivery, and translation into practical applications, which enriched my understanding of user needs and fostered my growth in this field. I'd also like to thank Dr. Marta Kedrzycki for her exceptional guidance in personal and professional organisation, as well as business management, empowering me to realise my research's full potential and real-world impact. Furthermore, I'd like to thank all my lab mates: Joao Cartucho, Chi Xu, Haozheng Xu, David Shapira, Alistair Weld, Joseph Davids and Alfie Roddan and my friend Oliver Stark. Their camaraderie, collaborative spirit, and shared passion for discovery have enriched my research journey and fostered an inspiring and supportive environment.

Finally, I would like to express my gratitude to my parents Sudhir and Rekha Tukra, and my extended family, who, despite facing tremendous adversity, provided me with the foundation for my life journey. Their resilience, courage, and unwavering love have been the bedrock of my development and successes. Their selflessness has been an inspiration for my accomplishments. I will forever be grateful for their sacrifices and the powerful legacy they have instilled in me.

This thesis was supported by the Royal Society (UF140290 and RGF\EA\180084) and the NIHR Imperial Biomedical Research Centre (BRC). All work was completed at the Hamlyn Centre for Robotic Surgery in the Department of Surgery and Cancer at Imperial College London.

Abstract

Robotics and Mixed Reality (MR) visualisation are revolutionising Minimally Invasive Surgery (MIS), enhancing precision and efficiency. These technologies promise better patient outcomes, shorter recovery times, and reduced complications. A key application is guiding tumor resection, where MR can overlay tissue characteristics from intra- and pre-operative sensors as dynamic holograms on anatomical structures, enhancing the surgeon's view. However, its clinical adoption has been slow due to challenges with real-time autonomy, robustness against moving objects, and soft-tissue deformation. Addressing these issues requires accurate, occlusion-aware, geometrically and temporally consistent depth estimation in complex, dynamic environments. In this thesis, I tackle four key challenges: (1) occlusions causing ambiguous depth, (2) sub-millimeter depth accuracy, (3) dynamic motion, and (4) temporal inconsistency. I propose Deep Learning (DL) solutions focused on self-supervised learning and generative modeling, circumventing the need for scarce ground truth data in surgery. First, I developed a bespoke video generative model that removes occlusions from a video and generates new frames with high spatial fidelity. To enhance depth estimation accuracy, I designed a multi-scale perceptual loss function that achieves State-of-the-Art performance even with a randomly connected neural network, surpassing many stereo models. I further adapted this approach to a stereo depth estimation model that achieves sub-millimeter accuracy without surgical training data. For dynamic motion and temporal consistency, I introduced a novel method that trains two models: (i) a Multi Layer Perceptron (MLP) for 3D scene-flow prediction and (ii) a video depth estimation model that learns temporal dynamics via Diffusion. These models optimise each other to ensure temporal consistency in depth estimation, even in highly dynamic environments. Finally, I present an Machine Learning Operations (MLOps) pipeline for deploying DL models on MR hardware, such as the Microsoft HoloLens, via the Cloud for clinical use.

Contents

1 Introduction **23**

1.1 Overview . . . 24
1.2 Clinical Needs and Assumptions . . . 27
1.3 Contributions . . . 28
1.4 Publications . . . 30
1.4.1 Related Publications . . . 30
1.4.2 Currently In Review . . . 30
1.4.3 Other Publications . . . 30

2 Background **31**

2.1 Deep Learning . . . 32
2.1.1 Machine Learning Preliminaries . . . 32
2.1.2 Feed Forward Neural Networks . . . 34
2.1.3 Optimisation . . . 36
2.1.4 Convolutional Neural Networks . . . 36
2.1.5 Recurrent Neural Networks . . . 39
2.1.6 Transformers . . . 39
2.2 Generative Modelling . . . 42
2.2.1 Perceptual Learning . . . 42
2.2.2 Generative Adversarial Networks . . . 43
2.2.3 Diffusion . . . 45
2.3 Self-Supervised Representation Learning . . . 47
2.3.1 Contrastive Representation Learning . . . 48
2.3.2 Masked Image Modelling . . . 49
2.4 3D Computer Vision . . . 50
2.4.1 Structure from Motion . . . 50
2.4.2 Stereo Depth . . . 51
2.4.3 Multi-view Stereo . . . 51
2.5 Clinical Research Review . . . 53
2.6 Challenges of Deploying Deep Learning . . . 55

3 See-Through Vision with Unsupervised Scene Occlusion Reconstruction in Videos **57**

3.1 Overview . . . 58

3.2 Introduction . . . 59
3.2.1 Contributions . . . 60
3.3 Related Works . . . 61
3.4 Methods . . . 64
3.4.1 Model . . . 64
3.4.1.1 Generator . . . 64
3.4.1.2 Discriminators D . . . 67
3.4.2 Loss function . . . 67
3.4.3 Implementation Details . . . 70
3.5 Results & Discussion . . . 72
3.5.1 Validation Datasets and Performance Metrices . . . 72
3.5.2 Performance Evaluation on MIS Data . . . 72
3.5.3 Performance Evaluation on Natural Scenes . . . 77
3.5.4 Discussion . . . 79
3.6 Conclusions . . . 80
3.6.1 Learnings & Future Work . . . 80

4 Randomly connected neural networks for self-supervised monocular depth estimation 82
4.1 Overview . . . 83
4.2 Introduction . . . 84
4.2.1 Contributions . . . 85
4.3 Related Works . . . 86
4.4 Methods . . . 88
4.4.1 Monocular Depth Estimation Model . . . 88
4.4.2 Loss Function Formulation . . . 91
4.4.3 Implementation Details . . . 93
4.5 Results & Discussion . . . 95
4.5.1 Performance Evaluation . . . 95
4.5.2 MIS Evaluation . . . 95
4.5.3 Natural Scenes Evaluation . . . 99
4.6 Conclusions . . . 101
4.6.1 Learnings & Future Work . . . 101

5 Improving Stereo Depth Estimation via Self-Supervised Contrastive Representation Learning 103
5.1 Overview . . . 104
5.2 Introduction . . . 105
5.2.1 Contributions . . . 106
5.3 Related Works . . . 107

5.3.1 Supervised Stereo Depth Estimation . . . 107
5.3.2 Self-Supervised Stereo Depth Estimation . . . 107
5.4 Methods . . . 108
5.4.1 Stereo Contrastive Representation Learning . . . 108
5.4.2 Decoder Architecture . . . 110
5.4.3 Self-Supervised Loss . . . 111
5.4.4 Implementation Details . . . 112
5.5 Results & Discussion . . . 113
5.6 Conclusions . . . 116
5.6.1 Learnings & Future Work . . . 116

6 Generalisable Stereo Depth Estimation via Masked Image Modelling 117
6.1 Overview . . . 118
6.2 Introduction . . . 119
6.2.1 Contributions . . . 120
6.3 Methods . . . 121
6.3.1 Stereo Masked Image Modelling . . . 121
6.3.2 Supervised Downstream Fine-tuning . . . 123
6.3.3 Implementation Details . . . 124
6.4 Results & Discussion . . . 125
6.5 Conclusions . . . 128
6.5.1 Learnings & Future Work . . . 128

7 Boosting Depth with Temporal-Geometric Consistency 130
7.1 Overview . . . 131
7.2 Introduction . . . 132
7.2.1 Contributions . . . 133
7.3 Related Works . . . 135
7.3.1 Single Image Depth . . . 135
7.3.2 Multi-View Depth . . . 135
7.3.3 Temporal Consistency . . . 136
7.4 Methods . . . 138
7.4.1 Overview . . . 138
7.4.2 Models . . . 140
7.4.2.1 Depth, Pose & Optical Flow estimation . . . 140
7.4.2.2 FlowMLP: 3D Scene-Flow estimation . . . 140
7.4.2.3 TCENet: Temporal Consistency Enforcement . . . 142
7.4.3 Loss Function . . . 143
7.4.3.1 Geometric Loss . . . 144

7.4.3.2 Scene Flow Auxiliary Loss . . . 146
7.4.3.3 Diffusion Auxiliary Loss . . . 147
7.4.4 Implementation Details . . . 148
7.5 Results & Discussion . . . 150
7.5.1 Qualitative Evaluation . . . 150
7.5.2 Quantitative Evaluation . . . 154
7.6 Conclusions . . . 156
7.6.1 Learnings & Future Work . . . 156

8 Deployment of Deep Learning Models on HoloLens 158
8.1 Overview . . . 159
8.2 Introduction . . . 160
8.2.1 Contributions . . . 161
8.3 Background . . . 162
8.3.1 MLOps Overview . . . 162
8.3.2 Benefits of adopting MLOps . . . 163
8.3.3 Limitations of MLOps . . . 163
8.3.4 Tools & Frameworks . . . 165
8.4 Methods . . . 166
8.4.1 Components . . . 167
8.5 Results & Discussion . . . 170
8.6 Conclusion . . . 174
8.6.1 Learnings & Future Work . . . 174

9 Conclusions 176
9.1 Main Conclusions . . . 177
9.2 Barriers to Translation . . . 181
9.3 Future Work . . . 182

References 185

Appendices 201
9.4 Permissions for Reuse of Materials . . . 201
9.4.1 IEEE . . . 201
9.4.2 Taylor & Francis . . . 201
9.4.3 Springer Nature . . . 202

List of Figures

2.1 Venn diagram displaying the taxonomy of AI algorithms . . . 32
2.2 displaying (A) 3 layer feed forward neural network and (B) the computation within a single neuron. . . . 34
2.3 Illustrating the convolution operation on an image patch (blue grid) by applying a convolution filter (yellow grid) to produce the output feature map (grey grid). . . . 37
2.4 Displaying a Recurrent Neural Networks (RNN) process for 3 time steps in a sequence where the hidden layer states are shared across time. . . . 39
2.5 Exhibiting the difference in training architecture styles between (A) standard Generative Adversarial Networks (GANs) process and (B) MSG-GANs process. Where solid and dotted lines signify forward propagation and concatenation accordingly. In all cases, Discriminator model is a simple sequential Convolutional Neural Network (CNN) similar to [170] . . . 44
2.6 Displaying the diffusion and the denoising process for training a diffusion generative model. . . . 46
2.7 Displaying different methods of obtaining depth . . . 50

3.1 Overview of the proposed generator G model. Solid lines are forward propagation and dotted lines are skip connections. All connections related to $stream_1$ are colour coded in magenta, and similarly for $stream_2$ in green . . . 64
3.2 The generator G model. In the encoder stream, the red, green and yellow blocks denote convolution filters of size 7x7x3, 5x5x3 and 3x3x3, respectively. The blue blocks between the skip connections of $stream_1$ are bottle neck layers. In the decoder, all convolution filters are 2D. The solid lines show forward propagation and the dotted lines are skip connections. . . . 65
3.3 The decoder block. All connections from $stream_1$ are colour coded in magenta and for $stream_2$ in green. All the layers with learnable weights are 2D and are yellow. 66
3.4 Occlusion reconstruction for MIS sequences where, rows **a** display the original candidate frame, **b** the input masked frame and **c** the reconstructed output. Sequence Ref shows the reference frames for Sequences 1,2 & 3. Sequences 4-5 were occluded by random synthetic occlusions on the entire video with Occlusion-to-Image Ratio (OIR)s ranging from 60-70% & 30-40% respectively. . . . 73

3.5 Comparison of qualitative results on a sample of consecutive frames from the EndoVis2018 data, presented in each column. **a** the original candidate frame, **b** the input masked frame and **c** the reconstructed output from the proposed model,**d** output from [123] and **e** output from [25]. 74
3.6 Performance evaluation of different occlusion recovery methods under varying OIR(%) on MIS data from the Hamlyn database. 76
3.7 Loss Ablation results, comprising of L1 error (top left), Peak Signal to Noise Ratio (PSNR) (centre), Structural Similarity Index (SSIM) (top right), Inception Score (IS) score (bottom left) and Temporal SSIM error (bottom right). The results were attained from the hamlyn test dataset. The metrics shown were calculated between the generated output and the ground truth from the hamlyn test set. $-NoAdver.$ being the proposed model with no adversarial component, $-NoPerc.$ denotes no perceptual component, $-NoStyle$ denotes no style component, $-NoWarp$ denotes no warping loss component and finally, $-Proposed$ is the proposed model with the all the proposed loss components 76
3.8 Qualitative results for different loss function combinations. (a) input masked image, (b) ground truth, (c) proposed loss, (d) no style loss, (e) no perceptual loss, (f) no warping loss, (g) no adversarial loss. 77
3.9 Glove Effect visualisation, where rows **a** display the original candidate frame and **b** the output of the glove effect, where the contours of the tool and the end effector are purposely visible. 77
3.10 "Bike" sequence 78
3.11 "Dancer" sequence 78
3.12 Occlusion recovery, on natural scenes from the Davis 2017 dataset. The rows **a** display the original candidate frame, **b** the input masked frame and **c** the reconstructed output. 78

4.1 The proposed encoder-decoder architecture. The solid and the dotted lines denote forward propagation and skip connections respectively. The purple lines signify the output left & right disparity maps generated at 4 scales, each increasing hierarchically with a scale factor of 2. 88
4.2 The process of generating a randomly connected deep learning architecture. The red and the grey nodes are the input and the output nodes of the block, respectively. The black nodes are the convolution operations. The purple module is a multi-head self-attention module. 89
4.3 Showing the proposed learn-able skip connections methodology. a is the network topology of a standard U-Net and b is the proposed new network topology. Solid lines and dotted lines denote forward propagation and skip connections respectively. 90

4.4 Discriminator model. The solid black lines and the dotted lines denote forward propagation and skip connections respectively. The skip connection inputs are concatenated with the forward propagating feature map prior to being processed by the next layer. The pink lines show the extraction of multi scale feature maps. 93
4.5 Depth maps generated by the proposed model on the *SCARED* and *Hamlyn* test splits shown in rows A and B, respectively. Inclusion of LS signifies presence of learn-able skip connections and $noLS$ without. all models were trained with the standard L_{ssim} loss (Eq. 4.1) unless L_{Final} is specified, which signifies training with the proposed loss function (Eq. 4.5).The colour bar displays the depth range in mm. 96
4.6 Comparison of depth maps generated by our model with [63] and [166] on the *Hamlyn* test split. The last column re-displays the depth produced by our proposed method (column 2) with the green boxes highlighting the key areas where more details are visible in our depth maps. The colour bar displays the depth range in mm. 97
4.7 Depth maps generated by the proposed model on the *Kitti* test split. Inclusion of LS signifies presence of learn-able skip connections and $noLS$ without. all models were trained with the standard L_{ssim} loss (Eq. 4.1) unless L_{Final} is specified, which signifies training with the proposed loss function (Eq. 4.5). 99
4.8 Depth maps generated by the proposed model on the *Kitti* test split. Inclusion of LS signifies presence of learn-able skip connections. all models were trained with the standard L_{ssim} loss (Eq. 4.1) unless L_{Final} is specified, which signifies training with the proposed loss function (Eq. 4.5). The green boxes highlighting the key areas where for comparison where our proposed final method resolved greater detail in depth. 100

5.1 T-SNE visualisation of model embeddings on Kitti2015 test data (200 images). Labels 0 and 1 signify embeddings of left and right views, respectively. (A) generated from pre-trained [129], (B) our proposed StereoCRL trained with Eq. 1&2 (C) our model variant trained only via self-supervision Eq.2, on Kitti raw split (42,382 stereo frames). 105
5.2 The stereo contrastive learning framework. (A) The intuition of Contrastive Representation Learning (CRL) and (B) Our proposed CRL approach. 109
5.3 StereoCRL model architecture. Solid black and dotted lines denote forward propagation and skip connections respectively. 110
5.4 Decoder block schema. The solid and dotted lines signify forward propagation and skip connections, respectively. 111
5.5 Depth maps generated on the *Hamlyn* and the Kitti2015 test split. Inclusion of (Eq.) signifies the type of loss the model was trained with. 113

6.1 Masked Image Modelling (Masked Image Modelling (MIM)) pipeline for StereoMAE comprising of a weight sharing transformer encoder-decoder model. Where N and D are the total number of transformer layers that make up the corresponding models. . . . 121
6.2 The downstream process of supervised stereo depth estimation training. Where the feature extractor encoders pre-trained from MIM, are packaged with an off-the-shelf decoder for depth estimation. Any decoder of user's choice can be used. . . . 123
6.3 Reconstructions of the left images on (A-B) SCARED and (C-D) Middlebury samples. Results of MIM pre-trained StereoMAE via Mean Square Error loss [73] and the proposed perceptual loss shown in 3^{rd} and 4^{th} columns, respectively. The inputs were masked at 75% mask-to-image ratio. StereoMAE was not trained on these datasets. . . . 125
6.4 Left disparity outputs on (A) SCARED, (B) Middlebury and (C) ETH3D samples. All models were trained only on synthetic data. . . . 126
6.5 Masked Image Modelling (MIM) pipeline for StereoMAE with Vector Quantisation module [155]. Where N and D are the total number of transformer layers that make up the corresponding models. . . . 129
6.6 The downstream process of supervised stereo depth estimation training using the pre-trained StereoMAE encoder with Vector Quantisation module. . . . 129

7.1 Overview of the proposed methodology. . . . 138
7.2 Pipeline used for inference without test-time-refinement. . . . 139
7.3 Video sampling strategy with reference to the current frame I_t (highlighted in green). R signifies the re-projection function and the subscripts fwd and bwd imply forward and backward re-projection, respectively. . . . 139
7.4 The pipeline and the model architecture for FlowMLP and TCENet. (A) The *FlowMLP* model process and the corresponding auxiliary losses for training it (L_{flow} and L_{pixel}). (B) The *TCENet* diffusion process which operates on a sequence of conditioning images and their noise infused depth maps, and the corresponding auxiliary diffusion loss L_{diff}. . . . 141
7.5 The TCENet reverse denoising process. . . . 143
7.6 The key components of TCENet 3D-Unet architecture. . . . 144
7.7 Qualitative results comparison on consecutive DAVIS (A, B) and MPI-Sintel (C, D) test frames, emphasizing dynamic motion, occlusions, and camera movement. In each sample, comparisons are displayed with (1) TCE-Net, (2) single-image depth estimation θ_{depth}, which is MiDASv2 [171], (3) [261] and (4) [136]. Note, TCE-Net is deployed without test-time training (frozen), using the pipeline from Fig. 7.2. Whereas [261] and [136] undergo test-time training. . . . 151

7.8 Displaying the difference in quality of depth maps produced by TCE Net pre-trained with different θ_{depth} models. Where $A - E$ columns are different samples from the DAVIS and MPI-Sintel test data. first row is outputs of θ_{depth} as MiDASv2 [171] and second row is, LeRes [250]. 152
7.9 Qualitative results comparison on consecutive SCARED (A & B) test frames. In each sample, comparisons are displayed with (1) TCE-Net, (2) [261] and (4) [136]. Note, TCE-Net is deployed without test-time training (frozen), using the pipeline from Fig. 7.2. Whereas [261] and [136] undergo test-time training. 153

8.1 The proposed MLOps pipeline for running inference of DL models on images / videos streamed from Microsoft HoloLens 2 mobile device. Black, yellow, blue and orange boxes signify operations related to HoloLens, Microsoft Azure cloud, Docker and version control, respectively. 166
8.2 The estimated optical flow and depth maps for the corresponding input frames streamed from the HoloLens2 via the proposed MLOps pipeline. Optical flow generated from RAFT [212] and depth generated from TCENet running in parallel on Azure Virtual Machine (VM) in parallel. These output estimates were streamed back to HoloLens 2 device where a screenshot was captured. 172
8.3 The estimated depth maps from 3 methods running concurrently on Azure VM on the corresponding input frames streamed from the HoloLens2 via the proposed MLOps pipeline. The outputs of these models were streamed back to HoloLens 2 device and where a screenshot was captured. 173

9.1 Email showing Taylor & Francis granting access. 202
9.2 Email showing Springer Nature granting access. 203

List of Tables

3.1 Performance evaluation on random masks of OIR 30-40% on the Davis2017 test data. STV-Net, was trained only on MIS. 79

4.1 Showing the mean absolute depth error in mm for test dataset 1 from *SCARED*. The compared methods are the self-supervised methods submitted to this challenge. (M) signifies Monodepth method, and (S) Stereodepth. The presence of L denotes the type of loss function used for training and presence of (no LS) implies no Learn-able skip connections were used in the model. Best results are highlighted in bold. 97

4.2 Showing the mean absolute depth error in mm for test dataset 2 from *SCARED*. The compared methods are the self-supervised methods submitted to this challenge. Where (M) signifies Monodepth method, and (S) Stereodepth. The presence of L denotes the type of loss function used for training and presence of (no LS) implies no Learn-able skip connections were used in the model. Best results are highlighted in bold. 98

4.3 SSIM index on the reconstructed images generated via the estimated disparity maps on the *Hamlyn* test dataset. Higher value indicates better performance. Where (M) signifies Monodepth method, and (S) Stereodepth. The M in parameter count signifies a million parameters. Methods, A,B,C,D and E are [60], [246], [249], [63] and [166] respectively. 98

4.4 Performance evaluation on the Kitti Raw Dataset for distances up to 80m. All results use crop from [57]. For columns with red headers, lower value is better and for the columns with green headers, higher value is better. All the our proposed architectures are trained with L_{ssim} (Eq. 4.1), unless L_{final} is specified, which denotes training with final proposed loss function (Eq. 4.5). The best performing model variations are highlighted in blue. 100

5.1 Performance evaluation on the *Hamlyn*. Best results highlighted in bold. 114

5.2 Mean absolute depth error (mm) on SCARED. Best self-supervised results are highlighted in bold. 114

5.3 Evaluation Results on Kitti2015 Benchmark. Measuring % of disparity outliers (D1). Where 'all' signifies outliers averaged over all ground truth pixels image. Best results highligted in bold. 115

6.1 Results of generalisation without fine-tuning on ETH3D, Middlebury, and SCARED datasets. Lower is better for all metrics. Method highlighted in bold is best. . . . 126
6.2 Mean absolute depth error (mm) on SCARED. Best results are highlighted in bold. None of the models were fine-tuned on SCARED, inference only. 127

7.1 Image reconstruction quality assessment on DAVIS using M_{pe}^{TC}. Red header columns favor lower values, green headers prefer higher. Bold indicates the best model. Methods marked with † utilize LeRes for θ_{depth}, while others use default MiDASv2. 154
7.2 Image reconstruction quality evaluation on MPI-Sintel via M_{pe}^{TC}. Red header columns favor lower values, green headers prefer higher. The best result is highlighted in bold. Methods marked with † utilize LeRes for θ_{depth}, while others use default MiDASv2. 155
7.3 Spatial performance assessment on Kitti Raw Dataset (depth capped to 80m) using [57] crop. Red header columns prefer lower values, green headers higher. Bold indicates the best model. In the 2^{nd} column, the time step is relative to current frame at time step 0 hence, $-ve$ and $+ve$ values signify preceding and future frames respectively. 155

Acronyms

Abs Rel Absolute Relative Error. 99, 100, 154
ADA Adaptive Discriminator Augmentation. 45
ADAM Adaptive Moment Estimation. 36, 94, 124
AI Artificial Intelligence. 32, 160, 161, 163, 176, 177, 180
ANN Artificial Neural Network. 33, 34
API Application Programming Interface. 162

BatchNorm Batch Normalisation. 38

CNN Convolutional Neural Network. 14, 36–39, 44, 45, 48, 50, 51, 82, 138
CRL Contrastive Representation Learning. 16, 28, 47–49, 103, 104, 106–109, 112–114, 116, 178
CV Computer Vision. 31, 43, 47, 49, 50, 57, 82, 103, 161, 176

DAG Directed Acyclic Graph. 88
DevOps Software Development and Operations. 160, 162
DL Deep Learning. 8, 18, 23, 26, 28, 29, 31–34, 39, 40, 42, 49, 51, 52, 55–57, 80, 82, 101, 103, 116, 130, 158, 160–162, 165, 166, 168, 170, 176, 179, 180, 182
DNN Deep Neural Networks. 32, 35, 36, 45, 46, 48, 50

EPE End-Point-Error. 125, 126

FCN Fully Convolutional Network. 80
FFNN Feed Forward Neural Network. 34, 37, 39–41
FSIM Feature Similarity Index. 42

GANs Generative Adversarial Networks. 14, 31, 42–46, 92, 176
GBM Glioblastoma Multiforme. 24, 27
GDPR General Data Protection Regulation. 162
GRU Gated Recurrent Units. 39
GT Ground Truth. 33

HIPAA Health Insurance Portability and Accountability Act. 162

IS Inception Score. 15, 60, 72, 75, 76, 79

LSTM Long Short-Term Memory. 39, 81, 182

MAE Mean Absolute Error. 36, 42
MCMC Markov Chain Monte Carlo. 42
MIM Masked Image Modelling. 17, 26, 29, 47, 49, 53, 117–125, 128, 129, 179, 183
MIS Minimally Invasive Surgery. 8, 14, 15, 19, 24, 25, 28, 29, 53, 58, 59, 72, 73, 75–77, 79, 85, 94, 99, 105, 106, 112, 117, 119, 120, 130, 132, 135, 176, 178–180
ML Machine Learning. 31–33, 47, 57, 160, 162, 163, 165, 167–169
MLOps Machine Learning Operations. 8, 18, 23, 26, 28, 29, 158–170, 172–174, 176, 180, 184
MLP Multi Layer Perceptron. 8, 26, 34, 48, 108, 130, 138, 141, 156, 179

MR Mixed Reality. 8, 24–26, 28, 29, 31, 53, 55–57, 130, 158–161, 167, 174, 176, 179, 180
MRI Magnetic Resonance Imaging. 24, 27, 61
MS-SSIM Multi-Scale Structural Similarity Index. 92, 97, 112
MSE Mean Square Error. 36, 42, 72, 122, 125
MVS Multi View Stereo. 51, 52, 132, 135

NCE Noise Contrastive Estimation. 48
NLP Natural Language Processing. 49

OIR Occlusion-to-Image Ratio. 14, 15, 19, 58, 60, 61, 71–73, 75–77, 79
OS Operating System. 158

PSNR Peak Signal to Noise Ratio. 15, 42, 60, 72, 75, 76, 79

ReLU Rectified Linear Unit. 35
RL Reinforcement Learning. 33
RMSE Root Mean Square Error. 99, 100, 154
RMSE Log Root Mean Squared Logarithmic Error. 99, 100, 154
RNN Recurrent Neural Networks. 14, 39, 41

SfM Structure from Motion. 31, 50, 51, 131, 132, 156
SGD Stochastic Gradient Descent. 36
SOTA state-of-the-art. 23, 26, 28, 31, 39, 41, 48, 72, 82, 99, 103, 113, 114, 117–119, 131, 133, 176
Sq Rel Square Relative Error. 99, 100, 154
SSIM Structural Similarity Index. 15, 42, 60, 72, 75, 76, 79, 95, 113

TGV Total Generalized Variation. 107

VAE Variational Auto Encoders. 42, 45, 46
ViT Vision Transformer. 49, 121, 123, 124
VM Virtual Machine. 18, 158–161, 165–167, 170, 172–174

w.r.t with respect to. 36
WS Watts Strogatz. 87, 89, 93

Chapter 1

Introduction

This thesis is organised as follows:

- *Chapter 1 provides an introduction to all aspects of the thesis, the problem case, contributions and the publications / impact the work has had in the research field.*
- *Chapter 2 outlines the background information and relevant literature review required to understand the core concepts the research is built upon.*
- *Chapter 3 presents the novel generative model and training strategy for occlusion removal in any vidos from the wild. It particularly describes in detail the relevant literature, methodology, experimental details and results*
- *Chapter 4 builds on 3 and exhibits the novel solution for achieving state-of-the-art (SOTA) accuracy in self-supervised monocular depth estimation via generative modelling and perceptual learning for view-point reconstruction. It particularly describes all details about the solution as outlined in Chapter 3.*
- *Chapter 5 builds on 4, and extends the perceptual representation learning to stereo depth estimation, achieving SOTA accuracy in self-supervised stereo depth estimation. It describes all details about the solution as done in previous chapters.*
- *Chapter 6 builds on 5 and presents a generalisable stereo depth estimation model that achieves sub-millimeter depth accuracy in sugical scenes despite not training on surgical data. It describes all details about the solution as done in previous chapters.*
- *Chapter 7 amalgamates all innovations from 3-6 and creates a novel solution for estimating temporally-and-geometrically consistent depth on any video from the wild. It describes all details about the solution as done in previous chapters.*
- *Chapter 8 presents a novel Machine Learning Operations (MLOps), capable of deploying any Deep Learning (DL) model on mixed reality headsets via cloud compute. It describes the pipeline itself, methods and experiments conducted.*
- *Chapter 9 presents the final conclusions and recommended next steps for pushing this research forward*

1.1 Overview

Addressing malignant brain tumors, continues to be a significant challenge in the field of surgical oncology, e.g. patients with Glioblastoma Multiforme (GBM) have an average life expectancy of merely 12-15 months [4, 81]. Surgery is a fundamental component of brain tumor treatment, since brain tumor resection enhances patient survival rates significantly [19, 99, 202]. At present, neurosurgeons face difficulties in achieving complete tumor resections. This causes tumour recurrence around the operated region from the residual tumour cells left behind, which is detrimental to patient health [154]. The challenge lies in localising tumor tissue, which has similar appearance and texture to healthy tissue, making it difficult for the human eye to differentiate between the two. Additionally, limited intra-operative visibility, typically caused by swelling of nearby tissues, instruments, blood, or other fluids, further complicates the situation.

Neurosurgery currently, relies on pre-operative imaging i.e. Magnetic Resonance Imaging (MRI) or Ultrasound to visualise valuable information about the tumor's size, location, and relationship to surrounding structures and guide surgical planning [36, 185]. However, pre-operative imaging's usefulness for surgical-navigation is limited due to tissue / anatomical structure deformation; caused by factors like pressure, swelling, foreign instruments etc. This alters the tumor's position and shape, making it difficult to maintain accurate intra-operative visualisation and near impossible to localise during surgery (resulting in an incomplete tumor resection).

The emergence of robotics and Mixed Reality (MR) is going to play a pivotal role in Minimally Invasive Surgery (MIS) with the promise of improved patient benefits [23, 65] such as as less pain, scarring / infection, blood loss, and high precision when operating in confined spaces like ophthalmic surgery or on minute anatomical structures of children [91]. In particular, computer vision has acted as a powerful driving force behind advancements in both robotics and MR. In MR it has given rise to holographic technologies [72] giving clinicians multi-modal view by holographically overlaying pre-operative imaging data (i.e. CT scans of vascular networks) on patient's anatomy during MIS. MR, combined with computer vision-based scene understanding, can provide real-time assistance during tumor resection. Comparable to robotics [200], it can enhance 3D localization and mapping for improved surgical navigation and planning. This 3D contextual information can bolster a surgeon's spatial awareness of the tumor and adjacent anatomical structures (lessening dependence on 2D images alone). It can also grant the same valuable capabilities to improve autonomous surgical robotics, for the same tasks, with greater precision i.e. autonomous tissue tracking [255].

However, in real-world clinical setting, overlaying multi-modal information on anatomical structures during surgery presents challenges due to the dynamic nature of the surgical environment [218]. The very same challenges have also been the root of limited adoption of autonomous robotics in surgery. Throughout the surgical procedure, tissue deformation occurs for various reasons, such

as tool-tissue interaction, cardiac or respiratory motion etc. These factors cause the 3D geometry of the scene to constantly change, making accurate mapping difficult to maintain. Therefore, 3D scene-understanding fails in robotic and MR applications.

Hence, **the core aim of this thesis is to *estimate occlusion-aware, geometrically and temporally consistent depth of the scene***, such that computer vision systems can understand the 3D world. Analogous to the human visual system which is capable doing of this from two-dimensional images obtained by the eyes. Solving depth is the foundational key ingredient to bringing such MR and autonomous robotic technologies to real-world clinical settings. However, achieving such human-like depth perception in machines for MIS settings is particularly challenging as its an unconstrained problem. Specifically the following 4 points are the key factors.

1. **Occlusions**: obstructions from surgical tools and anatomical features can lead to unclear and incorrect depth perception. Furthermore, occlusions due to instruments or bleeding impede surgeons' visualization of the surgical area through keyhole incisions. This diminished visual clarity contributes to the relatively lower adoption of minimally invasive methods in neurosurgery and pediatric surgery [91].
2. **Sub-millimeter depth accuracy**: MIS requires extremely high depth accuracy at the sub-millimeter scale to ensure surgical precision and patient safety
3. **Dynamic scene motion**: intra-operative conditions are highly dynamic with constantly moving surgical instruments and deforming tissues which causes depth perception to fail in areas where regions between video frames fail to register across.
4. **Temporal depth inconsistency**: surgical scene changes over time, hence depth maps become temporally inconsistent. This is exacerbated by dynamic motion (3), and causes depth estimations to flicker in a video sequence leading to inconsistent geometry.

Hence, in this thesis I present novel solutions to address these key four challenges to advance the research for evolving MR and robotic applications to be robust to dynamic motion and occlusion by objects prevalent in MIS and lastly to achieve sub millimeter depth accuracy inferred from images alone. The following are the key objectives I completed during my PhD to achieve the aforementioned goal of creating occlusion-aware temporally consistent depth, and upgrade the surgeon's visibility during an operation.

- **Occlusion Removal**: to address (1) I created the first tailored video generative model that removes occlusions in surgical videos, indicated by a binary segmentation mask. It produces temporally consistent, high-quality reconstructed frames to enhance visibility of the surgical field. Despite training only on surgical data via perceptual learning, it also generalises to natural scenes. This algorithm was patented for its unique contribution to potentially also improve a surgeon's visibility. (Further discussed in chapter 3)

- **Monocular Depth Estimation**: Expanding on the highly generalizable perceptual learning from the occlusion removal generative model, I aimed to apply it to depth estimation through self-supervised learning. I created a multi-scale perceptual loss function that, even when trained with a randomly connected neural network for monocular depth estimation; achieves state-of-the-art (SOTA) performance on both natural and surgical scenes, surpassing several stereo models (discussed further in Chapter 4).
- **Stereo Depth Estimation**: Building on my single-image depth estimation and improvement work, I expanded it to stereo settings to address (2). Utilising cutting-edge perceptual learning I developed a unique contrastive representation learning framework for training stereo depth estimation models. This led to enhanced self-supervised model performance (detailed in Chapter 5). However it still lacked the accuracy and sharpness required for clinical standards. I found that stereo methods trained on simulations outperformed self-supervised approaches due to clearer training signals from ground truth depth, however their generalisability was limited. Consequently, I created a new method that obtained the best of both worlds: generalisability via perceptual learning and high accuracy via supervised learning (on simulation data). I adapted the perceptual learning method to enhance representation learning in a novel Masked Image Modelling (MIM) framework for stereo depth estimation. This framework achieved sub-millimeter accuracy on surgical scenes and generalised well, even without training on surgical data (discussed in Chapter 6).
- **Geometric & Temporally Consistent Depth**: Building on my all my previous research, I addressed (3) and (4) jointly, by developing a model capable of producing occlusion-aware, temporally consistent depth. I devised an innovative method that trains two models: (i) a Multi Layer Perceptron (MLP) for predicting 3D scene-flow, and (ii) a video depth estimation model for learning temporal dynamics via Diffusion. These models optimize each other for temporal depth consistency. I also demonstrate that by incorporating video dynamics, depth can be resolved for objects that are occluded and/or possess high dynamic motion despite the presence of moving camera. (discussed further in Chapter 7).
- **Mixed Reality Deployment**: Lastly, I created a novel Machine Learning Operations (MLOps) pipeline that operates on a cloud service (i.e. Microsoft Azure) to deploy and run inference for these Deep Learning (DL) models on data captured from a MR headset such as Microsoft HoloLens. Specifically, I demonstrate that, despite DL models requiring substantial computational resources, offloading to the cloud enables deployment in constrained environments. The software architecture also incorporates data and model version control, allowing users to track changes over time and retrain the model to tackle new challenges as data and usage evolves.

The next section Section 1.3 describes my contributions for the PhD for each of these aforementioned objectives.

1.2 Clinical Needs and Assumptions

GBM represents one of the most urgent and significant unmet clinical needs in oncology. Despite the availability of multimodal treatment strategies, including surgery, radiotherapy, and chemotherapy, GBM remains nearly universally fatal, with a five-year survival rate of only 5% [145][98]. One of the primary challenges in treating GBM is its highly infiltrative nature, making complete surgical resection impossible and contributing to nearly inevitable tumor recurrence [181]. This recurrence is often driven by glioblastoma stem-like cells (GSCs), which are highly resistant to conventional therapies. Furthermore, residual tumor cells left behind during surgery can lead to recurrence and a poorer prognosis, with average life expectancy remaining low at 12-15 months for GBM patients.

The current standard of care for GBM, which includes maximal safe resection followed by chemoradiation, only extends survival by a few months [145]. Furthermore, the complexity of the tumor's micro-environment and the blood-brain barrier (BBB) presents substantial challenges for drug delivery and efficacy, with over 98% of potential therapies failing to penetrate the BBB at therapeutically relevant levels. These factors highlight the critical need for innovative approaches, including advanced imaging and surgical technologies that can improve the extent of resection and overcome biological barriers. Therefore, addressing the clinical challenges posed by GBM is far from a mere technical push; it is a necessary response to one of the most pressing and lethal problems in modern neurosurgery. New advancements in technology, such as enhanced imaging for real-time tumor differentiation and novel drug delivery systems, could significantly improve patient outcomes and fill this substantial gap in cancer care.

A key clinical need in this field is the enhancement of intraoperative visibility to improve the precision of tumor resection. Current technologies such as preoperative MRI and intraoperative ultrasound are essential tools for surgical planning. However, these imaging modalities offer limited assistance during the operation due to intraoperative tissue deformation. The deformation caused by factors such as surgical manipulation and swelling often displaces the tumor, making precise resection difficult and increasing the likelihood of incomplete tumor removal.

To address this issue, my research focuses on the development of novel deep learning and mixed reality (MR) technologies aimed at enhancing real-time 3D scene understanding during surgery. By improving depth estimation, reducing occlusions, and providing geometrically and temporally consistent visualisations, this work directly targets clinical needs. For example, enhanced depth perception through MR could help surgeons more accurately differentiate between tumor and healthy tissue during operations, potentially reducing the risk of recurrence by facilitating more complete resections.

1.3 Contributions

This work is focused on creating occlusion-aware, geometrically and temporally consistent depth of the surgical scene. In particular, this thesis outlines innovations and novel methods that address the 4 key aforementioned challneges to achieve this solely from standard camera image / video inputs. It provides the foundation of novel solutions that can be further improved to bring MR / robotics to MIS. This required DL models (which have become SOTA method for computer vision tasks) to encode context & 3D representations from an image input and/or video temporal dynamics & object motion from a video input. The thesis, specifically focuses on the development of DL models via self-supervised representation learning to overcome the issue of lack of ground truth data availability (prevalent in surgery) and improving generalisation for unobserved scenes from the wild. The work finishes by presenting a novel MLOps pipeline to deploy these models on MR headsets.
The six areas of contribution are:

1. Video Occlusion Removal with High Temporal and Spatial Fidelity: A novel encoder-decoder network was created, combining 3D and 2D convolutions for learning from volumetric video sequences and single images. A new 3D partial convolution layer was introduced to construct the model and custom loss function was designed (combination of adversarial, reconstruction, perceptual, style and temporal losses) for unsupervised training with varying occlusion-to-image ratios. The model is invariant to occlusion and scene types, successfully reconstructing occluded areas in both surgical and natural scenes. Marking the first online video-based approach for removing occlusions in surgical videos. (chapter 3)

2. Monocular Depth with Self-Supervised Learning I designed an innovative neural network architecture for dense monocular disparity estimation, utilising random neuron connections and attention mechanisms for feature extraction, resulting in the first randomly connected encoder-decoder structure. Furthermore, we developed a multi-scale loss function that combines deep feature perceptual and adversarial losses for enhanced view synthesis, error computation and for learning the underlying 3D distribution of the input image. (chapter 4)

3. Boosting Self-supervised Stereo Depth with Contrastive Representation Learning (CRL): I developed a stereo depth model training procedure using joint CRL and self-supervised learning. I introduced the first CRL methodology for stereo depth estimation. The method involved efficiently training encoder models, which later pair with a decoder architecture for high-quality disparity maps. The CRL method boosts performance for models trained with traditional self-supervised image re-projection error. Additionally, we also made CRL scalable and tailored for

highly similar images like stereo pairs, even on single GPU hardware. (chapter 5)

4. Stereo Depth with Sub-Millimeter Accuracy: A groundbreaking technique was developed for training stereo depth models by combining self-supervised MIM and supervised stereo depth estimation. This marks the first use of MIM representation learning for stereo depth estimation. A new approach was proposed to augment MIM by integrating a perceptual similarity loss term for learning generalizable visual semantic concepts. A flexible model architecture was introduced, allowing the combination of MIM encoder with any off-the-shelf decoder for enhanced depth estimation. The joint MIM-supervised method improves performance, resulting in a generalisable model with sub-millimeter accuracy in surgical depth. (chapter 6)

5. Temporally & Geometrically Consistent Depth: I designed a new method for adaptable, geometrically and temporally consistent depth estimation using denoising diffusion. A 3D-Unet was designed for learning video dynamics, and a model for continuous differentiable 3D scene-flow was developed. The unique loss function proposed optimises depth video consistency and scene-flow jointly, by enforcing multi-view geometry constraints on a video sequence and a combination of step-unrolled diffusion together with flow warping for video reconstruction. This method enables the model to learn video temporal dynamics and ensures consistent depth is generated for the scene even in presence of objects with dynamic motion and/or obfuscated by occlusions (prevalent in MIS). (chapter 7)

6. Machine Learning Deployment Pipeline on MR Headsets: Finally, I developed an MLOps pipeline that deploys DL models on cloud services like Microsoft Azure, utilizing mr headset data such as HoloLens. In particular I created a software architecture with version control embedded to record any changes during usage and for deployment on the cloud, which overcomes substantial computational resource requirements. (chapter 8)

Chapter 2 establishes the technical foundation for concepts in this work; exploring common DL algorithms, generative modelling and self-supervised representation learning. These essentials of DL serve as the basis for the models developed in this thesis. Furthermore, the process of depth estimation in both traditional computer vision techniques and DL are explored. Finally, review of the challenges associated with deploying DL models on edge devices is explored.

The subsequent chapters (3-8) detail the contributions of this work as outlined above. Chapter 9 concludes the thesis, summarising the contributions, implications in current settings, limitations, and suggestions for future research directions.

1.4 Publications

1.4.1 Related Publications

- (Chapter 3) S. Tukra, H. J. Marcus and S. Giannarou, "See-Through Vision With Unsupervised Scene Occlusion Reconstruction," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 7, pp. 3779-3790, 1 July 2022, doi: 10.1109/TPAMI.2021.3058410.
- (Chapter 4) Samyakh Tukra & Stamatia Giannarou (2022) Randomly connected neural networks for self-supervised monocular depth estimation, Computer Methods in Biomechanics and Biomedical Engineering: Imaging Visualization, 10:4, 390-399
- (Chapter 5) Tukra, S., Giannarou, S. (2022). Stereo Depth Estimation via Self-supervised Contrastive Representation Learning. In: Wang, L., Dou, Q., Fletcher, P.T., Speidel, S., Li, S. (eds) Medical Image Computing and Computer Assisted Intervention – MICCAI 2022. MICCAI 2022. Lecture Notes in Computer Science, vol 13437. Springer, Cham.

1.4.2 Currently In Review

- (Chapter 6) Tukra, S., Giannarou, S. (2023). Generalisable Stereo Depth Estimation with Masked Image Modelling. Medical Image Computing and Computer Assisted Intervention – MICCAI 2023
- (Chapter 7) S. Tukra and S. Giannarou, "Boosting Monocular Depth with Temporal Geometric Consistency on Any Video," in IEEE Transactions on Pattern Analysis and Machine Intelligence 2023.

1.4.3 Other Publications

- **Patent:** S. Tukra, H.J Marcus, S. Giannarou, 2021, "Video Editing", Imperial College Innovations Ltd, Reference number: CSD/101231PCT1
- Tukra, S., Lidströmer, N., Ashrafian, H., Giannarou, S. (2021). AI in Surgical Robotics. In: Lidströmer, N., Ashrafian, H. (eds) Artificial Intelligence in Medicine. Springer, Cham.
- João Cartucho, Samyakh Tukra, Yunpeng Li, Daniel S. Elson Stamatia Giannarou (2021) VisionBlender: a tool to efficiently generate computer vision datasets for robotic surgery, Computer Methods in Biomechanics and Biomedical Engineering: Imaging Visualization, 9:4, 331-338, DOI: 10.1080/21681163.2020.1835546
- Cartucho, Joao, Alistair Weld, Samyakh Tukra, Haozheng Xu, Hiroki Matsuzaki, Taiyo Ishikawa, Minjun Kwon et al. "SurgT: Soft-Tissue Tracking for Robotic Surgery, Benchmark and Challenge." arXiv e-prints (2023): arXiv-2302.

Chapter 2

Background

The foundation of this work is rooted in learning deep semantic representations of videos and images. This is achieved by training a Machine Learning (ML) model which has parameters that are optimised to learn the probability distribution for a given task. This enables the model to accurately predict the desired outcome based on a given input. In hindsight they are function approximators; mapping an input to a desired output. Recent advancements in Deep Learning (DL) have earned them a reputation of being universal function approximators [39], making them the state-of-the-art (SOTA) ML algorithms applied in computer vision.

This chapter presents a thorough overview of the fundamental concepts that form the basis of deep learning, self-supervised learning, and generative modeling. I start by discussing the essential principles of deep learning, including artificial neural network architectures, backpropagation, and optimization techniques. I then examine the most prominent generative models, such as Generative Adversarial Networks (GANs) and diffusion models, emphasizing their core ideas, strengths, and drawbacks. Subsequently, I delve into the realm of self-supervised learning, investigating how it facilitates the unsupervised acquisition of meaningful representations from data, without the need for ground truth label data. This comprehensive exploration serves as a valuable resource for for advancing the future of Computer Vision (CV) techniques, which will be designed to learn a generalisable distribution of the underlying data and associated tasks

Then fundamental concepts of 3D vision i.e. depth estimation is discussed, including triangulation-based methods, Structure from Motion (SfM), and stereo depth, as well as their strengths and limitations. I also delve into more recent techniques that leverage the power of deep learning to achieve improved performance. Finally, the challenges associated with deploying DL models in on Mixed Reality (MR) / mobile devices considering computational resource constraints and data requirements.

In this chapter, I present the essential technical aspects of the methods employed in the contribution chapters of this thesis. The content is not meant to cover all aspects, but instead focus on the most common techniques related to the core concepts discussed in subsequent chapters.

2.1 Deep Learning

2.1.1 Machine Learning Preliminaries

Artificial Intelligence (AI) is the overarching field that focuses on developing algorithms, techniques, and models to create intelligent systems capable of performing tasks without explicit human intervention. It encompasses a wide range of approaches, including rule-based systems, search and optimization algorithms, and machine learning techniques [66]. Machine Learning (ML) is a subfield of AI that deals with developing algorithms that can learn patterns and make decisions based on data. Deep Learning (DL) is a specialised sub-field of ML that focuses on artificial neural networks with many layers, known as Deep Neural Networks (DNN). These networks are capable of automatically learning complex representations and patterns from large amounts of data. The taxonomy of these algorithms is displayed in Fig 2.1

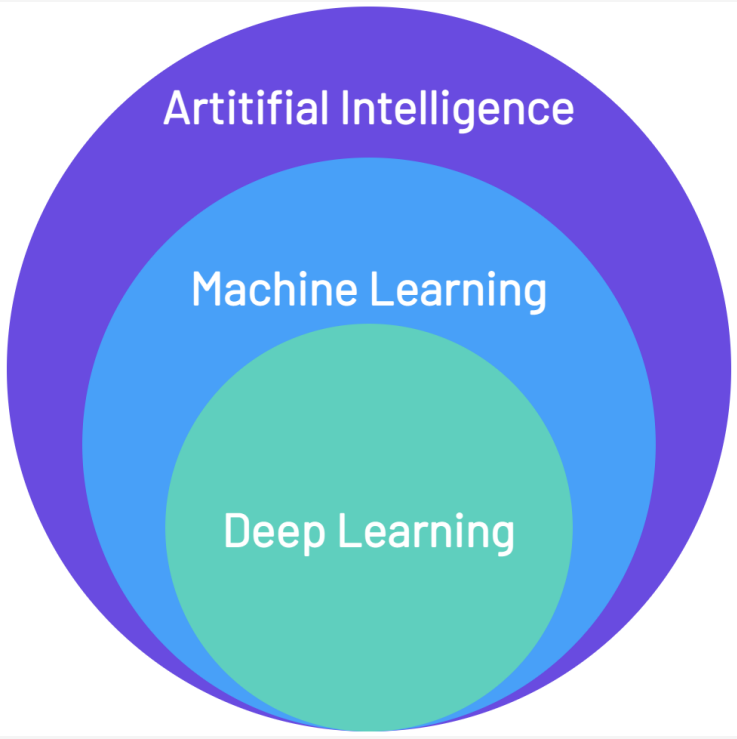


Figure 2.1: Venn diagram displaying the taxonomy of AI algorithms

.

A primary function of ML models is to carry out function approximation, which involves identifying an underlying function that represents the relationship between input and output variables in a given problem. The model is composed of learn-able parameters that are updated on a training dataset based on a specified metric. This metric becomes an objective function (also known as loss), which the algorithm optimises. Generally, ML algorithms are classified into two categories based on the presence of target labels in the training dataset: supervised or unsupervised learning.

A third category, known as self-supervised learning, has recently gained prominence, wherein the model learns to predict portions of its input using other parts of the input (inspired my language modelling [220]).

Supervised Learning In supervised learning, the training dataset includes Ground Truth (GT) labels that represent the desired outcomes for the model to predict during training. The model's parameters are updated based on the difference between its current prediction and the actual ground truth label from the training set. Typical supervised learning predictive tasks are classification and regression.

Unsupervised Learning In unsupervised learning, the training set has no GT labels, requiring the model to learn hidden patterns solely from the data's inherent features. For images, this might involve pixel intensities, local textures, or shape attributes. Unsupervised ML algorithms encompass clustering and dimensionality reduction techniques.

Self-Supervised Learning Also known as pretext task, is conducted as a preliminary step prior to conducting supervised learning. In this approach, the model learns to predict parts of its input (typically masked) using other parts (unmasked) of the same input, effectively generating its own labels from the data. In the context of images, a model could be tasked with reconstructing missing portions of an image based on the surrounding visible context. This enables the model to learn meaningful representations and features of the data distribution without relying on externally provided labels. Once this distribution is learnt, downstream supervised / semi-supervised learning has shown to yield boost in performance and generalisability.

There are additional branches of ML such as Reinforcement Learning (RL). It is a type of machine learning where an agent learns to make decisions by interacting with an environment. In RL, the agent takes actions to achieve a specific goal and receives feedback in the form of rewards or penalties. The aim of the agent is to learn a policy, which is a mapping from states to actions, that maximizes the cumulative reward over time. Such methods have trained agents to play video games [149], control robots [151] etc. However, this is not applied directly in the work presented in this thesis only the 3 aforementioned methods are relevant.

DL has emerged as a new breed of ML algorithms, possessing certain advantages over traditional ML methods, which have contributed to its widespread adoption and success in a variety of applications i.e. computer vision [2, 175, 182], natural language processing [40, 220], robotics [151], time series prediction [128], etc. Such advantages include, the ability to handle high-dimensional complex data; i.e. unstructured data such as images, audio, and text. Traditional ML algorithms often struggle with these types of data or require significant pre-processing and feature engineering to be effective. DL models comprise of Artificial Neural Network (ANN) stacked in layers. Each

ANN hence, leverages the power of numerous individual decision-making units (other ANNs, whose weighted contributions are combined to produce a cohesive output. Thus, as a model (multiple ANN stacked in multiple layers), they are capable of automatically learning hierarchical representations from raw data. Each ANN can discover and learn increasingly complex and abstract features, without requiring manual feature engineering.

These properties make DL models highly scalable and handle large volumes of data and high-dimensional input features for comples tasks like image recognition, natural language processing, and speech recognition. Furthermore, DL models can be trained end-to-end, meaning they can directly map raw input data to desired outputs without requiring intermediate processing steps. This simplifies the overall pipeline and often results in better performance. Moreover, the end-to-end learning also allows DL models to conduct transfer learning; where a pre-trained model can be fine-tuned for a specific task with relatively little data. This enables the model to leverage the knowledge gained during its initial training, improving the performance on the new task and reducing the training time.

2.1.2 Feed Forward Neural Networks

Feed Forward Neural Network (FFNN), also known as Multi Layer Perceptron (MLP), represent the essential foundation of deep learning models. A FFNN is a type of ANN consisting of multiple layers of interconnected neurons (also known as nodes), organized in a hierarchical manner. This is illustrated in Fig 2.2.

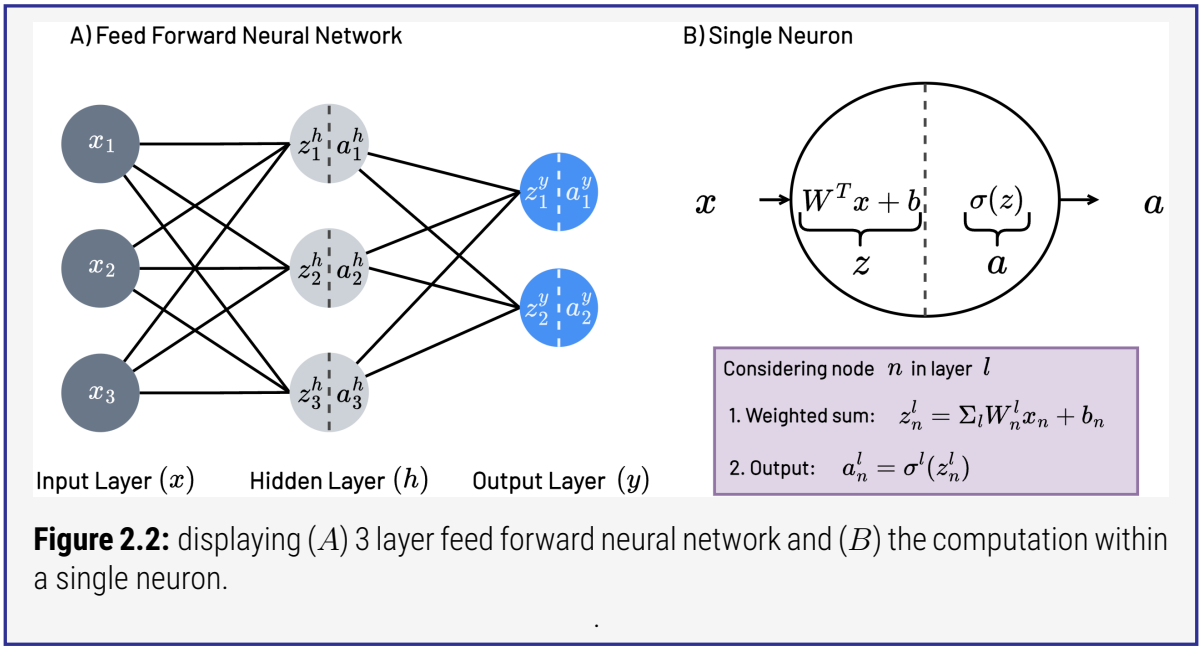


Figure 2.2: displaying (A) 3 layer feed forward neural network and (B) the computation within a single neuron.

In this network, information flows (forward) in one direction: from the input layer, through one or more hidden layers, and finally to the output layer. Each neuron processes the incoming data, applying a nonlinear activation function, and passes the result to the next layer. Given an input variable

x a single neuron conducts affine transformation of the input data in the following manner:

$$z = \sigma(wx + b) \tag{2.1}$$

Where w and b signify the neuron's parameters; weights and biases respectively. σ is a non-linear activation function e.g. logistic, sigmoid, tanh or Rectified Linear Unit (ReLU) [150]. Non-linear activation functions are critical, without it, a neural network would essentially reduce to a linear regression model, with limited capacity to model complex relationships between inputs and outputs. Hence σ allows neural networks to learn non-linear decision boundaries and map inputs to higher dimensional feature spaces, making them capable of solving a wide range of complex problems.

To understand how Eq. 2.1 is extended to a multi-layer neural network, consider a dataset ($X = [x_1, ..., x_i], Y = [y_1, ..., y_i]$), where X are the input samples, Y are the corresponding ground truth and i are the total no. of samples. Each sample x_i is a 3-dimensional input vector which is fed into the 3-layer DNN displayed in Fig 2.2 A. The weights which are the connections in of the DNN can be visualised as a matrix where the superscript denotes the layers the it connects to (start,end) and the subscript signifies the nodes it connects to in the corresponding layers (start,end). this is shown mathematically as:

$$W_{x,h} = \begin{bmatrix} w_{1,1}^{x,h} & w_{1,2}^{x,h} & w_{1,3}^{x,h} \\ w_{2,1}^{x,h} & w_{2,2}^{x,h} & w_{2,3}^{x,h} \\ w_{3,1}^{x,h} & w_{3,2}^{x,h} & w_{3,3}^{x,h} \end{bmatrix} \tag{2.2}$$

Where $W_{x,h}$ are all the weights between layers x and h. The same can be conducted for the final layer i.e. h and y. Essentially the no. of rows and columns in the matrix denote the no. of neurons and layers respectively. Similarly the bias matrix B_h for the layer h can be defined as $[b_1^h, b_2^h, b_3^h]$. The weighted output Z of layer h is calculated via the following:

$$Z_h = x_i W_{x,h} + B_h \tag{2.3}$$

Where Z_h is the output for each input x i.e. $Z_h := [z_1^h, z_2^h, z_3^h]$ (similar computation to Eq. 2.1, for the whole layer). This then is parsed through a non-linear activation function σ_h to get the final output a_h which is then forward propagated to the next layer, where the same computation Eq. 2.3 and 2.4 are repeated for the respective nodes connecting to layer (y).

$$a_h = \sigma_h(Z_h) \tag{2.4}$$

Where a_h is the output $a_h := [a_1^h, a_2^h, a_3^h]$ propagated to next layer. Considering the full forward propagation chain, input x_i to output predictions a_y, the calculation is as follows:

$$a_i^y = \sigma_y(\sigma_j(x_i W_{x,h} + B_h)W_{h,y} + B_y) \tag{2.5}$$

The neural network is then trained to optimise the weights w such that output a_i^y is same as its corresponding ground truth label y_i via backpropagation algorithm [66].

2.1.3 Optimisation

Continuing from the example above, to train a neural network, a loss function L needs to be defined, typically the difference between the prediction i.e. a_i^y and the ground truth label y_i. Most common loss is Mean Square Error (MSE), defined as:

$$L = \frac{1}{N}\Sigma_{i=1}^{N}(a_i^y - y_i)^2 \tag{2.6}$$

The type of loss function utilised depends on the type of task being performed i.e. classification losses typically utilise cross-entropy, whereas regression uses L1 error (also known as Mean Absolute Error (MAE)) or MSE. The aim of DNN is thus to best predict y_i. To optimise the weights W to achieve this, the loss i.e. Eq. 2.6 needs to be minimised. A common technique employed for this purpose is gradient descent. Gradient descent incrementally adjusts the parameters using a learning rate η to optimise the model's parameters w, defined mathematically as:

$$W = W - \eta\frac{\delta L}{\delta W} \tag{2.7}$$

To train all the weights of the DNN partial derivative of L must be calculated with respect to (w.r.t) the all the layer's weights W. This is where backpropagation, using the chain rule, comes into play. An example of this is shown mathematicallyf or the gradient of L w.r.t hidden layer weights $W_{x,h}$ below:

$$\frac{\delta L}{\delta W_{x,h}} = \frac{\delta L}{\delta W_{h,y}} \cdot \frac{\delta W_{h,y}}{\delta W_{x,h}} \tag{2.8}$$

The chain rule calculates derivative for each weight w in the DNN and gradient descent (Eq. 2.7) updates them iteratively until $L \approx 0$. Choosing the right learning rate η is vital, if too large, weight updates may overshoot, increasing the loss function L. If η is too small, the optimiser may get trapped in a local minimum.

Lastly, the above functions calculate the gradient over all the samples in a dataset, this may not be feasible for large datasets or models with larger parameters. Hence, different variants of gradient descent optimisers exist to update model parameters on batches of data i.e. Stochastic Gradient Descent (SGD) [204], SGD with momentum, Adaptive Moment Estimation (ADAM) [109] etc.

2.1.4 Convolutional Neural Networks

Convolutional Neural Network (CNN) are specialised neural networks tailored for processing data exhibiting grid-like arrangements, such as time-series data with its 1-D grid of consistent time intervals and image data composed of a 2-D pixel grid. CNNs incorporate two additional inductive

biases compared to FFNN; (1) local connectivity and (2) weight sharing. Instead of linking every input feature to every output feature, connections are limited to specific neighborhoods. This approach highlights the significance of local information, as output features rely on a small range of adjacent input features. Commonly, 1D CNNs handle time series, 2D CNNs manage images, and 3D CNNs process videos. In this thesis mostly 2D and 3D CNNs were utilised since, all input data were images and/or videos.

The convolution operation in CNN utilises a set of filters (also known as kernels), similar to traditional computer vision. The exception is that instead of having hand-crafter filters with pre-defined values e.g. Sobel filter [52], CNNs learn the filters (weights w) automatically during the training process. They adapt their weights to detect patterns and features relevant to the specific task, resulting in a more data-driven and flexible approach to feature extraction compared to hand-crafted filters.

When processing an image input, each filter consists of a set of neurons, typically organised in a square matrix, like a 3x3 grid. This filter glides across the input image in a sliding-window manner, starting from the top-left corner and moving towards the bottom-right. During this traversal, convolution operation is performed for each patch; which involves element-wise multiplication and summation of the filter's neurons and the corresponding input pixels. The resulting value is stored in the output feature map at the location corresponding to the filter's center. This procedure is performed iteratively at each sliding window position. As a result, the output is derived from the local information surrounding a specific point, capturing the spatial dependencies and local patterns within the input image. This is illustrated in Fig 2.3.

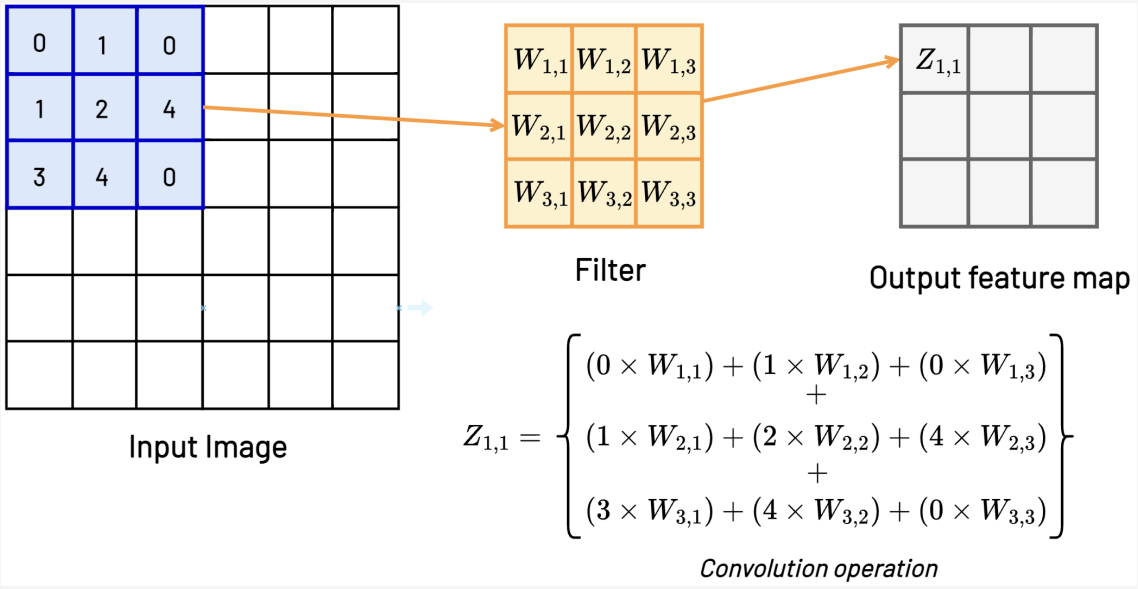


Figure 2.3: Illustrating the convolution operation on an image patch (blue grid) by applying a convolution filter (yellow grid) to produce the output feature map (grey grid).

When a filter convolves with an image, it functions in areas where it fully fits, with all the values inside the filter contributing to the output, which is positioned at the filter's center. Hence, the output feature map is smaller than the input image by $(f-1)/2$ pixels in each dimension due to edge constraints. f is the filter size i.e. $f \times f$. To counter this, padding, often zero-valued pixels, is added to the input image before convolution. It maintains the output size equal to the input size, by adding additional pixels typically 0 valued around the image perimeter. The value of the padded pixel can be user-selected.

The size of the output feature map is also influenced by the stride, which represents the number of pixels the filter shifts while progressing through the sliding window process. A single-pixel stride allows every input pixel to have a corresponding output in the resulting image, considering the padding issue. Hence, the output feature map size post convolution can be calculated by the following relationship:

$$size = \frac{(W - f + 2P)}{S} + 1 \tag{2.9}$$

Where W is the image width, f is the convolution filter size, P is the padding dimension and S is the stride. Typical input tensor dimensions for a 2D convolution layer is $[b \times c \times h \times w]$ where b, c, h and w are batch, channels, height and width respectively. Input tensor dimensions for a 3D convolution layer are $[b \times c \times s \times h \times w]$ where the added dimension s is sequence.

CNN architectures usually consist of not only convolution layers but also additional layers paired with them. These supplementary layers may include:

- **Pooling**: These layer are not parameterised, and are used to reduce the spatial dimensions of the feature maps generated by convolutional layers. This downsampling aids to make the network more robust to small variations, decrease the risk of overfitting, and reduce computational complexity of the model [188].
- **Dropout**: is a regularisation technique used in neural networks to prevent overfitting and improve generalization [77]. During the training process, dropout randomly deactivates a fraction of neurons (or connections) in a layer at each training iteration. This "dropping out" is typically controlled by a probability parameter, which determines the likelihood of a neuron being deactivated. By deactivating neurons, dropout effectively creates an ensemble of smaller, diverse networks that learn different aspects of the data
- **Batch Normalisation**: Batch Normalisation (BatchNorm) is used to stabilise and accelerate the training process. It addresses the issue of internal covariate shift, which occurs when the distribution of each layer's input changes during training, making it difficult for the network to learn. BatchNorm is applied to the input or output of a layer, normalising the data in each batch by calculating the mean and standard deviation, and then scaling and shifting the data accordingly. This process ensures that the data fed to the subsequent layers has a consistent distribution with zero mean and unit variance, making it easier for the network to learn [90].

2.1.5 Recurrent Neural Networks

Recurrent Neural Networks (RNN) are a type of neural network specifically designed to handle sequential data, such as time series, natural language, or audio signals. Unlike FFNN, RNNs have built-in memory through recurrent connections, enabling them to retain and utilize information from previous time steps in the sequence. This particular type of model was not employed in this thesis, mainly because only sequential data utilised was a video for which 3D CNN was utilised which learns spatio-temporal features. However, for complete DL literature, an overview of this architecture design is described below.

In a RNN, the hidden state at each time step is calculated using the input at that time step and the hidden state from the previous time step. This process allows the network to learn temporal dependencies and patterns in the data, capturing the context and order of the sequence. This process is illustrated in Fig 2.4.

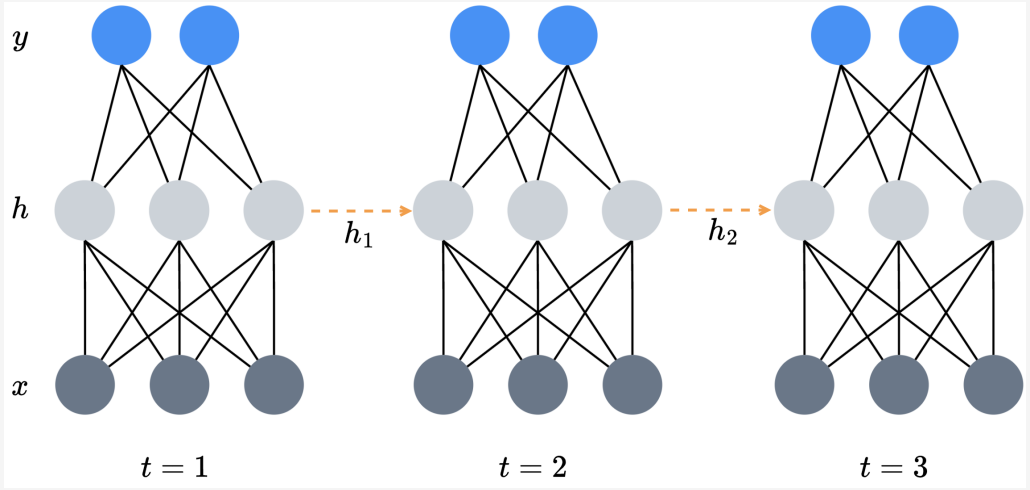


Figure 2.4: Displaying a RNN process for 3 time steps in a sequence where the hidden layer states are shared across time.

However, the standard RNN often struggles with learning long-range dependencies due to issues like vanishing and exploding gradients. To overcome these challenges, more advanced RNN architectures, such as Long Short-Term Memory (LSTM)s [80] and Gated Recurrent Units (GRU) [33], have been developed. These architectures introduce specialised gating mechanisms that enable better control over the flow of information and help to capture long-range dependencies more effectively.

2.1.6 Transformers

Transformers are a new breed of deep learning models, specifically designed for handling sequential data like text, audio, or time-series. They have superseded RNNs in sequence modelling [40] and even CNNs in certain computer vision [114] use-cases, achieving state-of-the-art (SOTA) results

and earned a reputation of a generalisable DL architecture [93]. They were introduced in [220] and have since revolutionised DL as a whole due to their ability to capture long-range dependencies and context in data effectively.

Transformers architecture typically consist of an encoder and a decoder, each comprising multiple layers with self-attention and FFNN. The encoder processes the input data and generates a continuous representation, while the decoder uses this representation to generate the output. The primary mechanism that powers transformers is self-attention, which allows the model to weigh the importance of each input element relative to others when processing the data. This self-attention mechanism enables the model to focus on different parts of the input depending on the context, leading to improved performance in any tasks. Since the model learns what to pay attention to, it can learn to generalise on any input type.

Self-attention is calculated for a given input x by converting it into query q, key k and value v feature vectors.

- Query q vector: Represents the element of x for which attention scores are calculated. It helps the model determine how much focus should be given to which element of x while processing x as a whole.
- Key k vector: Represents the elements in the input that should be compared against the query vector. The dot product between the query and key vectors is used to compute the attention scores, indicating the relevance of other elements in the context of the current element.
- Value v vector: Represents the content or "value" of each element in the sequence. Once the attention scores are computed, these value vectors are weighted by the corresponding attention probabilities, resulting in a context-aware representation that captures the influence of other elements in the sequence.

The value vector v is then forward propagated to next layer, so that hierarchically attention is calculated, all the way from the input to deep features and finally the output. Mathematically self-attention is calculated as follows:

$$A = softmax(\frac{Q \cdot K^{\intercal}}{\sqrt{d_k}})V \tag{2.10}$$

Where A is the attention output. Q, K, V are Query, Key and Value FFNN that take x as input, and output the q, k and v feature vectors respectively. d_k is the dimension of k, in Eq. 2.10 it is simply used to normalise the the dot product of Q and K. To boost performance the original authors showed [220] that instead of computing attention once on a given input x, computing multiple times i.e. multi-headed self-attention enables the transformer to capture different aspects / perspectives of the input data at various positions and scales. Thereby, improving the expressiveness and learning capacity of the self-attention mechanism. This is simply achieved by concatenating

all the different attentions for n no. of times conducted and parsing them through another FFNN for compression, i.e.

$$MA = FFNN(Concat[A_1, ..., A_n]) \tag{2.11}$$

Where MA signifies multi-head self-attention output, and n is the no. of times (heads) Attention is calculated. These attention scores are derived from the input elements' embeddings, which are then combined to form a context-aware representation for each element. The self-attention mechanism is highly parallelisable on GPU hardware, allowing transformers to efficiently handle large input sequences and scale effectively. This has made it a key component in SOTA deep learning models for various applications.

Another essential component of the transformer architecture is positional embeddings. Unlike RNN, transformers employ FFNN, which do not have shared weights across the sequence. Consequently, transformers require positional embeddings to incorporate positional information, as their self-attention mechanism lacks an inherent concept of element order or position in the input sequence. Various methods can be employed to create positional embeddings, the most common approach however, is sinusoidal embeddings [220].

In this method, each element in the input sequence x (i.e. $x = [x_1, ...x_i]$ where i is the dimension of the vector), is represented by a vector, with each element in the vector being a combination of sine and cosine functions of different frequencies. These embeddings can be precomputed, and they are added to the content embeddings (produced by the first FFNN layer in the transformer) before being fed into the transformer's self-attention layers. This is mathematically computed as follows:

$$pe_{pos,2i} = sin(\frac{pos}{10000^{2i/d_{model}}}) \tag{2.12}$$

$$pe_{pos,2i} = cos(\frac{pos}{10000^{2i/d_{model}}}) \tag{2.13}$$

Where pe is the output positional embedding for input x element at position pos of dimension i. In other words, each dimension of the positional encoding is represented by a sinusoid. The wavelengths exhibit a geometric progression, ranging from 2π to $10000 \cdot 2\pi$.

The sinusoidal positional embeddings have the desirable property, of being able to model relative positions thus, being easily extensible to different sequence lengths and different sequence types (i.e. sentences, audio waves, images etc.).

2.2 Generative Modelling

Generative modeling in deep learning refers to a class of algorithms that aim to learn the underlying data distribution of a given dataset and generate new, synthetic samples that closely resemble the original data. A variety of techniques, such as probabilistic graphical models, DL-based approaches like Variational Auto Encoders (VAE) and Generative Adversarial Networks (GANs), and other methods like Markov Chain Monte Carlo (MCMC), are employed to achieve this goal. Applications of generative modeling span numerous domains, including image synthesis [102, 105], drug discovery [173, 263], art [47, 187], and more, offering valuable insights and fostering innovation in diverse fields. In this thesis we focus generative modelling applied on image / video data.

2.2.1 Perceptual Learning

To train a DL model to generate high fidelity image / videos, a metric is required to measure the similarity between the predicted output and the ground truth. A naive way of achieving this is to use regression loss functions like MAE and/or MSE between the predicted and ground truth pixels. However, these are simply pixel-wise distance metrics that compare the predicted output to the ground truth. This is inherently low-level semantic comparison, it fails to evaluate high-level semantic information (such as relationships between objects in a scene) or structural similarities between similar contexts. Furthermore, These losses are sensitive to minor spatial mis-alignments which can yield poor optimisation and blurry outputs [262]. Hence, to overcome these limitations, perceptual loss functions are used, which evaluate perceptual similarity between images.

The goal of perceptual similarity [259] is to emulate human visual perception. Humans can interpret images at an abstract level, depending on high-level concepts and semantic cues that establish the connections between various entities within the scene. Perceptual similarity seeks to replicate this human-like assessment by formulating metrics that quantify perceptual distance, with smaller distances corresponding to image representations that more effectively capture visual semantic concepts.

Structural Similarity Index (SSIM), an early version of perceptual loss, aims to identify the characteristics of an image that, when altered, are perceived as significantly different by humans. Images are evaluated based on three crucial aspects: (i) luminance (*i.e.* pixel intensity), (ii) contrast and (iii) structure [233]. Subsequent research expanded this approach to assess similarity across multiple scales [230]. Concurrently, alternative metrics have been suggested, including Peak Signal to Noise Ratio (PSNR) [82], Feature Similarity Index (FSIM) [258], and HDR-VDP-2 [138]. However, these traditional perceptual metrics rely on handcrafted features and capture relatively simple properties of images, which can lead to poorer generalisation and limited semantic understanding

A popular way to measure perceptual similarity in the DL era, is to compare the features learned by a neural network with those from another frozen pre-trained network, rather than comparing pixel

differences. Computer Vision (CV) models pre-trained on large datasets like ImageNet (2 million natural scene images) [75, 195] can be used. This method assumes that higher-level, meaningful features needed for accurate classification are captured in this comparison. Introduced by [94] where a pre-trained VGG network (on ImageNet) is used for comparing features from predicted and ground truth images. This feature matching-based approach has been successfully applied to numerous computer vision tasks since its introduction [219, 224, 227]. This is mathematically defined as:

$$L_{feat}^{\phi=VGG} = \sum_{j=1}^{J=2,6,8} \frac{1}{N_j} ||\phi^j(I^{'})) - \phi^j(I)||_1 \tag{2.14}$$

Where ϕ is the pre-trained VGG model, j are the layer index at which feature maps from VGG are extracted for comparison, I is the ground truth image and I' is the predicted image. The loss function enforces (eventually during training) I' and I to have similar feature distributions, going beyond simply pixel-matching.

Perceptual learning and generative modeling are related concepts since they both involve learning meaningful representations of the data. In some cases, the representations learned during perceptual learning can be used as a basis for generative modeling. For instance, a model trained to recognise objects in images (perceptual learning) might have learned a latent space that captures the structure of the data, which could then be used to generate new images (generative modeling). In this thesis, particularly in chapters 3 to 7 I present innovative methods that combine perceptual representation learning with generative modelling to learn both high fidelity spatial outputs and the underlying data distribution.

2.2.2 Generative Adversarial Networks

While perceptual similarity seeks to encapsulate the semantic structure of images, generative models aim to represent the underlying distribution of image data. An example is GANs [101, 103, 104, 106]. In particular GANs comprise of two neural networks: a generator G and a discriminator D. The generator creates synthetic data samples, while the discriminator evaluates these samples, determining whether they are real (from the training data) or fake (generated by the generator). Hence, the discriminator essentially performs binary classification (real$= 1$ and fake$= 0$). Hence, the generator's goal is to produce samples that are indistinguishable from real data, while the discriminator's goal is to accurately identify whether a given sample is real or fake. Through this adversarial process, the generator learns to create more realistic data samples, and the discriminator becomes better at identifying real and fake samples. The training continues until an equilibrium is reached, where the generator produces high-quality synthetic data that the discriminator can no longer distinguish from real data.

The original formulation for training GANs [67] is based on binary cross entropy loss, which is

defined as follows:

$$L_G = \frac{1}{m}\Sigma_{i=1}^{m}[log(1 - D(G(z_i)))] \tag{2.15}$$

$$L_D = \frac{1}{m}\Sigma_{i=1}^{m}[log(D(x_i) + log(1 - D(G(z_i)))] \tag{2.16}$$

Where L_D and L_G are the loss functions for the discriminator and the generator respectively. z defines noise input for G, x is a real input sample from the data distribution and m defines a batch of data. L_D trains D to ensure D predicts the correct label for both real data x_i and generated $G(z_i)$. Whereas G is trained solely to maximise that D predicts 1 for generated data. As GANs implicitly learn the underlying data distribution through a discriminator, their original formulation was capable of generating high-fidelity images. However, they experienced training instability and mode collapse, where the network could only capture a portion of the variance present in the data distribution. This made training them very unstable. Hence, various different versions of GANs losses are introduced in literature that claim to improve stability in the loss formulation [9, 96, 140].

Later research, such as Pro-GAN [103], further improved by was further improved by MSG-GAN [101] (illustrated in Fig 2.5) incorporated the concept of multi-scale generation to stabilise the generator, allowing for a more comprehensive representation of the underlying data distribution. (Note the loss function for training these GANs is still effectively the same, the change is in the architecture design which improves training stability).

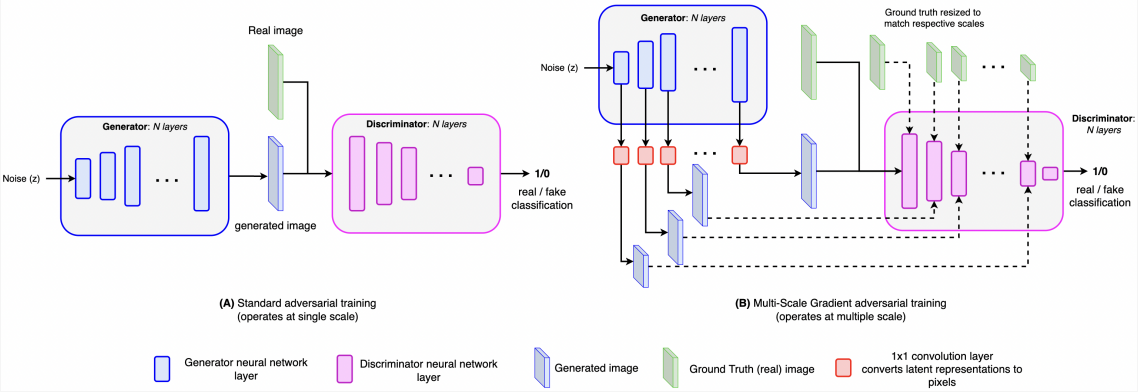


Figure 2.5: Exhibiting the difference in training architecture styles between (A) standard GANs process and (B) MSG-GANs process. Where solid and dotted lines signify forward propagation and concatenation accordingly. In all cases, Discriminator model is a simple sequential CNN similar to [170]

The recent StyleGAN series of papers [104–106] adopt the multi-scale GANs approach and introduce numerous enhancements on top, to the learning of the discriminator, aimed at increasing the stability of the generated images. One such improvement is perceptual path length regularisation [259], which ensures that minor alterations in the input latent code result in changes of

a similar magnitude in the discriminator's feature maps. This technique effectively normalizes the input codes, contributing to the overall stability of the generated images. Another advancement is Adaptive Discriminator Augmentation (ADA) [104], which allows for heavy augmentation during discriminator training, preventing overfitting even with limited training data while ensuring that these augmentations do not influence the generator's output. Both ADA and perceptual path length regularization contribute to a more stable underlying feature space in response to minor alterations in low-level image statistics. Another advancement is ADA [104], which allows for heavy augmentation during discriminator training, preventing overfitting even with limited training data while ensuring that these augmentations do not influence the generator's output.

In generative modeling, explicit generative models offer an alternative to GANs by aiming to directly represent the underlying data distribution. While these methods circumvent issues like mode collapse found in implicit modeling, they tend to be more sensitive to the amount of training data, as they need to model the entire distribution. Examples of such models include VAE [76, 110], Flow-based models [43, 111], and Diffusion models [41, 152].

2.2.3 Diffusion

Diffusion models, also known as denoising score matching or denoising diffusion probabilistic models, are a new class of generative models that aim to learn the data distribution via denoising process. These models leverage a stochastic process called diffusion to transform data samples into noise and then learn to reverse the process, recovering the original data from the noise. Recently, they have become the state of the art in image [42, 183] and video generation [79, 196] surpassing even GANs.

The diffusion process can be viewed as a Markov chain that transforms an initial data sample (e.g. an image) into a completely noisy version over a series of steps (timesteps t). At each step, a small amount of noise is added to the data, gradually obscuring the original content. This is illustrated in Fig 2.6. The key idea behind diffusion models is to train a DNN, typically a U-Net based CNN [182], is trained to learn the reverse of the diffusion process (i.e. remove the noise). The goal is to estimate the conditional distribution of the less noisy data, given the current noisy state, at each time step t. The network learns to map the noisy input back to the original data sample by minimizing the denoising score matching loss L_{diff}, which measures the discrepancy between the model's predictions and the true less noisy data. This is mathematically shown below in Eq. 2.17 and 2.18.

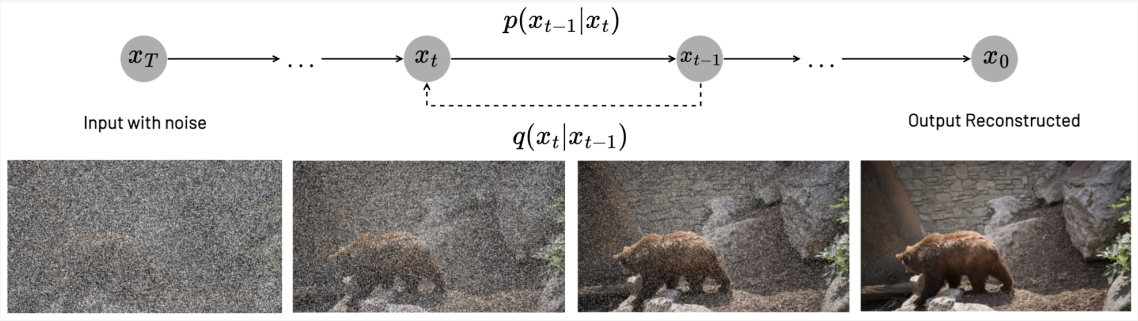


Figure 2.6: Displaying the diffusion and the denoising process for training a diffusion generative model.

Normally distributed noise q is added to the original image I iteratively for t steps. This is mathematically illustrated as:

$$q(x_t|x_{t-1}) := \mathcal{N}(x_t|\sqrt{\tilde{\gamma}_t}x_{t-1}, (1-\tilde{\gamma}_t)I) \tag{2.17}$$

Where γ_t is the variance for the normal distribution, following a linear schedule. To generate new samples, the diffusion process is reversed. Starting from a completely noisy version of the input, the model iteratively denoises the data, following the learned transition probabilities in the reverse order. Hence, DNN model (θ) is then reconstructing the input as:

$$p(x_{t-1}|x_t) := \mathcal{N}(x_{t-1}; \theta(x_t, t), \sigma^2(x_t, t)) \tag{2.18}$$

where $p(x_{t-1}|x_t)$ is the synthesised data sample modeled as a conditional distribution on the data distribution over t steps. σ^2 denotes the transition variance which remains fixed. Consequently, the training loss is merely the L2 distance between the actual noise added and the predicted noise. It optimizes a re-weighted variational bound on the maximum likelihood objective, ensuring a more accurate representation of the data distribution, defined as:

$$L_{diff} = ||x_{t-1} - \theta(x_t, t)|| \tag{2.19}$$

Compared to other generative models like GANs and VAE, diffusion models have shown promising results in terms of sample quality and diversity. However, they typically require more computation during the generation process due to the iterative nature of the diffusion process. However, one major drawback of diffusion models is their computational requirements during the generation process, as they necessitate multiple iterations to generate a sample. Recent research has focused on improving the efficiency of diffusion models while maintaining their generative capabilities [179].

2.3 Self-Supervised Representation Learning

Self-supervised learning is the latest cutting-edge sub-field of Machine Learning (ML) that focuses on training models using automatically generated supervision signals, derived from the input data itself, rather than relying on human-annotated labels. This approach aims to exploit the vast amount of unlabeled data available and learn meaningful representations or features from it, which can be later fine-tuned for specific tasks using smaller labeled datasets.

Self-supervised learning methods typically involve formulating a pretext task (also known as auxiliary task), which the model learns to solve using the input data alone. By learning to solve the pretext task, the model also learns useful features representations that can be transferred to other downstream Computer Vision (CV) tasks (i.e. classification, detection, etc.). Some common examples of self-supervised pre-text approaches in CV include:

- Predicting the relative position of image patches: Given an image, the model is trained to predict the relative position of n no. of randomly selected patches [44, 153].
- Image colorisation: The model is trained to predict the color channels of a grayscale image, learning spatial and semantic features in the process [260].
- Predicting distortion: Minor distortions in an image will not alter its original semantic meaning or geometric shapes. Consequently, images with slight distortions are regarded as equivalent to their original counterparts, and the features learned should demonstrate invariance to such distortions. Hence in this case, model is fed an image distored (i.e. rotated, flipped, re-coloured, etc.) and is tasked to predict the distortion [45, 62]

Self-supervised learning has gained significant attention, as it allows models to learn from large-scale, unlabeled data and reduce the reliance on expensive human-annotated datasets. During self-supervised learning the model has learnt rich context-aware semantic features from the data to solve the pretext tasks. These learned representations capture underlying patterns and structures in the data, which serve as a powerful starting point for transfer learning and fine-tuning on downstream task.

The two most notable forms of self-supervised learning in the field of computer vision are Contrastive Representation Learning (CRL) (also known as contrastive learning) [28, 107, 217] and Masked Image Modelling (MIM) [12, 73]. These methods have demonstrated significant improvements in downstream tasks and transfer learning scenarios, reducing the need for large labeled datasets.

2.3.1 Contrastive Representation Learning

The objective of CRL is to learn useful representations by enforcing similarity between semantically similar instances and pushing apart dissimilar instances in the feature space. CRL can be employed in both supervised and unsupervised contexts. In the case of self-supervised data, contrastive learning stands out as one of the most effective techniques in self-supervised learning.

The core idea of CRL is to create positive and negative pairs of samples. Positive pairs are instances that share similar semantics (e.g., two different views of the same object, or different translations of the same sentence), while negative pairs are instances with different semantics (e.g., different objects or unrelated sentences). The learning process then tries to maximize the similarity between the positive pairs and minimize the similarity between the negative pairs.

A typical CRL pipeline involves the following steps:

- **Heavy Data Augmentation**: Apply various transformations to the input data to create multiple views or versions of the same instance. This allows multiplication of positive pairs where a single image can be compared with itself as a positive pair due to transformations changing the appearence. These transformed instances will serve as positive pairs during training.
- **Feature Extraction**: typically a Convolutional Neural Network (CNN) is trained on these pairs to extract high-level features or representations from the input data i.e. extract embeddings. Once outputs are generated they are parsed into a Multi Layer Perceptron (MLP) which acts as a projection model, reprojecting Deep Neural Networks (DNN) embeddings to vector space.
- **Contrastive Loss**: finally a similarity metric (e.g., cosine similarity) between the extracted features of the positive and negative pairs, are computed. This is defined as the contrastive loss function. It encourages the model to maximize the similarity for positive pairs and minimize it for negative pairs. Common contrastive loss includes the Noise Contrastive Estimation (NCE) loss [71] or the InfoNCE loss [156].

The most successful CRL algorithms [107][74] are based on SimCLR [28] Which trains a ResNet50 to conduct the following steps and contrastively learn ImageNet classes. This pre-trained model once fine tuned on ImageNet via supervised classification task, achieves state-of-the-art (SOTA) results. However, They utilise extremely large batch size (8192 samples per batch), which requires intensive computational resources (in their paper 128 TPU cores were used). This is vital as the loss function can then cover a diverse collection of challenging negative samples, enabling the model to learn meaningful representations to differentiate various examples, only when the batch size is sufficiently large.

An additional, key necessity is that the image pairs the model is training with must have sufficient dissimilar samples, such that a contrastive loss can be computed. If majority of the pairs appear similar, then the contrastive loss will be almost 0 and the model will not learn the feature distribu-

tions to distinguish different contexts.

Overall, learning to differentiate between similar and dissimilar instances, CRL effectively captures the underlying structure in the data, enabling the model to learn meaningful and transferable representations without the need for explicit supervision. These learned representations can then be used for various downstream tasks, such as classification, object detection, or natural language understanding, often with minimal fine-tuning.

2.3.2 Masked Image Modelling

Masked image modeling is a self-supervised learning approach used in CV that focuses on learning useful representations by predicting missing or occluded parts of an image. The idea is to train a model to reconstruct the masked areas using the context provided by the surrounding unmasked regions, thereby learning meaningful features and structures in the data. This leads to models with feature representations that exhibit broad applicability across multiple tasks. This self-supervised approach has demonstrated notable success in Natural Language Processing (NLP). [17, 133, 220] and recently in CV too [12, 27, 73]. The general steps for masked image modeling are as follows:

- **Masking**: A portion of the image is randomly masked, creating an incomplete or occluded version of the original image.
- **Feature Extraction**: A Deep Learning (DL) model is trained (typically a Vision Transformer (ViT)) to extract features from the masked image. The model then learns to leverage the context provided by the unmasked regions to make sense of the image's content.
- **Reconstruction**: The model is trained to predict the occluded parts of the image.

The loss function then essentially becomes a reconstruction loss between the predicted outputs and the ground truth image. Initial work, such as iGPT [27], involved downsizing images and predicting unknown pixel values in an autoregressive fashion. More recent methods focus on predicting full-resolution patches within an autoencoder framework [12, 73, 243]. The most recent method Masked Auto Encoders i.e. MAE [73] adopts a simpler method, demonstrating that direct pixel prediction of masked regions using Mean Squared Error is effective when a significant portion of the image is masked. However, all current MIM methods learn form simple classification / regression loss functions which is low-level semantic comparison of the reconstructed image outputs (as outlined in Section 2.2.1). Hence, in this thesis, specifically in chapter 6 I show a way to improve MIM by incorporating perceptual learning for boosting performance.

Overall, MIM encourages the model to learn rich representations of the data by understanding the underlying spatial structure and context present in images. The learned representations can be fine-tuned for various downstream tasks, such as image classification, object detection, or semantic segmentation, often requiring less labeled data than fully supervised approaches.

2.4 3D Computer Vision

3D CV is a subfield of CV that focuses on understanding, interpreting, and reconstructing the three-dimensional (3D) structure of objects or scenes from visual input, typically 2D images or videos. The goal is to extract spatial and geometric information from the visual data to create 3D models or representations, which can be used for various applications, such as robotics, autonomous navigation, virtual and augmented reality, 3D object recognition, and medical imaging [52]. There are various approaches to obtain this, Fig. 2.7 shows a few strategies.

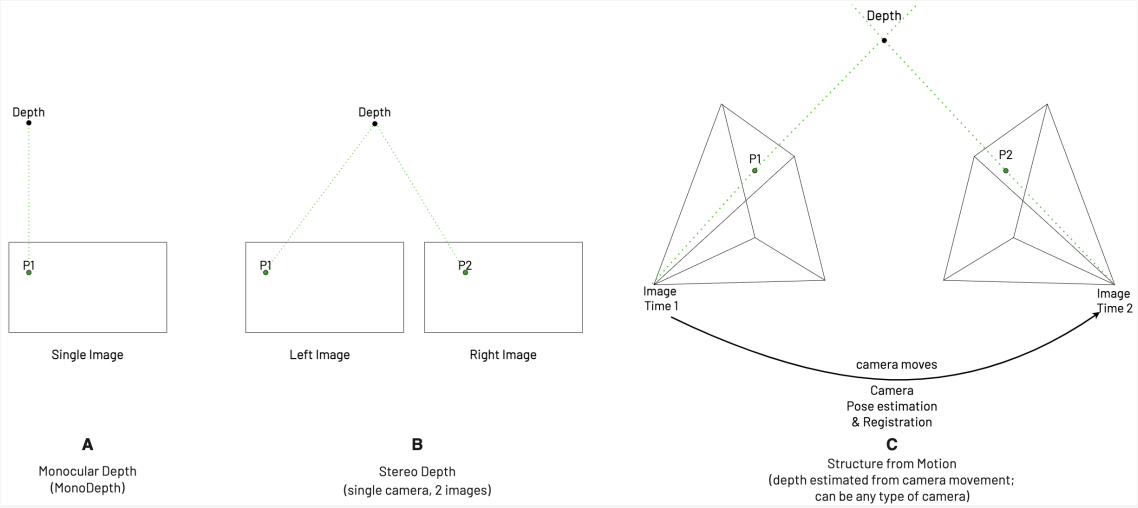


Figure 2.7: Displaying different methods of obtaining depth

2.4.1 Structure from Motion

Structure from Motion (SfM) involves reconstructing the 3D structure of a scene or object from a series of 2D images taken from different viewpoints. These view points can be a sequence in a video, or from arbitrary locations. It works by exploiting the motion parallax; i.e. the apparent change in the position of objects in the scene due to the movement of the observer (in this case, the camera). By analysing the changes in the object's positions across multiple images, SfM can estimate both the camera's motion (extrinsic parameters) and the 3D coordinates of the object points (structure) in the scene. The main steps involved in the SfM process are:

- **Feature Extraction**: Detect and describe distinctive points (features) in each image. Traditional feature detectors and descriptors can be used, such as SIFT, SURF, and ORB [52]. Or as done in more modern approaches a DNN can be used (typically a CNN) [211].
- **Feature Matching**: Traditionally these features need to be identified in corresponding pairs of images. Traditionally, this is done using techniques like k-nearest neighbors or matching algorithms such as RANSAC to find the most reliable matches while accounting for potential outliers.

In the DL era, this is automatically done within the model, of by development of a cost volume [237].

- **Estimating Camera Motion and 3D Points**: once the matched feature points are identified, relative camera poses (rotation and translation) and the 3D coordinates of the matched points are estimated using techniques like the eight-point algorithm, five-point algorithm, or bundle adjustment [52]. This step may also involve solving the essential matrix or the fundamental matrix to recover the camera motion and structure. In recent methods this just becomes a differentiable loss [237]. At which point the model estimates, depth end-to-end.
- **Triangulation**: Traditionally For each matched feature point, triangulate its 3D position by finding the intersection of the corresponding camera rays (projection lines).

2.4.2 Stereo Depth

Unlike SfM stereo depth is a technique used to infer depth information from two horizontally displaced perspectives of a scene at a known distance (baseline). This is analogous to the way the human visual system perceives depth using two eyes separated by a small distance. The key principle behind stereo depth estimation is the concept of triangulation, which exploits the geometric relationships between the cameras, the baseline, and the objects in the scene [52].

Similar to SfM process, it initiates with finding corresponding feature points in both images. Same traditional techniques can be used, however in the DL era a CNN is utilised [24, 70, 129]. These points represent the same object or feature in the scene. Once corresponding points are identified, the disparity (horizontal shift) between these points in the left and right images can be calculated. Disparity is inversely proportional to the distance of the object from the cameras i.e.

$$D_z = \frac{(f \cdot b)}{d} \tag{2.20}$$

Where D_z is the depth, f is the focal length of the cameras and b is the baseline, the distance between the two cameras. d is the disparity between the corresponding points. Hence depth is estimated through triangulation. Deep learning models are end-to-end thereby not requiring triangulation. The models are designed to take stereo images as input and directly output disparities. Most work is involved in designing an architecture that best compares the differences between the two input views.

2.4.3 Multi-view Stereo

Multi View Stereo (MVS) is an extension of stereo vision that involves using multiple images of the same scene, captured from different viewpoints, to estimate depth or reconstruct the 3D geometry of the scene. While stereo depth estimation typically relies on a pair of images, multi-view stereo

leverages the additional information provided by multiple images to improve depth estimation and 3D reconstruction accuracy. The main steps in a multi-view stereo approach are:

- Capturing multiple images of the same scene from different viewpoints. This can be achieved using multiple cameras or a single moving camera.
- Performing feature extraction and for traditional methods; matching across all images to find corresponding points. These points represent the same object or feature in the scene as seen from different viewpoints. In some cases, camera calibration and pose estimation are also needed to determine the relative positions and orientations of the cameras.
- Constructing a cost or similarity volume that represents the matching cost or similarity between corresponding points in the different images. This volume can be used to infer depth or disparity information for each pixel in the reference image(s).
- Optimising the cost or similarity volume to find the most likely depth or disparity values for each pixel. This can be achieved through various optimization techniques, such as global optimization, local optimization, or deep learning-based methods.

Lastly depth maps of the scene are created using the estimated depth or disparity values. In some cases, this may involve additional steps, such as surface reconstruction, point cloud densification, or mesh generation. Similar to stereo depth estimation DL models, many MVS dl models are end-to-end [89, 118, 248]. The key aspect of MVS is the creation of cost volume as it stores all the salient features for comparing the feature correspondences across the multiple frame inputs. A lot of work in dl literature focuses on designing and experimenting with architecture design and cost volume regularisation.

Multi-view stereo methods offer several advantages over traditional stereo vision techniques, including increased accuracy, robustness to occlusions, and the ability to handle complex and untextured surfaces. However, they are also more computationally intensive and sensitive to image quality, camera calibration, and feature matching errors.

2.5 Clinical Research Review

Masked Image Modelling (MIM) has undergone significant technological transformation over the past few decades. However, several challenges remain, particularly in achieving accurate depth perception and real-time 3D scene understanding during surgery. This review will cover foundational methods in robotic surgery and surgical imaging, showing how these early efforts set the stage for recent advances in deep learning and Mixed Reality (MR) technologies, as well as how the research presented in this thesis continues to push the field forward.

Robotic surgery has been a focal point of innovation in surgery since the 1990s, with early systems like AESOP and ZEUS paving the way for more sophisticated robotic platforms [98]. These early systems provided surgeons with enhanced precision and control, but their utility was limited by their reliance on 2D imaging and the absence of real-time intraoperative data. The introduction of the da Vinci Surgical System [113] in the late 1990s marked a major advancement, as it provided surgeons with a high-definition 3D view and improved dexterity through robotic arms. According to [162], hand-held medical robots have shown promise in augmenting a surgeon's dexterity and precision, particularly in complex anatomical regions such as the pelvis and cranium. However, the da Vinci system still relied heavily on the surgeon's manual input and had limited autonomy [145][177] and constrained by existing imaging and visualisation technologies at that time. Achieving a better understanding of real-time dynamics in surgery could significantly improve robotic performance, particularly in fields like neurosurgery where precision is paramount. The development of autonomous robotic systems capable of reacting to dynamic changes in the surgical environment is still in its early stages. [174] discussed the future of autonomous robots in surgery, highlighting the need for real-time processing of intraoperative data to improve surgical outcomes.

As robotic surgery became more widespread, researchers recognised the need for real-time depth perception to address challenges such as tissue deformation during surgery. The lack of real-time depth perception, makes it difficult for surgeons to differentiate between tumor tissue and healthy tissue, particularly in complex surgeries such as tumor resections in the brain. This lack of depth information often results in incomplete resections, which can lead to tumor recurrence. A study by [34] on image-guided interventions provided a framework for enhancing 3D scene understanding in real-time during surgery. They emphasised the potential for intraoperative imaging to guide surgical tools more accurately, particularly in delicate procedures such as brain and spinal surgery. Early efforts to enhance depth perception in Minimally Invasive Surgery (MIS) relied on stereo endoscopy [191, 201] and intraoperative ultrasound. Early algorithms, such as graph cuts and dynamic programming [29, 158], attempted to improve depth accuracy by optimising for smoothness constraints across pixel disparities. While these methods provided some improvement, they were hindered by poor resolution and a lack of robust depth cues, making it difficult for surgeons to navigate complex anatomical structures. More recently, deep learning approaches have revolutionized

depth estimation in surgery. Methods like SuPer Deep [135], a deep learning-based framework, address the limitations of traditional stereo matching algorithms by using neural networks to accurately estimate depth in real time. Similarly, research presented in this thesis on depth estimation through deep learning builds on these advancements by incorporating occlusion handling and temporal consistency, allowing for more accurate and reliable depth maps in dynamic surgical settings. This represents a significant leap forward from earlier methods, as it allows surgeons to navigate more confidently in minimally invasive procedures, reducing the risk of incomplete resections or accidental damage to surrounding tissues.

In the early stages of robotic surgery and MR applications, occlusion was managed through straightforward techniques such as stereo endoscopy and optical tracking systems (OTS). These systems often relied on fixed camera angles or manual adjustments by the surgeon to mitigate occlusions caused by surgical tools or other anatomical structures. For example, early image-guided surgery systems like StealthStation by Medtronic provided static overlays of preoperative images onto the surgical field, but they lacked real-time updates and could not handle dynamic occlusions. Recent advancements in occlusion handling have moved away from static solutions toward dynamic systems that leverage real-time data and sensor fusion to better navigate complex surgical environments. Recent advancements in occlusion handling [240] have moved away from static solutions toward dynamic systems that leverage real-time data and sensor fusion to better navigate complex surgical environments. [221] introduced a full-perception robotic surgery environment designed to address occlusion through a global-local joint positioning framework. This system integrates data from multiple sources to continuously track the positions of surgical instruments and anatomical structures, allowing it to dynamically adapt to occlusions as they arise during surgery. Another innovative approach comes from the work on Surgical-DINO, which applies deep learning for depth estimation in endoscopic surgery [38]. Although primarily focused on depth estimation, Surgical-DINO also addresses occlusion by using advanced vision transformers to predict occluded areas and reconstruct the hidden parts of the surgical scene. This allows for more accurate visualisation of surgical tools and tissues, even when parts of the scene are temporarily blocked by other objects.

The research presented in this thesis continues this trend of integrating real-time data with advanced machine learning models to handle occlusion. By developing deep learning models that not only estimate depth but also predict and reconstruct occluded areas.

2.6 Challenges of Deploying Deep Learning

Deploying deep learning models in real-world applications presents a unique set of challenges, which can be broadly categorized into computational requirements, data issues, model interpretability, and robustness

- **Computational requirements**: Deep learning models have grown in size, with top-performing models containing hundreds of millions to billions of parameters [18, 253]. This demands substantial computational resources for training and inference, posing challenges for deployment on low-power or resource-limited devices like mobile phones or Mixed Reality (MR) headsets. Techniques such as model pruning, quantization, and knowledge distillation help reduce model size and computational requirements without major performance loss. However, these methods may not yield consistent results for all model types. Furthermore, hardware accelerators like GPUs and TPUs have become essential for fast matrix multiplication but may not always be available and can incur significant costs. [190]
- **Data issues**: Deep Learning (DL) models are very data hungry which can be time-consuming and expensive to collect and annotate. Acquiring representative data samples for every possible scenario is often impractical, leading to models with reduced generalisation capability. Furthermore, even though Transfer learning, few-shot learning, and data augmentation techniques can help alleviate this issue; data-drift is a consistent problem [3]. This where the data distribution of the real world during inference is different from training, causing a domain shift effect. This causes model performance to drop. Furthermore, data privacy concerns may limit access to data in the first place, which inhibits deployment and growth of new methodologies. Current strategies rely on federated learning or differential privacy as a way around it, however these are still in research and development phase [176].
- **Model Interpretability**: due to the nature of hierarchial and deep hidden features in DL models, they are often seen as "black boxes", lacking transparency in their decision-making processes. This lack of interpretability can hinder the adoption of deep learning models in safety-critical or regulated industries such as medicine and surgery, here understanding the model's rationale for a particular decision is crucial and affects patient health. Techniques like integrated gradients [186] and attention [128] can be used to help build trust in model's outputs, though this is still an ongoing research field.
- **Robustness**: DL models are sensitive to adversarial attacks, input perturbations, or distribution shifts, which can lead to unexpected or undesirable outcomes [137, 159]. Ensuring robustness is a significant challenge, particularly when deploying models in dynamic or changing environments. Techniques like adversarial training, data augmentation, and domain adaptation can help improve model robustness, but guaranteeing complete robustness remains an open research question.

Lastly, in some cases there are ethical considerations to take into account (depending on application). This is particularly important in clinical field, where any DL model output / subsequent decision making can severely impact patient's health. Furthermore, presence of infrastructure that ethically stores data and maintains user and patient privacy is crucial, which is an on-going development. Additionally, DL models can inadvertently learn and perpetuate biases present in the training data, leading to unfair or discriminatory outcomes. Ensuring fairness and addressing ethical concerns are critical challenges when deploying these models. Techniques like fairness-aware machine learning, re-sampling, and re-weighting can help mitigate biases, but it is essential to continuously monitor and evaluate the model's impact on different demographic groups.

Given that the work in this thesis is based on 3D Reconstruction of scenes, and no testing was performed in a surgical theatre nor on a patient; ethics did not create a barrier to test on videos in the wild nor on open source datasets designed for experimentation. For any testing done on my personal videos i.e. from the phone and/or MR headset; I formally had given and give consent for use.

Chapter 3

See-Through Vision with Unsupervised Scene Occlusion Reconstruction in Videos

In this chapter I address the first problem outlined in Introduction: Occlusions. The presence of surgical tools and constant deformation of anatomical structures leads to ambiguous and erroneous depth perception. Additionally, issues caused by instrument blockages or bleeding through small incisions can hinder the surgeon's view of the surgical field. This decreased visual acuity contributes to the comparatively lesser adoption of minimally invasive approaches in neurosurgery and adoption of technologies like Mixed Reality (MR) to boost precision. Hence, the first task of my PhD was to create a method that can remove occlusions in a video recorded during surgery. To remove occlusions a Machine Learning (ML) model must understand the spatial scene context (from neighbouring pixels not occluded) including their relation in a temporal context (change in occlusion throughout a video sequence). Hence, the work presented in this chapter focuses on techniques from generative modelling to learn such spatio-temporal feature representations of a video. These features are then used to reconstruct the scene with the occlusion removed. I created the first online deployable occlusion removal Deep Learning (DL) model for surgical scenes, which generalised to natural scenes (without needed to train on it). The proposed method is the foundation of the work presented in the succeeding chapters (4-7) which follow the same theme of feature representation learning in a self-supervised manner to boost performance in different Computer Vision (CV) tasks. This feature representation learning is vital in surgery as it forces the model to learn a generalisable data distribution which can then be fine-tuned for specific use cases. The key learning elements of this chapter which aids in the succeeding ones are: a loss function for learning high spatial fidelity, video temporal dynamics and model architecture design for handling sequential inputs and sequence-to-sequence translation.

The work in this chapter was published in the IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) in July 2022 [219]. Furthermore, it has also attained a patent for being a radical new computer vision technology capable of enhancing surgeon's visibility during operation.

3.1 Overview

Among the greatest of the challenges of Minimally Invasive Surgery (MIS) is the inadequate visualisation of the surgical field through keyhole incisions. Moreover, occlusions caused by instruments or bleeding can completely obfuscate anatomical landmarks, reduce surgical vision and lead to iatrogenic injury. The aim of this work is to propose an unsupervised end-to-end deep learning framework, based on Fully Convolutional Neural networks to reconstruct the view of the surgical scene under occlusions and provide the surgeon with intraoperative see-through vision in these areas. A novel generative densely connected encoder-decoder architecture has been designed which enables the incorporation of temporal information by introducing a new type of 3D convolution, the so called 3D partial convolution, to enhance the learning capabilities of the network and fuse temporal and spatial information. To train the proposed framework, a unique loss function has been proposed which combines feature matching, reconstruction, style, temporal and adversarial loss terms, for generating high fidelity image reconstructions. Advancing the state-of-the-art, our method can reconstruct the underlying view obstructed by irregularly shaped occlusions of divergent size, location and orientation. The proposed method has been validated on in vivo MIS video data, as well as natural scenes on a range of Occlusion-to-Image Ratio (OIR) ratios. It has also been compared against the latest video inpainting models in terms of image reconstruction quality using different assessment metrics. The performance evaluation analysis verifies the superiority of our proposed method and its potential clinical value.

3.2 Introduction

MIS offers several advantages over traditional open surgery including less postoperative pain, fewer wound complications and reduced hospitalisation. However, it comprises of multiple challenges, among the greatest being inadequate visualisation of the surgical field through keyhole incisions. The human body has a complex anatomical structure, often deformed by disease and shifting during an operation, which makes navigation difficult. Moreover, the surgical field can become occluded by instruments or bleeding. The effects of such occlusions are exaggerated when working in confined spaces, such as the brain, or in children, where even slender instruments or moderate bleeding can completely obfuscate anatomical landmarks. The dangers are obvious and reduced surgical vision has been reported as a reason for the comparatively limited uptake of minimally invasive approaches in neurosurgery and paediatric surgery respectively [91]. The clinical corollary is that technological interventions that reconstruct occluded parts of the surgical field, and therefore enhance surgical vision, may improve surgical safety.

Thus far, methods based on inpainting have been proposed for image occlusion reconstruction and object removal. Inpainting the task of image completion, i.e. given an image that is damaged or suffering from empty pixels, the aim is to restore the image by filling these empty regions. The reconstructed regions must contain visually plausible information that matches the spatial statistics of the rest of the image. Inpainting methods can be classified into image or video based approaches. In the former case, a single image is processed and the spatial statistics of the input data are learned. However, when video sequences are processed, inpainting each video frame independently can not encode the temporal connections between the video frames which may produce results with no temporal consistency. To overcome this issue, video inpainting methods incorporate temporal information to deal with complex motions and the essential requirement of temporal consistency. Although video inpainting methods have higher computational complexity compared to image inpainting, they provide more accurate reconstructions due to the incorporation of prior information available in the previous frames. Recently, deep learning has been extensively used for inpainting, displaying significant progress in the field.

To this extent, the aim of this work is to use Deep Neural Networks to reconstruct occluded surgical scenes online by formulating the occlusion reconstruction as a video inpainting task. A novel fully convolutional architecture the so called See Through Vision network (STV-Net) is proposed based on a dual input stream encoder and a single decoder. The first stream aims at learning spatio-temporal features from a sequence of input frames, via 3D convolutions. More specifically, a new type of convolutional layer, the 3D partial convolution layer is introduced. The second stream utilises the optical flow between the current candidate frame to be reconstructed and one frame prior to it in the past. Using dense connections between the encoder and the decoder, the proposed network can generate high fidelity reconstructions of occluded areas in video frames. Furthermore,

a custom loss function has been designed to tune the model which enables the generation of temporally consistent reconstructions. This ensures that the model is generalisable to other types of scenes, without the requirement of fine tuning or re-training. For the method to be invariant to the shape, size, location and orientation of the occluded area, the model is trained with randomly generated masks following the example of [132]. The proposed method takes as input preceding frames from the current for reconstruction hence, recovering occlusions online

3.2.1 Contributions

The contributions of this work are:

1. A novel densely connected fully convolutional encoder-decoder network has been developed which combines 3D and 2D convolution operations to allow learning from volumetric (i.e. sequence of video frames) and single image data, respectively.

2. A 3D partial convolution layer has been introduced to learn from volumetric video data.

3. A bespoke loss function has been designed combining adversarial, reconstruction, perceptual, style and temporal losses to attain accurate and temporally coherent occlusion reconstructions on video data.

4. The proposed model is trained in an unsupervised way, using random irregularly shaped masks of varying OIR.

5. Our method is invariant to the shape, size, location and orientation of the occluded areas as well as, the nature of the scene as it can recover occlusions on both MIS data and natural scenes with no additional training or fine tuning.

6. To the best of our knowledge this is the first online video-based approach to reconstruct occluded areas in surgical scenes.

The performance of the proposed method has been evaluated both on MIS data and natural scenes in terms of image quality by estimating the L1 error, Peak Signal to Noise Ratio (PSNR), Structural Similarity Index (SSIM) and lastly, Inception Score (IS). Additionally, the temporal consistency of the generated reconstructions has been assessed based on flow warping photometric error. The method has also been compared to state-of-the-art deep learning-based online video inpainting approaches. Currently, the proposed method is capable of processing 24 frames per second on an NVidia P100 Tesla, for input images of size 256 x 256.

3.3 Related Works

This section identifies and describes the relevant literature that aims to address the issue raised in the introduction.

In literature, approaches to reconstruct occlusions are mainly based on image inpainting. Image editing methods such as, PatchMatch [13] interpolate the pixel values for the occluded region using the surrounding visible parts of the image. However, these methods do not consider the semantic information of the image such as texture, producing smoothened inpaintings which are distinguishable from the original image. Recently, deep learning-based methods have been proposed for image inpainting. In [132], the introduction of the 2D partial convolution layer, derived from regular convolutions, enabled the network to identify the information to use for inpainting. Binary occlusion mask are created to define the occluded and visible pixels. The convolution operation is performed on the input image, and its output feature map is multiplied by the occlusion mask before passing onto the next layer. However, the occlusion mask is progressively updated by converting its pixels corresponding to the reconstructed pixels in the output feature maps to visible. This allows the generation of the output image to be done in a hierarchical manner. Other deep learning-based methods which have been introduced recently utilise deep CNNs conjoint with adversarial training [252] and [88].

For medical images, image inpainting methods have been based on the Mumford-Shah functional [50], which segments the image into regions of different smoothness level, thereby using the sub region statistics for inpainting. Their main drawback is that they are limited to only performing inpainting on small synthetic occlusions such as straight-line drawings on brain Magnetic Resonance Imaging (MRI) images. Recent learning-based models [10], [199] are based on the generative adversarial modelling approach and have been tested on regular shaped occlusions such as, square image patches. These models have been only used to inpaint either 2D x-ray scans or MRI images and they have limited applicability since in real scenarios, occlusions come in varying shapes, sizes and orientations. All of the above methods inpaint, instead of reconstructing the underlying image structure and for OIR greater than 40%, their performance deteriorates and they ultimately produce blurred inpaintings. Moreover, none of the above inpainting methods utilise temporal information for scene reconstruction.

To incorporate temporal information, video inpainting has been proposed. Due to the additional time dimension, challenges such as, viewpoint changes, temporal inconsistency and high computational complexity are introduced. Earlier methods in literature extended the PatchMatch framework for video inpainting i.e. 3D based PatchMatch. More specifically, [238] considered video inpainting as an optimization problem such that the occluded regions could be filled with similar patches from the visible areas around it. This has been further developed by [86] which tackled the issue of camera motion by incorporating optical flow estimation for these patches of interest. However, these methods have limited applicability as they are based on the assumption that repetitive patches

appear in previous frames of the video. Furthermore, performing a 3D based PatchMatch method for video is highly computationally expensive rendering these methods unusable for real-time inpainting. Therefore, the above methods are not suitable for surgical applications where real-time performance is required and long occurring occlusions due to the presence of surgical tools or blood are prevalent.

Recently, deep learning-based methods for video inpainting have been introduced. The work in [108] proposes a fully convolutional network with a recurrent module, for performing the inpainting task. It is based on optical flow for generating temporal consistency. The method takes as input the current frame to inpaint, and a set of past and future reference frames to learn scene representations. Hence, the limitation of this method is that it can not run online as it requires future frames as input.

A more recent video inpainting method proposed in [123], utilises a 2-stage process namely, an alignment process and a context matching process. The alignment process aligns similar pixel patches between the current frame and previous frames. Then, the context matching process aims to find similarities between reference frames and the candidate frame. The neural network essentially learns to copy the relevant pixels through and paste it onto the candidate frame. However, since the method learns to copy and paste the relevant pixels with high similarity, this can become troublesome for scenes with non-uniform appearance or non-repeating patterns and varying illumination conditions.

The video inpainting model proposed in [25] employs 3D gated convolutions and uses an adversarial training procedure to enhance temporal consistency in the reconstructed frames via the introduction of a temporal discriminator. Furthermore, occlusion masks of regular shapes, ellipses and lines are randomly generated. These masks are assigned a movement trajectory for creating a motion-based mask dataset, imitating the movement of the occluded regions in a video. However, they don't necessarily capture the broad nature of potential occlusions due to the limited variability in shapes.

In this chapter, the STV-Net is proposed which is an unsupervised video-based approach for online occlusion reconstruction. An adversarial training approach is followed, comprising of three networks, one generator and two discriminators. The generator is the model that performs the reconstruction operation, whereas both discriminators are utilised for training purposes only. The generator is an end-to-end fully convolutional densely connected deep encoder-decoder network. The network takes as input a sequence of frames, including the candidate frame to be reconstructed. The inputs are processed through two separate streams. Stream 1 encodes features for reconstruction and stream 2 encodes features for flow displacement using optical flow. The encodings from each streams are combined together in the decoder, which learns to hierarchically reconstruct occlusions, generating images that are temporally coherent with the past frames and have high spatial fidelity. Additionally, 3D partial convolution layers are introduced to allow the model to

learn spatio-temporal features. The model is trained in an unsupervised way with occlusion masks of random shapes, size and locations to enable the method to be invariant to the occlusion type. No labelled or ground truth data is required. To the best of our knowledge this is the first video-based occlusion removal approach for surgical scenes.

3.4 Methods

The method consists of a generator G and two discriminators, a spatial discriminator D_S and a temporal discriminator D_T. The generator, reconstructs a video frame at time t given a sequence of S frames from the past. The generator in this work is a novel fully convolutional architecture comprising of both 2D and 3D convolutions. Furthermore, a bespoke loss function has been designed that integrates several loss terms, including an adversarial loss term. Thus, requiring the discriminator networks for adversarial training.

3.4.1 Model

The proposed generator model comprises of an encoder and a decoder with skip connections between them, similar to U-net [182]. The encoder performs downsampling, creating low resolution feature encodings, which are later upsampled by the decoder to reconstruct the unoccluded scene. Residual connections between the encoder and decoder are preferable over other standard encoder-decoder structures due to it's symmetric upsampling path, enabling precise localisation of features. The overview of the generator model G architecture is shown below in Fig 4.1, a more detailed view is shown in Fig 3.2. Furthermore, two additional models, the spatial and temporal discriminators are trained simultaneously. The discriminators are deep convolutional networks which learn and downsample the data up to the last layer where binary classification is performed.

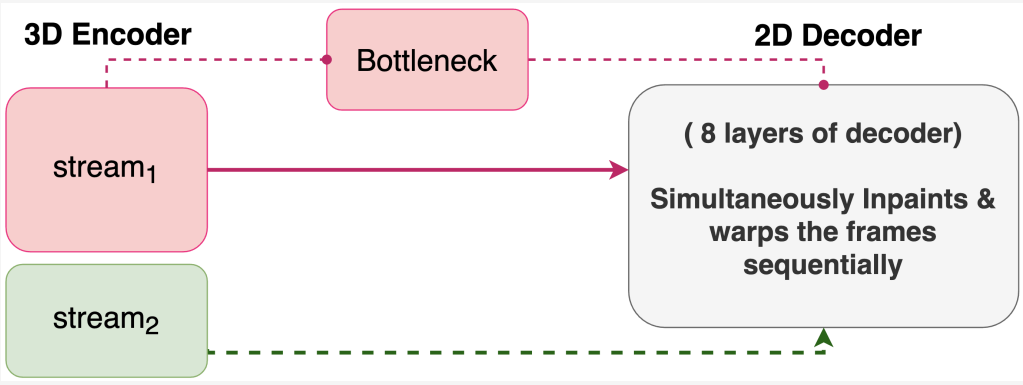


Figure 3.1: Overview of the proposed generator G model. Solid lines are forward propagation and dotted lines are skip connections. All connections related to $stream_1$ are colour coded in magenta, and similarly for $stream_2$ in green

.

3.4.1.1 Generator

The encoder is made of two input streams to allow the learning of both spatio-temporal features from the sequence of frames and flow displacement between the candidate frame and frames in

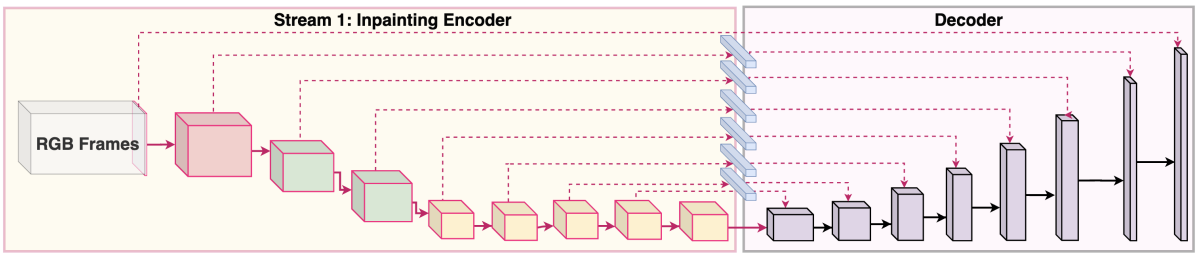


Figure 3.2: The generator G model. In the encoder stream, the red, green and yellow blocks denote convolution filters of size 7x7x3, 5x5x3 and 3x3x3, respectively. The blue blocks between the skip connections of $stream_1$ are bottle neck layers. In the decoder, all convolution filters are 2D. The solid lines show forward propagation and the dotted lines are skip connections.

the past. $stream_1$ takes as input a sequence of video frames, where the last frame is the candidate frame to be reconstructed. Given that the frame to be reconstructed is at time t, the frame sequence in the time interval $[t - S : t]$ is considered for processing. In our work, the frames in the past are sequential although this is not necessary.

The $stream_1$ encoder learns the relevant features for inpainting the candidate frame based on the sequence of past input frames. The features encode both the spatial information from the candidate frame to be reconstructed and the temporal information from the past sequence, as shown in Fig. 3.2. This enables the model to learn the relevant information conveyed in the past frames to reconstruct the candidate frame. To allow the processing of temporal volumetric data, 3D partial convolutional layers are introduced in this work and incorporated into the proposed $stream_1$ encoder. Similar to the 2D partial convolution operation [132], the 3D partial convolution is defined as:

$$output = \begin{cases} W^T \times (X \odot M) \times \frac{sum(1)}{sum(M)} + b, & \text{if } sum(M) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{3.1}$$

where, W^T is the 3D convolution filter that learns the weights for the task, b is its corresponding bias, X is the input data and M the corresponding binary occlusion mask.

In $stream_2$, an external optical flow network is used which acts as a catalyst for training the encoder of $stream_1$ and the decoder. In our experiments, the FlowNet [51] was used. It takes as input the candidate frame at time t and one frame prior to it at time $t-1$, to generate an optical flow map. This map is later scaled to create a pyramid such that it can be used in each layer of the decoder to warp the reconstructed frame to the previous frame to enforce temporal consistency. Note, the flow maps are not used as input into the network, but only to teach it to warp the reconstructed frame hierarchically in the decoder.

The feature maps from each layer of $stream_1$ (8 in total) are extracted and parsed by the decoder. This allows the decoder to conjointly learn the features for inpainting and for warping the output such that it is temporally consistent with the previous frame. The extraction of these maps also al-

lows retaining the learnt information from each encoding layer throughout the forward propagation process.

The feature maps from the $stream_1$ skip connections are 4D tensors of dimension (sequence x height x width x channels). However, these need to be compressed for reduction in computational memory and to form an output image which is a 3D tensor of dimension (height x width x channels). For this purpose, bottleneck blocks are used in the skip connection of each $stream_1$ layer, as shown in Fig. 3.2. These blocks consist of a series of 1x1x1 regular 3D convolutional layers. The 1x1 convolutions are favourable due to their ability to reduce the depth dimension of the extracted feature maps while learning from the input data, enabling a "learnable" compression [206]. The feature maps from $stream_2$ do not require compression as they are already 3D maps.

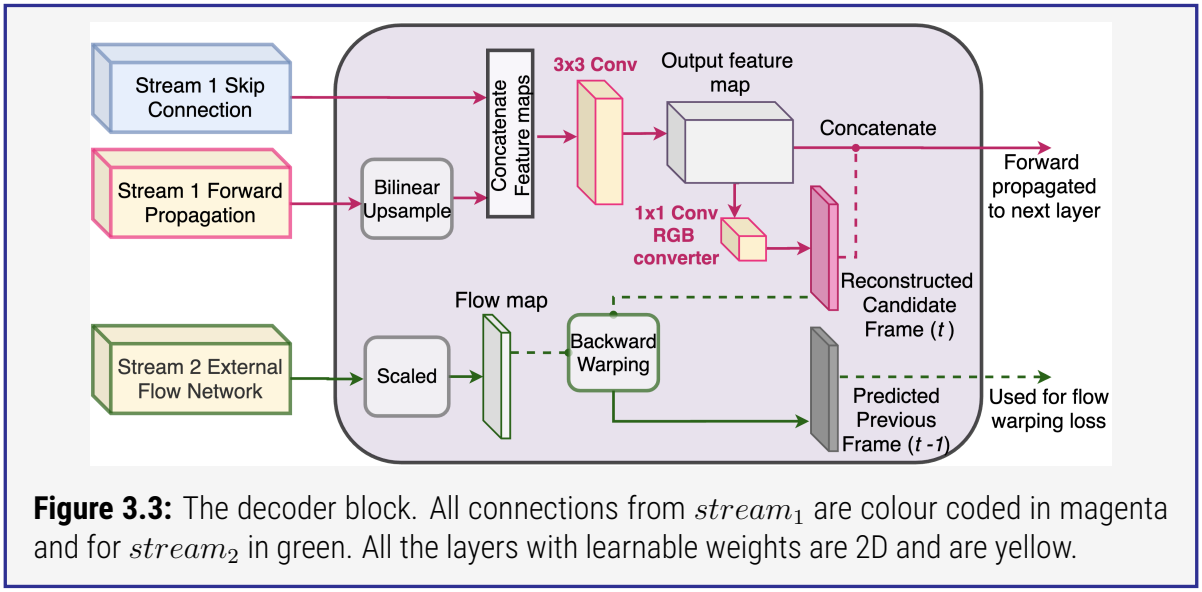


Figure 3.3: The decoder block. All connections from $stream_1$ are colour coded in magenta and for $stream_2$ in green. All the layers with learnable weights are 2D and are yellow.

In the decoder layer, there are two pathways as shown in Fig. 3.3. In $stream_1$ pathway, bi-linear interpolation using scale factor of 2 is performed on the forward propagated feature maps, followed by concatenation with the skip connection feature maps. Using 3x3 2D convolution the output feature map is produced which is then converted to a downsampled version of the reconstructed candidate frame using 1x1 convolution. This reconstructed frame and the output feature map are later concatenated and propagated forward to the next layer. In $stream_2$ pathway, the optical flow map is scaled to match the size of each decoder layer. By performing backward warping on the reconstructed frame with this flow map, generates the predicted previous frame which is later used to compute the warping loss.

$stream_2$ and its respective connections are only required during training, as they act as a catalyst for learning warping. The reconstructed previous frame generated via $stream_2$ in the decoder, goes through an unsupervised warping loss as explained in Section 3.2. This loss when back-propagated will update the convolution weights in both the decoder and encoder via gradient descent, such that the output feature maps and reconstructed frames generated in the decoder

are temporally consistent with the previous frame. Since $stream_2$ is used only during training, the proposed model does not rely on optical flow during inference. Furthermore, through hierarchical skip connections from the respective encoder layers, the reconstructed candidate frame is refined spatially at each forward pass. The final output of the decoder is a single image. Since, all the feature maps from the encoder streams are retained throughout the decoder stage, there are dense connections between the encoder and the decoder. These connections allow no loss of information by the network which is exploited by the decoder for reconstruction.

3.4.1.2 Discriminators D

The goal of both discriminators is to perform binary classification, to determine whether the input data is real (label 1) or fake (label 0) where the inputs are the generated and ground truth unoccluded images. Hence, the objective of the generator, is to fool the discriminator such that the output of the discriminator given the generated image is always closer to 1. The spatial discriminator D_S, evaluates a single frame content and structure, optimising for high spatial fidelity. The temporal discriminator D_T, evaluates a sequence S of frames, to ensure G learns the temporal dynamics of the input video, which D_S alone fails to capture. The network architecture for D_S is the pre-trained VGG network [195] with the final layer replaced for fine tuning. The D_T architecture is an eight layer 3D sequential convolutional network. Both the discriminators must be trained simultaneously.

3.4.2 Loss function

To successfully train this model separate loss functions are defined for the generator and the discriminator. The generator loss function designed in this work aims at integrating multiple aspects of image reconstruction quality namely, pixel-wise reconstruction accuracy, image generation quality, feature matching and temporal consistency. Hence, the loss function combines adversarial, reconstruction, perceptual, style and temporal losses as explained below.

The first term of the generator loss function is the **adversarial loss**. It is based on the relativistic Least Squares GAN [95] which forces the generator to create high quality images that fool the discriminator. Since, a neural network (the discriminator) is utilised in this loss component, it enables the generator to learn high level feature representations inherently, which other losses can not enforce. Considering the generated output of the model, G_{out}, and the ground truth unoccluded image, GT_{curr}, the adversarial loss is mathematically defined as:

$$\begin{aligned} L_{GAN} = &\frac{1}{2}E[(D(GT_{curr}) - D(G_{out}) + 1)^2] \\ &+\frac{1}{2}E[(D(G_{out}) - D(GT_{curr}) - 1)^2] \end{aligned} \tag{3.2}$$

where, D is the discriminator. Since our network includes a spatial D_S and a temporal D_T dis-

criminator, the adversarial loss is defined as:

$$L_{adv} = L^S_{GAN} + L^T_{GAN} \tag{3.3}$$

Where, the superscripts denote the respective discriminator.

The **reconstruction loss** is defined as the L1 error between the generated output G_{out} and the ground truth unoccluded scene GT_{curr}. Similar to [132], it is estimated separately for the visible and occluded components of the scene. The aim is to ensure that G_{out} still matches GT_{curr} under the occlusions (L_{occ}) whilst not distorting the original visible components from the input ($L_{visible}$). Hence, the L1 error for the occluded and visible regions is calculated as:

$$L_{occ} = ||(1 - mask) \bigodot (G_{out} - GT_{current})||_1 \tag{3.4}$$

$$L_{visible} = ||mask \bigodot (G_{out} - GT_{current})||_1 \tag{3.5}$$

To further improve the visual aspect of the reconstructed output, the **content perceptual loss** [94] has been incorporated into our loss function. This term represents the dissimilarity between the feature representations of G_{out} and the features of GT_{curr}. It forces G to learn the same feature distribution as found in the ground truth scene of the current frame. In this work, we further modified this loss to ensure that the generated output G_{out} also resembles the temporal features of the previous unoccluded frames in the sequence. This loss is defined as:

$$L_{perc} = L^S_{perc} + L^T_{perc} \tag{3.6}$$

L^S_{perc} is responsible for enforcing spatial feature matching with GT_{curr} and it is given by:

$$\begin{aligned} L^S_{perc} = \sum_{n=0}^{N-1} &||D^n_S(G_{out}) - D^n_S(GT_{curr})||_1 \\ &+||D^n_S(G^{fused}_{out}) - D^n_S(GT_{curr})||_1 \end{aligned} \tag{3.7}$$

where, D^n_S is the n^{th} layer feature map of the spatial discriminator. G^{fused}_{out} is created by overlaying the visible (unoccluded) parts from the input on G_{out}. Using VGG as our spatial discriminator, reduces the computational complexity of our system, as no additional networks need to be incorporated for D_S, and the rich feature maps of VGG are still retained. The term L^T_{perc} is responsible

for enforcing temporal feature matching and it is defined as:

$$L_{perc}^{T} = \sum_{n=0}^{N-1} ||D_T^n(G_{out}^{seq}) - D_T^n(GT_{curr}^{seq})||_1 + ||D_T^n(G_{out}^{seqfused}) - D_T^n(GT_{curr}^{seq})||_1 \tag{3.8}$$

where, the superscript seq denotes a sequence of each of the variables G_{out} and GT_{curr}. It is created by storing the S past frames in a queue structure where $S = 10$ in our work. D_T^n is the n^{th} layer feature map of the temporal discriminator. Since D_T directly learns the temporal features for the sequences via 3D convolution, perceptual loss is computed in a single forward pass. This feature matching with the discriminators directly also aids with optimising the adversarial loss faster. Since, the feature maps from G_{out} resemble GT_{curr} more over time due to L_{prec}, the discriminators are fooled quicker.

The **style loss** [94] is considered for performing neural style transfer between two frames. This is vital for learning texture representations from the ground truth. Similar to L_{perc} we conduct this loss in both spatial and temporal domain as:

$$L_{style} = L_{style}^{S} + L_{style}^{T} \tag{3.9}$$

Where L_{style}^{S} is defined as:

$$L_{style}^{S} = \sum_{n=0}^{N-1} ||\psi_n(D_S^n(G_{out})) - \psi_n(D_S^n(GT_{curr}))||_1 + ||\psi_n(D_S^n(G_{out}^{fused})) - \psi_n(D_S^n(GT_{curr}))||_1 \tag{3.10}$$

Similarly, L_{style}^{T} is defined as:

$$L_{style}^{T} = \sum_{n=0}^{N-1} ||\psi_n(D_S^n(G_{out}^{seq})) - \psi_n(D_S^n(GT_{curr}^{seq}))||_1 + ||\psi_n(D_S^n(G_{out}^{seqfused})) - \psi_n(D_S^n(GT_{curr}^{seq}))||_1 \tag{3.11}$$

where ψ_n are the gram matrices. For L_{style}^{T} we feed the sequence input as a batch of samples to enable calculation of the gram matrix.

A **warping loss** has also been incorporated. While L_{perc}^{T} enables learning temporal consistency in the feature space, the warping loss is used to learn temporal consistency in the pixel space. The warping loss is defined as the Structural Similarity Index (SSIM) [232] between the warped G_{out}

and the ground truth scene at $t-1$:

$$L_{warp} = \Sigma_{n=1}^{N} \frac{O^n \cdot SSIM(GT_{prev}^n, BW(G_{out}^n, F_{t-1\Rightarrow t}^n))}{N} \tag{3.12}$$

where, BW is the backward warping function, $F_{t-1\Rightarrow t}$ is the optical flow from $t-1$ to t and n is the respective scale. O is the occlusion mask, estimated using the range-map method [229] to include in the estimation of the loss only the valid pixels present in both images, excluding occlusions/disocclusion. All the temporal losses are vital, otherwise image-to-image based reconstruction without temporal coherence results in flicker artifacts due to mismatch of image patches caused by objects moving within the scene.

Lastly, we incorporate the **total variation loss** L_{tv} which acts as a smoothing penalty, enabling spatial smoothness and eliminating artefacts around the reconstructed regions. Similar to [132], this term is defined as the L1 loss, between adjacent pixels within the reconstructed image. Our final loss function for the generator is as follows:

$$\begin{aligned} L_G = \delta_1 L_{adv} + \delta_2 L_{occ} + \delta_3 L_{visible} \\ +\delta_4 L_{perc} + \delta_5 L_{style} + \delta_6 L_{warp} + \delta_7 L_{tv} \end{aligned} \tag{3.13}$$

The delta values above are found through training as explained in detail in Section 4.

The losses for both discriminators are identical, involving a classification loss. In the proposed loss function, the relativistic GAN discriminator loss has been employed [95], and more specifically the least squares version [139]. In contrast to the standard discriminator, the relativistic discriminator estimates the probability of how real the generated data appear relative to real data and vice versa. Mathematically expressed as:

$$\begin{aligned} L_D = \frac{1}{2} E[(D(GT_{current}) - D(G_{out}) - 1)^2] \\ +\frac{1}{2} E[(D(G_{out}) - D(GT_{current}) + 1)^2] \end{aligned} \tag{3.14}$$

Thus, the aim becomes not to classify, but to increase the probability of the generated data appearing, realistic relative to real data. The losses for the spatial and temporal discriminators are denoted as L_D^S, L_D^T, respectively.

3.4.3 Implementation Details

Our training dataset includes videos from the Hamlyn database only while, the EndoVis and Davis databases plus, the unseen sequences from the Hamlyn database were used for testing. Individual image frames were extracted from the videos and resized to 256×256 pixels. A total of 39 distinct videos were used for training, equating to (144,798 frames) and 15 distinct videos were used for

testing (19,631 frames).

The videos frames were processed sequentially. For the EndoVis and Davis data, we dilate the segmentation mask defining the occlusion by a disk of size 15 pixels to enlarge the occluded region. Then, the visible parts of the original input image (un-eroded mask) are fused onto the occlusion reconstruction output. This is to remove any artefacts formed around the borders of the occluded regions. However, the dilation rate is an arbitrary variable and can be chosen by the user upon experimentation.

The model was not trained on explicit masks. Instead, random occlusion masks of varying size, orientation and OIR, were created during training using the irregular masks training set [132]. In addition, random augmentations were performed on these masks involving, rotations (in the range [0, 90] at 25intervals) and flipping. All occlusion masks, are in the form of binary masks, where 0s correspond to occlusions and 1s to visible image parts. Performing element wise multiplication between a binary mask and the original image results in an occluded image. Both the previous frames and the current frame were occluded with a random mask. Thus, the network, had to learn to reconstruct the video with the relevant spatio-temporal information learnt from both encoding streams. In the generator loss function, the weightings δ_1, δ_2, δ_3, δ_4, δ_5, δ_6, δ_7 are set to 0.5, 6, 2, 0.08, 100.0, 0.7, 1.0, respectively. Adam optimizer is used with a learning rate of 0.0002 and beta coefficients of (0.5, 0.999). A batch size of 1 was utilised and convergence was achieved by an average of 300,000 iterations (3 epochs).

3.5 Results & Discussion

3.5.1 Validation Datasets and Performance Metrices

The proposed framework has been extensively validated on both MIS data and natural scenes. This includes videos available at the Hamlyn Centre video database [61] as well as, the EndoVis2017 [7], EndoVis2018 [8] and DAVIS2017 [169] datasets. The method has been tested using a range of OIR. The performance of our occlusion recovery method has also been compared to two state-of-the-art online video inpainitng methods namely, the CPNet [123] and GateTemp[25]. These methods were selected as they were the latest ones at the time of this work. The performance evaluation is based on a number of quantitative measures including, L1 error, PSNR, the SSIM, IS and lastly temporal consistency. L1 error is the direct pixel difference between the generated output and the ground truth hence, lower values denote less error. PSNR is similar to Mean Square Error (MSE) however, MIS is dependent on the image intensity scaling whereas PSNR eradicates this problem by scaling MSE to the image range. Higher values signify greater signal to noise ratios in the reconstructed images. SSIM measures similarity between two input images by comparing 3 components; mean, variance and cross-correlation over small patches of the image. Higher values of SSIM signify greater structural similarity between the generated image and the ground truth patches. The fundamental idea of IS is that the feature distribution of the generated image and the ground truth must be as close as possible, therefore, the higher the IS value the better. Lastly, temporal consistency is evaluated in the form of photometric error, as in Eq. 3.12.

3.5.2 Performance Evaluation on MIS Data

In this Section, the performance of the STV-Net is analysed and compared to the state-of-the-art (SOTA). The ability of the proposed method to recover occlusions due to the presence of surgical tools in the scene is shown on various MIS sequences in Fig. 3.4. As it can be observed, the proposed model generates high quality outputs for all the video sequences.

In Sequences 4 and 5 it can also be observed that the method can successfully recover occlusion even from highly irregularly shaped masks, and still learn the relevant information from the previous visible regions and the spatial statistics of the current frame. As shown in Sequence 4, even when the majority of the image is occluded (OIR set to 60-70%), the model still generates temporally consistent high fidelity reconstructions. It is interesting also to highlight that with the STV-Net model lighting is inferred well, despite the presence of large occlusions. Scene areas with different lighting conditions are accurately reconstructed. Similar conclusions can be derived from Sequence 5 where random occlusion masks of smaller size (OIR set to 30-40%) have been applied on MIS data.

The performance of the proposed occlusion recovery method is qualitatively compared to the CP-Net [123] and GateTemp [25] approaches in Fig. 3.5. As it can been seen, the proposed method

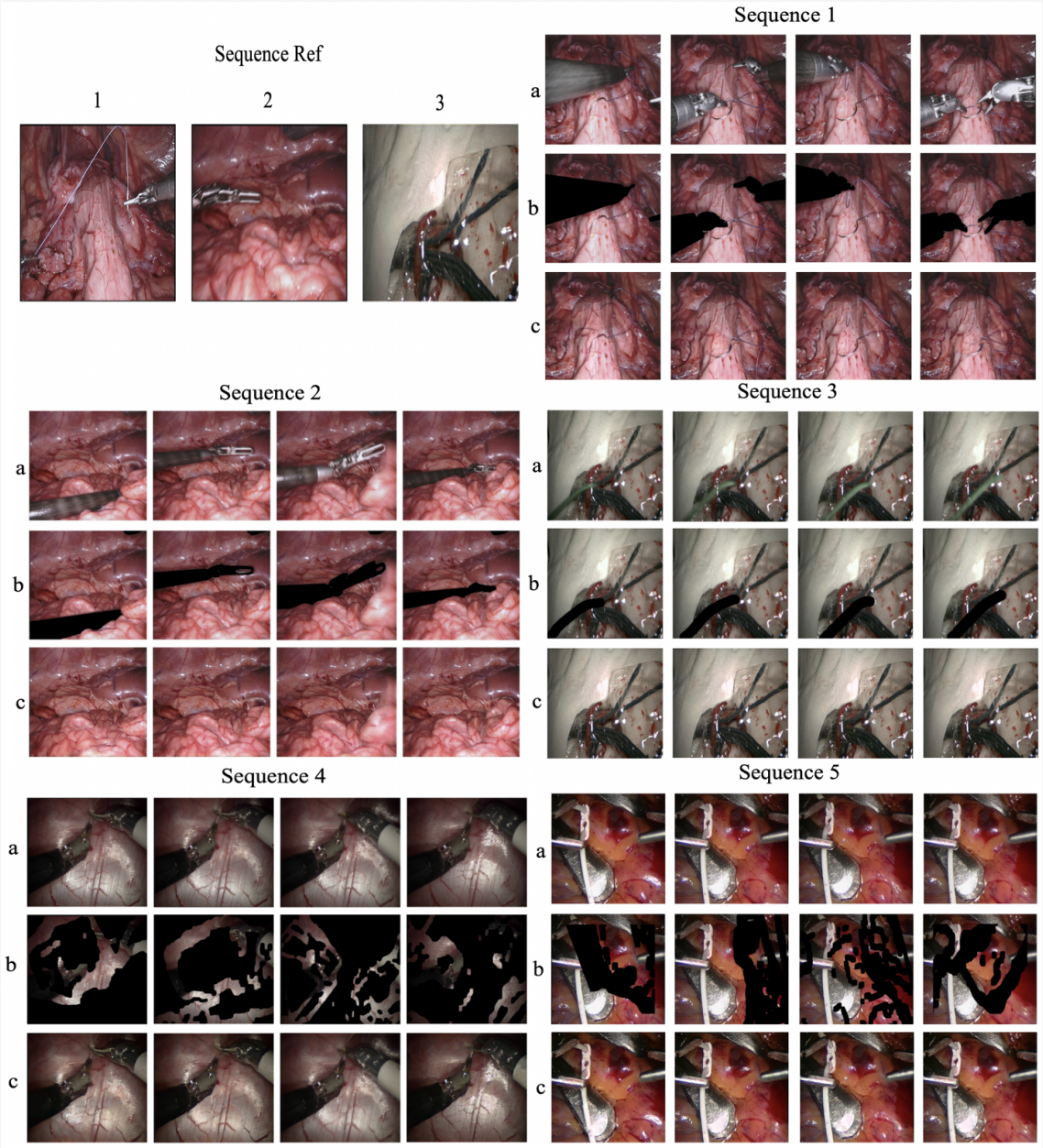


Figure 3.4: Occlusion reconstruction for MIS sequences where, rows **a** display the original candidate frame, **b** the input masked frame and **c** the reconstructed output. Sequence Ref shows the reference frames for Sequences 1,2 & 3. Sequences 4-5 were occluded by random synthetic occlusions on the entire video with OIRs ranging from 60-70% & 30-40% respectively.

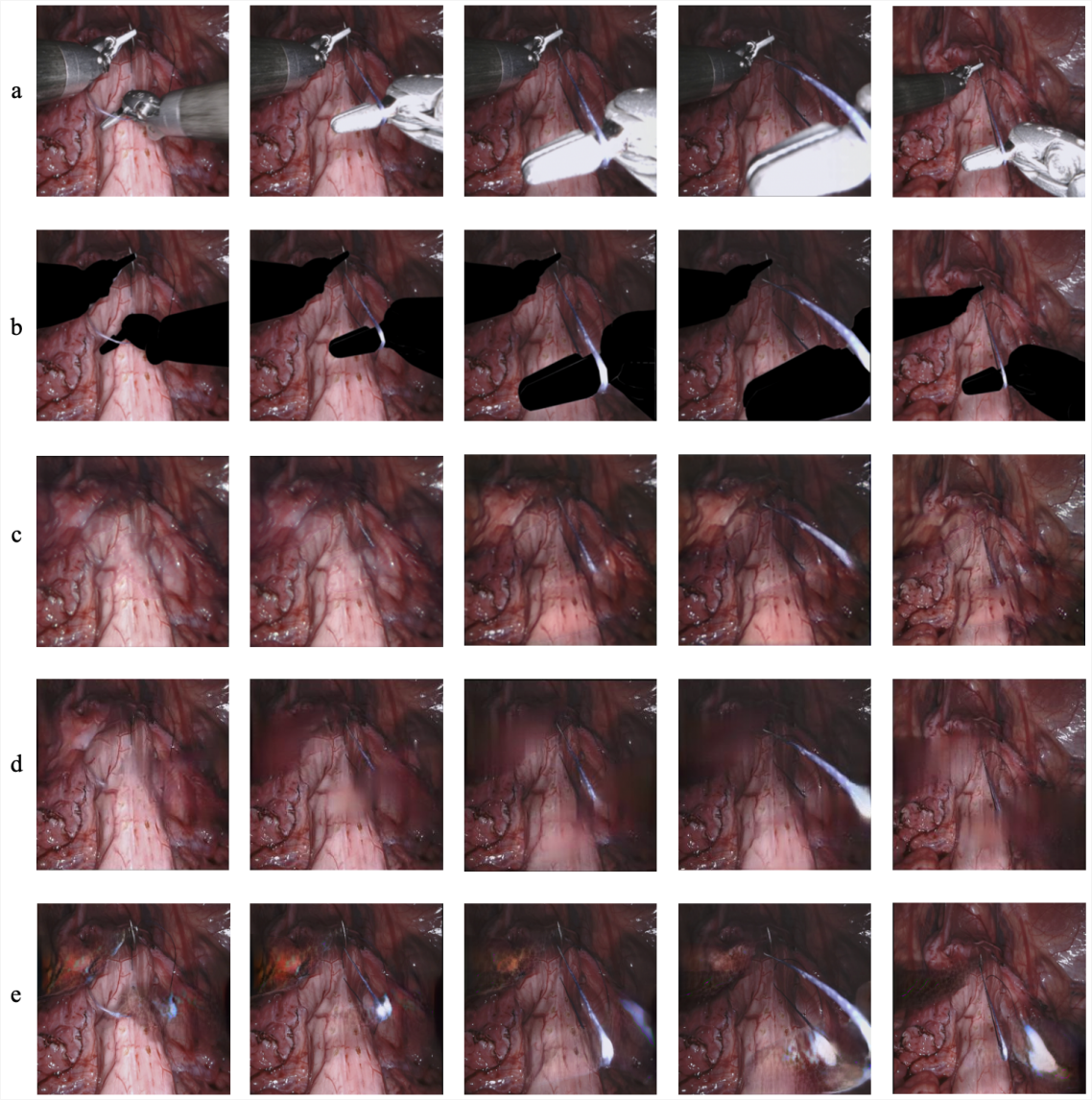


Figure 3.5: Comparison of qualitative results on a sample of consecutive frames from the EndoVis2018 data, presented in each column. **a** the original candidate frame, **b** the input masked frame and **c** the reconstructed output from the proposed model,**d** output from [123] and **e** output from [25].

outperforms the other deep video inpainting models by accurately regenerating the scene details hidden behind the occlusion. GateTemp generates a lot of artefacts when reconstructing the occlusion as it can be observed in row e. This is mainly due to the lack of movement in the tool in the previous frames. Furthermore, it also creates dark shadow elements within the reconstructed region. This shadow effect appears more severe in the frame at the last column where image scaling due to the camera's zoom-out affects significantly the occlusion recovery result. The CPNet approach in row d is significantly affected by the changing lighting conditions in the scene, resulting in shaded textures around the reconstructed regions. It can be noticed that this method is not affected by image scale changes which must be due to the alignment network learning the relevant context patches. In all cases, the perceptual and style losses incorporated in the proposed method allow the model to learn the distribution of the movement of the occlusions and the visible parts within the image. This results in the generation of high fidelity reconstructions which are robust to changes in lighting and image scaling.

The performance of the video inpainting methods has also been quantitatively compared on MIS data under varying OIR% based on the mean scores of the L1, PSNR, SSIM, IS and Temporal SSIM error as performance evaluation measures. The graphs in Fig. 3.6 show the superiority of the proposed method which achieves higher reconstruction accuracy and lower L1 error for the various OIR% on the tested MIS datasets. As it is expected, the general trend is that the reconstruction accuracy decreases as the OIR% increases. This is due to the lack of sufficient prior visible information available for reconstructing the occluded area which results in filling the output reconstruction with artefacts, similar to performing spatial image inpainting.

The contribution of the individual loss components in the final generator loss function L_G, has been assessed by training separate models with each component removed from the loss function during the training procedure. As it is shown in Fig. 3.7 and 3.8, the most important components are the style and perceptual losses as the performance of the model varies significantly when these components are not included in the loss function. The warp loss, aids in creating temporally coherent outputs while the quality of the generation is governed by the adversarial, style and perceptual loss components. In the Temporal SSIM error graph in Fig. 3.7, as the OIR% increases, the warping loss impact reduces. Since occlusions are large, the network prioritises, filling the occluded pixels with plausible values. Hence, relying more on the feature matching and adversarial losses. In the qualitative ablation study (Fig. 3.8), ablating the feature matching loses (d & e), results in artefacts being formed around the occluded area and the model having an incorrect colour gamut. The other losses in f and g aid in refining the output and artefact removal since more consistent results & textures are visible. The style and perceptual components allow the network to learn the implicit feature distribution similar to that of the ground truth and the adversarial loss enforces the network to generate high fidelity reconstructions. The addition of the warp loss and the adversarial loss result in a higher IS score, which can observe in Fig 3.7, as both these loses result in high quality of the images generated.

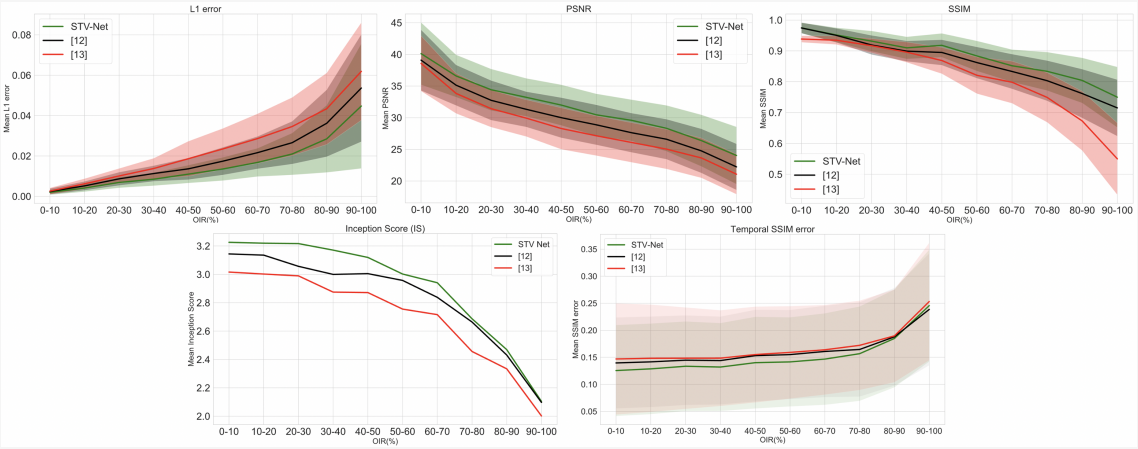


Figure 3.6: Performance evaluation of different occlusion recovery methods under varying OIR(%) on MIS data from the Hamlyn database.

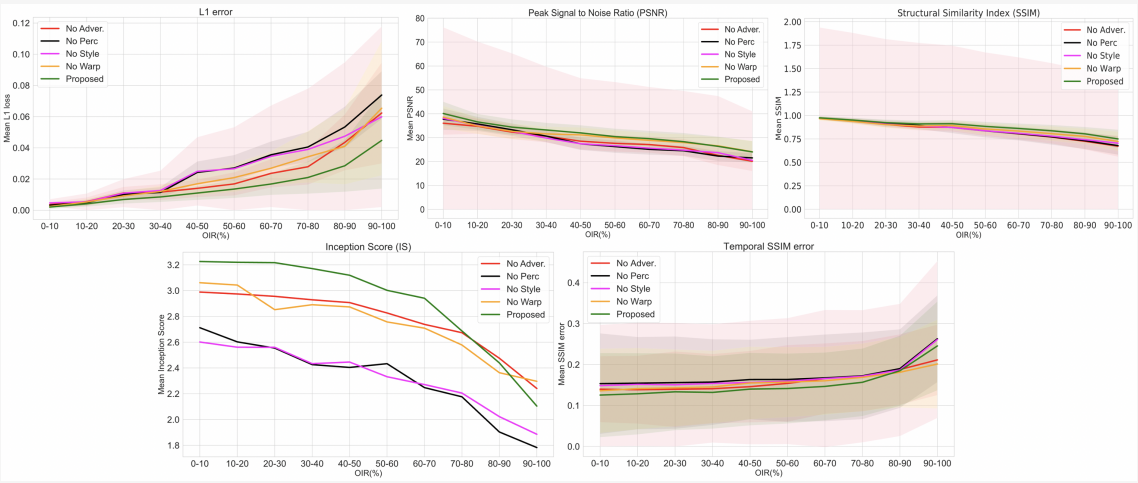


Figure 3.7: Loss Ablation results, comprising of L1 error (top left), PSNR (centre), SSIM (top right), IS score (bottom left) and Temporal SSIM error (bottom right). The results were attained from the hamlyn test dataset. The metrics shown were calculated between the generated output and the ground truth from the hamlyn test set. $-NoAdver.$ being the proposed model with no adversarial component, $-NoPerc.$ denotes no perceptual component, $-NoStyle$ denotes no style component, $-NoWarp$ denotes no warping loss component and finally, $-Proposed$ is the proposed model with the all the proposed loss components

Apart from visualising the reconstruction of the entire occluded area, other visualisations may be more preferable. For example, a surgeon may want to visualise the underlying structures that are obscured whilst being cognizant of the robotic tool in space. To this extent we develop a visualisation entitled the "**Glove Effect**" which removes the occlusions while it retains the contour of the original tool and superimposes the end effector onto it. As it is shown in Fig. 3.9, this visualisation provides the surgeon with a see-through vision while being aware of the position of surgical instrument in the scene and their interactions with the underlying tissue.

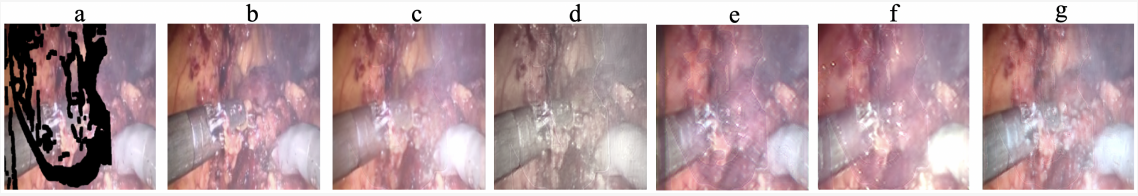


Figure 3.8: Qualitative results for different loss function combinations. (a) input masked image, (b) ground truth, (c) proposed loss, (d) no style loss, (e) no perceptual loss, (f) no warping loss, (g) no adversarial loss.

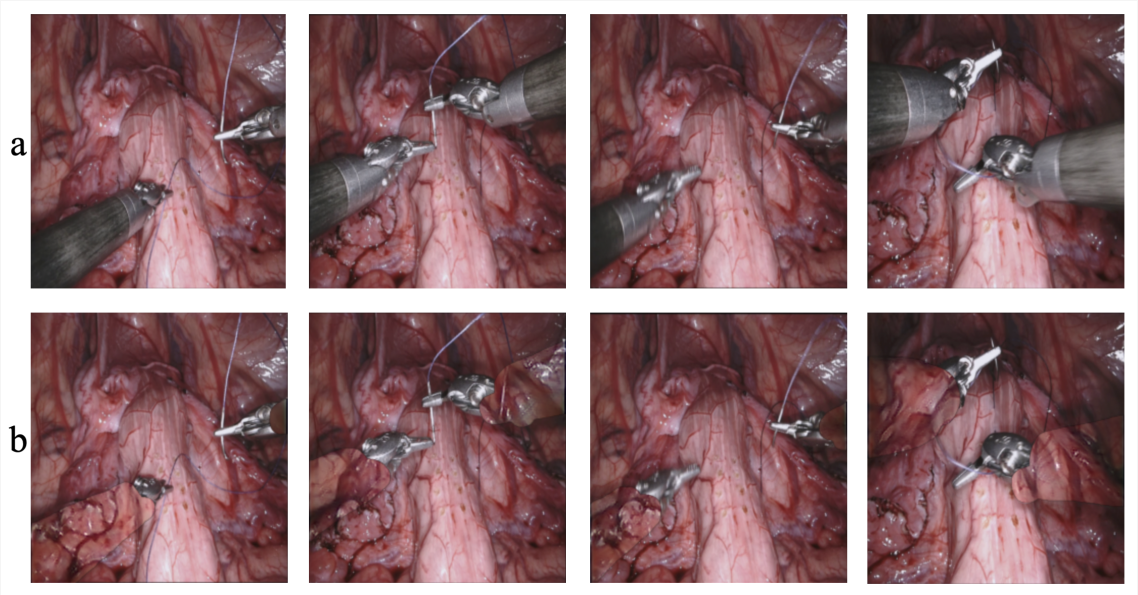


Figure 3.9: Glove Effect visualisation, where rows **a** display the original candidate frame and **b** the output of the glove effect, where the contours of the tool and the end effector are purposely visible.

3.5.3 Performance Evaluation on Natural Scenes

In this section, we also show our model can go beyond just surgical scenes and perform exceptionally on natural scenes too without requiring further training or fine tuning.The recovery of occlusions on natural scenes from the Davis2017 database is presented in Fig. 3.12. These are challenging sequences as in the "Bike" sequence, both the camera and the object move over time while in the "Dancer" the background where the occluded object is centred mainly in the same location throughout the video, providing little information from the past for reconstruction. No further training nor fine-tuning was performed. As it can be observed, although our model has been trained on MIS data only and was not fine tuned nor trained further on the DAVIS dataset, it successfully learns the essence of reconstruction. Furthermore, the performance of the compared video inpainting methods on the Davis2017 dataset for random masks and OIR of 30-40% is shown in Table. 3.1.



Figure 3.10: "Bike" sequence



Figure 3.11: "Dancer" sequence

Figure 3.12: Occlusion recovery, on natural scenes from the Davis 2017 dataset. The rows **a** display the original candidate frame, **b** the input masked frame and **c** the reconstructed output.

Table 3.1: Performance evaluation on random masks of OIR 30-40% on the Davis2017 test data. STV-Net, was trained only on MIS.

Method	L1	PSNR	SSIM	IS	Temp. Consis.
STV-Net	0.023 $\pm$0.015	24.62 $\pm$2.92	0.809 $\pm$0.045	9.12	0.131 $\pm$0.095
[123]	0.023 $\pm$0.009	24.48 $\pm$3.07	0.815 $\pm$0.059	13.49	0.125 $\pm$0.086
[25]	0.022 $\pm$0.011	22.05 $\pm$2.91	0.797 $\pm$0.062	11.23	0.134 $\pm$0.081

3.5.4 Discussion

The performance evaluation analysis shows that the proposed model, outperforms the latest video inpainting models on surgical data. This is due to the novel architecture which fuses spatio-temporal information by introducing the input data into 2 separate streams for the sequence of frames and optical flow and combining the features learnt with the guided optical flow maps in the decoder. The dense connections within the network allow the decoder to utilise all the information gathered from each layer to reconstruct and refine the output in a hierarchical manner. Furthermore, the combination of multiple losses in the proposed loss function has been tailored to generate high fidelity and accurate image reconstructions that are temporally consistent.

In our performance evaluation analysis we noticed that the performance of the proposed method is affected by abrupt camera motions producing artefacts in the occlusion reconstructions. This is due to lack of coherence in the temporal features extracted from the previous frames. Since the scene has shifted to a new setting, the prior information fed in is actually not useful for feature extraction. However, abrupt camera motion are rare in the proposed application of minimally invasive surgery. Additionally, we observed that if the occlusion occupies majority of the scene and is localised in mainly one region, without moving between frames, then the scene reconstruction generated includes colour saturation artefacts. This is due to the lack of temporal discrepancy in the previous frames, resulting in fewer regions that the model can use to learn from which, results in blurry synthesis. To deal with this, input frames at n interval can be considered as opposed to taking consecutive frames as input. In this case, n is a user defined interval window. This will allow consideration of frames with sufficient temporal distance where the occlusion has moved.

3.6 Conclusions

In this chapter, the STV-Net has been proposed which is an unsupervised end-to-end deep learning framework, based on Fully Convolutional Network (FCN) networks. The proposed model utilises 2D and 3D partial convolutions to combine information from the past and current frames in order to infer the appearance of the surgical scene under occlusions. The performance evaluation analysis has shown that the proposed model is superior to state-of-the-art video inpainting models. This is due to the novel architecture which fuses temporal and spatial information. The dense connections within the network allow the decoder to utilise all the information gathered from each layer to reconstruct and refine the output in a hierarchical manner. Furthermore, the proposed loss function has been tailored to generate high fidelity and accurate image reconstructions with temporal coherence. Combining multiple losses enables the network to capture vital features essential for reconstruction. This has resulted in a system robust to occlusions of various shapes, sizes, locations and orientations. STV-Net can further be combined with any segmentation network for occlusion mask generation, resulting in an end-to-end occlusion detection and removal framework. Our future work will focus on making our model real time for online occlusion removal. Additionally, we aim to extend this work to other challenging occlusions in surgery such as smoke and blood.

3.6.1 Learnings & Future Work

In this work, I employed generative modeling and perceptual learning techniques to develop an effective framework for processing and removing and reconstructing occlusions from videos. This experience provided me with valuable insights into designing task-specific loss functions, model architecture and embedding temporal dynamics within Deep Learning (DL) models. Additionally, my work on this chapter emphasized the importance of embedding temporal dynamics in DL video models. By incorporating temporal information through 3D partial-convolution intrinsically and via optical flow extrinsically, the model was able to capture the dependencies between consecutive frames and maintain coherence across time. This understanding of temporal consistency proved instrumental in Chapter 7, where I applied similar techniques to obtain temporally consistent depth maps addressing problems #3 and #4 (outlined in Introduction).

The following are the key limitations of the current framework where I have suggestions for improvements:

1. Input resolution: DL evolves rapidly; the average lifetime of a model is merely five months now, before new advancements emerge. Since the completion of this work, other studies have introduced innovative modules for handling high-resolution inputs on single image inpainting i.e. [205]. One obvious step forward would be to replace standard convolution layers with

Fourier convolution as shown in [205] adapted for sequences (i.e. combined with self-attention or convolutional-Long Short-Term Memory (LSTM)s. This may improve inpainting fidelity and make the model light weight such that it can generate outputs at greater resolution.

2. Learn scene-flow explicitly via warping only: In this work I primarily used FlowNet to account for optical flow. However since completing this work, other models with greater optical-flow accuracy and performance have been introduced [212][87]. Simply, replacing FlowNet with these models for both inputs and warping loss will significantly improve results as these models will estimate occlusion deformation with greater precision. Furthermore, one may not even need to feed in optical flow as input into the model. Explicitly enforcing flow constraints as a loss function (as done in Eq. 3.12) may be sufficient, as shown by this video inpainting paper [54]. This enables new model architecture designs to be explored which are potentially light weight, easy to deploy and thereby focusing more on input resolution issue.

3. Model architecture: 3D convolution are computational resource demanding when it comes to training. Combining potential work from 1 and 2, may result in $stream_2$ being redundant, which will lower the computational complexity of the model. However, finding alternatives to 3D convolution may further reduce model complexity which can boost runtime performance. Perhaps exploring architectures based on 2D convolution combined with LSTM [194] or going for a full video transformer [11] etc.

Chapter 4

Randomly connected neural networks for self-supervised monocular depth estimation

The subsequent challenge in my PhD involved attaining sub-millimeter depth accuracy (problem #2 outlined in Introduction), a vital aspect for ensuring surgical precision and preserving patient safety. To understand training a Deep Learning (DL) model for depth estimation, I initiated with monocular depth. Handling single-image inputs lessened computational complexity, streamlined model design and training strategy investigations, providing an excellent starting point. This chapter's work also benefits clinical cases like laparoscopic surgeries with monocular views [197]. Lack of stereo vision hinders depth perception and causes surgeon fatigue. Skilled surgeons utilise monocular cues like lighting and object size to aid hand-eye coordination. Currently, no enhanced Computer Vision (CV) system addresses this issue. This is due to the following challenges: (1) lack of dense ground truth data for surgical settings, restricting the training DL models, and (2) bias in existing models towards natural scenes which demonstrate poor performance in surgical environments even after transfer learning due to distribution drift. Building on the work done in chapter 3; leveraging the power of perceptual learning and generative modelling, I addressed (1) by creating a new loss function for for learning generalisable feature representations. To tackle (2), I proposed a novel architecture; a randomly connected Convolutional Neural Network (CNN) that is unbiased in its design and has the flexibility to learn the most appropriate weights and explore effective design topologies. This method achieved state-of-the-art (SOTA) performance in monocular depth estimation on both surgical and natural scenes and even surpassing some stereo models. The work of this chapter became the foundation for chapter 5 where I increased the complexity of the problem and solution to cater for stereo inputs.

The work in this chapter was published in the journal of Computer Methods in Biomechanics and Biomedical Engineering: Imaging Visualization in 2021 [216]. It also won the best paper award in the 15th Augmented Environments for Computer Assisted Interventions (AE-CAI) MICCAI 2021 workshop.

4.1 Overview

A plethora of deep learning-based methods have been proposed for self-supervised monocular depth estimation. The majority of these models utilise a U-Net based architecture for disparity estimation. However, this architecture may not be optimal, and as shown in self-supervised approaches, replacing standard U-Net encoder with more complex architectures like the ResNet encoder achieves superior performance. In monocular depth estimation, the success of the method is attributed to the model architecture design capable of extracting high order features from a single image and the loss function for enforcing a strong feature distribution. In this work, a novel randomly connected encoder-decoder architecture has been designed for self-supervised monocular depth estimation. To enable efficient search in the connection space, the 'cascaded random search' approach is introduced for the generation of random network architectures. We introduce a new U-Net topology capable of utilising the semantic information of feature maps. For high quality image reconstructions, a loss function has been proposed which efficiently extends perceptual and adversarial loss to multiple scales. We conduct performance evaluation on 2 surgical datasets, including comparisons to state-of-the-art self-supervised depth estimation methods. The performance evaluation analysis verifies the superiority of our randomly connected network architecture.

Brief video explaining the work: 13 min video

4.2 Introduction

Depth estimation is one of the fundamental problems in computer vision constituting of predicting the distance of each point in a captured scene relative to the camera. Depth is essential in a divergent range of applications including robotic vision, surgical navigation and mixed reality. Although humans perceive depth relatively easily, the very same task for a machine has proven to be quite challenging. Depth estimation involves the detection and matching of salient features between multiple views of a scene. Recently, deep learning models have become very attractive in the depth perception workflow, providing end-to-end solutions for depth and disparity estimation [127]. The learning in these methods can be either supervised [53] [46] [245], or self-supervised [63], [241], [57].

The best performing models tend to be trained via supervised methods since they directly learn to regress the disparity values. However, they require ground truth which is very arduous to capture and unfeasible to re-collect every time the dynamics of a scene change and re-train a model. Additionally, transfer learning also fails when target data does not match the training [251]. This hinders their application in surgical scenes where data is limited and scenes are substantially different from the natural scenes counterpart. Hence, self-supervised methods are preferred, since no ground truth is required for training. Instead, they are trained by conducting novel view synthesis and optimising for photometric re-projection error [63]. However this, error can be optimised even if the geometry is inconsistent (i.e. incorrect disparity values in low textured areas). Thus, requiring stronger view synthesis loss functions that can penalise inconsistencies in such regions.

In monocular depth estimation, most self-supervised methods [64][164][235] adopt the U-Net based encoder-decoder architecture DispNet [143] which generates a high resolution photometric output. However, this may not be the most optimal model. Hence many approaches, replace the standard U-Net encoder with more complex architectures like the ResNet encoder achieves superior performance. In the literature, very little is done to explore architecture design in self-supervised monocular methods. Since, monocular settings does not involve inference from multiple views, the success of deep learning methods is therefore, attributed to the network architecture design and its ability to extract meaning-full features. In addition to architecture design, a strong learning signal must be provided for the salient features to be encoded. Hence the corresponding loss functions used for optimisation, need to be engineered accordingly.

To improve self-supervised monocular depth estimation approaches, it is imperative to engineer both the architecture components and the loss function carefully to enable strong features to be learnt, regardless of the data type. The aim of this paper is to design a bespoke model and loss function for self-supervised monocular depth estimation, capable of learning strong features advancing the standard *DispNet* architecture used widely.

4.2.1 Contributions

Our main contributions are the following:

1. We design a novel architecture for dense monocular disparity estimation using random neuron connections, further augmented with attention for feature extraction. To the best of our knowledge, we design the first randomly connected encoder-decoder architecture

2. We introduce the 'cascaded random search' approach for generation of randomly connected architectures which enables better search in the connection space

3. We introduce a new U-Net topology to improve the expressive power of skip connection feature maps both spatially and semantically

4. We design a loss function which efficiently extends deep feature perceptual and adversarial losses to multiple scales for high fidelity view synthesis and error calculation

The proposed monocular model is trained in a self-supervised manner, using stereo data similar to [63] which we use as our base method for comparison. We didn't use the more recent [64] which train on monocular videos as our base, because of its inability to be trained on scenes with static cameras or small camera movements. Which is prevalent in the surgical domain thereby deteriorating performance when trained on surgical datasets. Our study proves that a randomly connected network architecture combined with a well designed loss function can outperform the state-of-the-art. The performance evaluation was conducted on Minimally Invasive Surgery (MIS) data, Hamlyn data [61] and the SCARED stereo benchmark [6]. The proposed method can be easily extended to stereo disparity estimation and employed for any image-to-image translation task.

4.3 Related Works

In literature, approaches for depth estimation fall into two categories namely, supervised which require ground truth data and self-supervised which do not rely on any ground truth. In **supervised** depth estimation, dense depth maps have been produced via a 2 stage process, where in the first stage a deep neural network produces a coarse depth map followed by the second stage where another deep neural network refines it [46]. The authors treated the problem as a regression task and defined losses that were invariant to image scale. Other methods were later released that built upon [46], such as [245] which treated the problem as a classification task of the valid pixels in the ground truth depth maps. Furthermore, [53] treated it as a regression problem, and introduced a novel type of loss based on ordinal regression. Supervised approaches depend upon high quality, ground truth depth during training, which is subject to availability. Lastly, since they learn to directly regress depth values, if the range of depth changes they need to be re-trained. Given the range of depth in surgery (mm) is substantially different from natural scenes (m), their adaptability is hindered and transfer learning will have limited effect [251].

Whereas **self-supervised** methods do not require this. Their learning focuses on the reconstruction of images based on warping of the input image with the output predicted disparity map. The methods in [241] and [56] both utilise this image reconstruction based losses to train their deep learning models for 3D reconstruction. In [56] the reconstruction loss is not fully differentiable, hence Taylor approximation is performed to linearise the loss with respect to the model which adds further complexity to the method. In [63], bilinear sampling is performed, similar to that of [92] thus resulting in a differentiable loss. The loss components described in [63] have become somewhat standard in self-supervised depth estimation. [167] further extended [63]'s work to trinocular based depth estimation, further improving the performance. [166] experimented with cycle adversarial loss to enhance the photometric error. However the photometric loss can be optimised for multiple range of disparity output values, and such adversarial losses are unable to penalise for artefacts formed in edges of the generated images.

Furthermore, these self-supervised methods are all based on the *DispNet* architecture. [63] showed replacing the encoder parts with ResNet- 18 & 50 [75] models achieved better performance. This is attributed to (1) their innovative method of (residual) network connections (similar to why other methods like DenseNets [84] are successfull). (2) The pre-trained Imagenet feature distribution which leads to stronger optimisation. (2) is applicable to natural scenes, but limited on surgical. Hence, upon evaluating (1) further, we wanted to optimise encoder design for having more innovative connections. Thereby improving the expressive power of the model for feature extraction. Hence, to remove limitations on the network connection space, we explore randomly connecting the convolution operations in a deep learning model. Inspired by [242], they utilised traditional

random graph generators from graph theory i.e. Watts Strogatz (WS) algorithm [236], to define randomly connected networks, which they then convert to neural networks for classification training. However, [242] work was limited to classification architectures and they limit their connection space to only have a single random graph topology to build the full model. We extend their work to generate randomly connected encoder-decoder architectures with varying topologies, that can be utilised for any image-to-image translation task.

In this work we are making the hypothesis that a random search on the network's connection space could yield ideal network candidates and achieve comparable results to the state-of-the-art. This is because, the connections between operations are simply the weight values which are updated during the training process via gradient descent optimisation. Thus, performing an optimisation step i.e. gradient descent on connections that were randomly searched, will narrow down the architecture search space automatically. Since, training them will still yield ideal weight values.

4.4 Methods

The proposed disparity estimation model follows that of an encoder-decoder structure and takes a single image as input. The encoder is a randomly connected fully convolutional neural network. It comprises of convolutional blocks that perform down-sampling of the input data and the decoder part performs up-sampling of the encoder feature maps to generate the left and right disparity maps hierarchically. Furthermore, we introduce skip connections with learn-able weights for improving the expressive power of the encoder. The overall architecture is shown in Fig. 4.1. For training, we experiment with the loss function introduced in [63] as our base loss function, and propose a multi-scale adversarial image generation loss function.

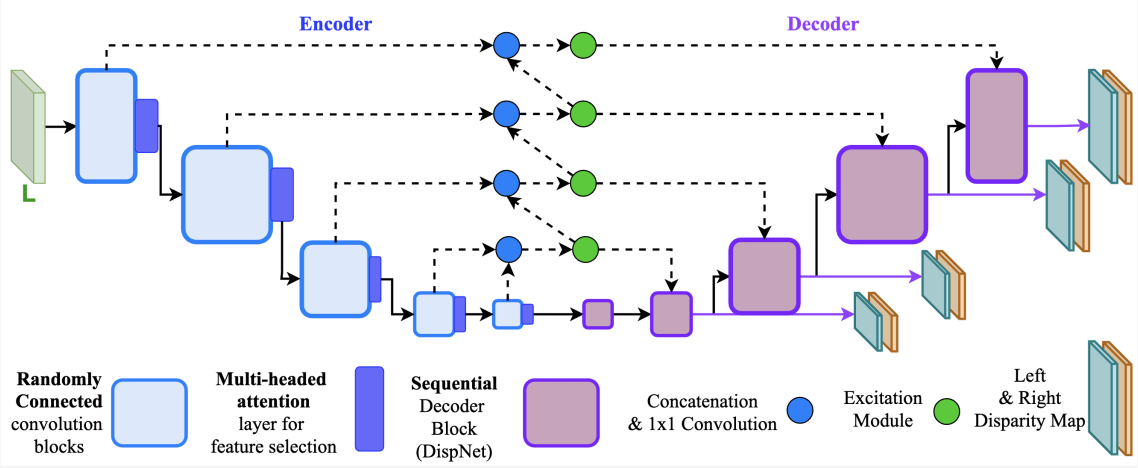


Figure 4.1: The proposed encoder-decoder architecture. The solid and the dotted lines denote forward propagation and skip connections respectively. The purple lines signify the output left & right disparity maps generated at 4 scales, each increasing hierarchically with a scale factor of 2.

4.4.1 Monocular Depth Estimation Model

Randomly Connected Encoder: The encoder is initiated with a random graph which is generated once in the beginning and subsequently converted to a Directed Acyclic Graph (DAG). We modify [242] to enable development of randomly connected encoder-decoder architectures, for depth estimation. To define a randomly connected encoder-decoder architecture, we first divide the network into blocks as shown in Fig. 4.1. This allows down-sampling to take place and provides more control on the size of the feature maps produced. Each block comprises of its own random connections with n number of nodes (i.e. operations) within it. The nodes within this graph are then mapped to their respective operations. The operations in the encoder layer are defined by a single convolution layer of a user defined filter size, followed by normalisation and activation. The activation operation we use is eLU [35]. The process of random graph generation as illustrated in Fig. 4.2, is the following:

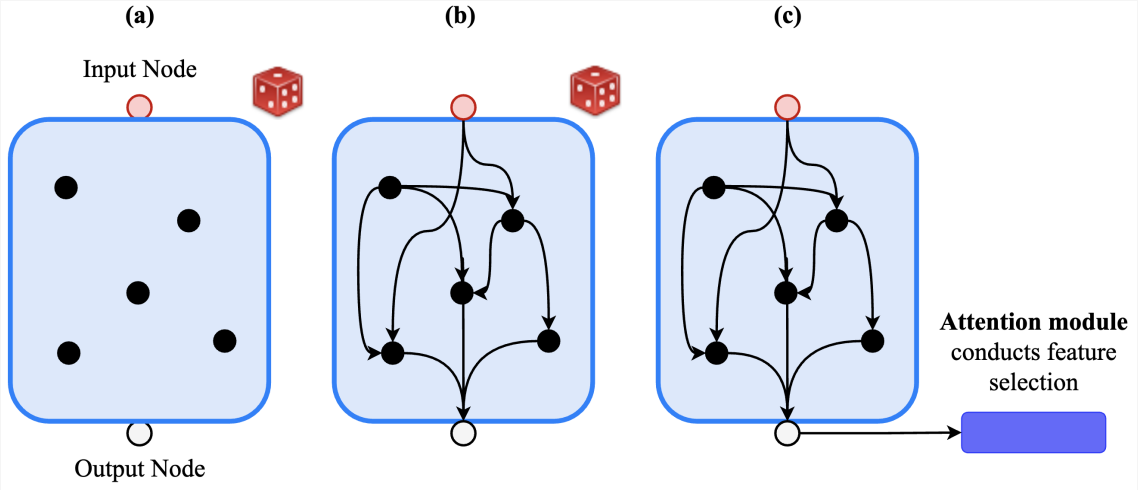


Figure 4.2: The process of generating a randomly connected deep learning architecture. The red and the grey nodes are the input and the output nodes of the block, respectively. The black nodes are the convolution operations. The purple module is a multi-head self-attention module.

- Step a - Generate a list of n nodes in a block and two fixed nodes namely, an input and an output node. The two fixed nodes are placed in addition to the n nodes within the block to allow inflow and outflow of feature maps;
- Step b - Using the WS algorithm, randomly connect the nodes in the block;
- Step c - Convert the nodes in the block to their respective operations;
- Step d - On the aggregated feature maps from the output node, utilise multi-head self-attention for feature selection;

Once these blocks have been initiated they are simply used as modules in the overall architecture as shown in Fig. 4.1. In all cases, if there are more than one connections as input into a node, then the incoming feature maps are aggregated via a weighted sum with learn-able weights and then fed into their respective operations. If the output is to be sent to multiple nodes, then copies are made and sent. Since, the feature maps are randomly connected, there is potentially misalignment between them when aggregated. Hence, we augment this backbone by using a multi-head, self-attention mechanism [220] at each encoder block. In particular, we use efficient attention with linear complexity as described in [192]. These, attention mechanisms enable convolution layers to focus on key aspects via knowledge distillation, by learning what to pay attention to. This is also vital for the skip connections, where this feature map misalignment can lead to poor localisation of features for disparity generation.

Furthermore, to generate connections between nodes, a seed is used to initialise a pseudo-random number generator. The pseudo-random number generator, is used to generate up to k pairs of ran-

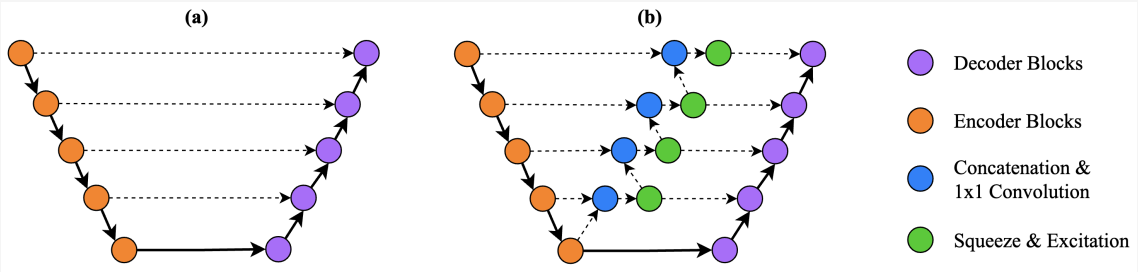


Figure 4.3: Showing the proposed learn-able skip connections methodology. a is the network topology of a standard U-Net and b is the proposed new network topology. Solid lines and dotted lines denote forward propagation and skip connections respectively.

dom numbers for each node. k is the maximum number of nearest neighbour nodes each node can connect to and these pairs define the index of the other nodes that the candidate node will connect to. The process of graph generation, as shown in Fig 4.2, occurs in steps a and b which are highly susceptible to the random seeds selected, for initialising this pseudo-random generator. If only one seed value is selected (i.e. a global seed), then a single graph will be generated which will be copied in every random block in step d. A global seed will initialise the same set of random number pairs every time which will constrain the search space of connections. Another contribution of our work is the 'cascaded random search' process which generates a large exploration space for connections. We modify the method in [242], by introducing multiple seeds by multiplying the global seed with a multiplier which in our case is the block number assigned sequentially. This allows each random block to have its own unique random seed. For example, if the global seed is 2, then for the model in Fig. 4.1 and the first encoder block, the seed is 4 (i.e. 2x2, global seed value $\times$ layer number), for the second encoder block, the seed is 6 (2x3), for the third encoder block, the seed is 8 (2x4) and so on. This allows divergent graphs with unique random connections to be formed at each random cell enabling the exploration space to be large. We term this process 'cascaded random search'. The multiplier can be any set of arbitrary numbers such that upon multiplying with the global seed value, a new seed value is generated for each block.

Learn-able Skip Connections: For the decoder to produce sharp disparity maps, it requires strong spatial and semantic features. In our case, semantic features are deep relationships hidden in surgical scenes, such as different tissue types and objects. This is typically stored in the channels of the feature maps. Spatial features help in localising boundary information, and this is fed to the decoder via skip connections as shown in Fig. 4.3a. However, not much attention is given to semantic features. [266] showed that greater semantic feature representations are developed deeper through the network. For this reason, they created a densely connected U-Net, where every encoder layer feature map is connected to every ascending decoder layer. However, this explodes the parameter count and substantially increases training and inference times. Conversely, we take a more efficient approach to preserving these semantic features. Instead of densely connecting each

encoder feature map, we concatenate the previous feature map with the adjacent one and perform 1x1 convolution for weighted compression. This prevents the parameter explosion. This is followed by squeeze & excitation modules [83] which individually weights these channels in importance (i.e. channel-wise attention) for the following convolution layer in the decoder. These weighted semantic features are then passed on to every ascending layers hierarchically. Allowing the model to maintain semantic features throughout each layer and to enable weighted aggregation of them for feature selection as shown in Fig. 4.3*b*.

Decoder: Our decoder advances the architecture design proposed in [143]. Unlike [143] which uses bi-linear interpolation for up-scaling the feature maps in the decoder, we utilise the Pixel Shuffle operation [193] followed by a convolution layer (filter size 3x3). As stated above, semantic features are usually overlooked and interpolation up-scales feature maps by linearly interpolating the spatial features. Instead we reshuffle the deep channel feature maps to convert them into new spatial pixels and perform convolution for inferring these semantically up-sampled features into disparities. Fig. 4.1, shows the final proposed model architecture, with all the components together where the encoder is randomly connected and the decoder is sequential as defined above. We also experimented with other model variants such as a fully random architecture where both the encoder and decoder blocks comprise of randomly connected graphs and a hybrid model, where the encoder layers are sequential (from Resnet18 model [75]) and the decoder layers comprise of randomly connected blocks. In our experiments the proposed model performed the best, as the sequential decoder does not suffer from feature map misalignment like the random ones, which results in blurry disparities.

4.4.2 Loss Function Formulation

Our model is trained in a self-supervised manner using stereo video sequences, without requiring any ground truth disparities, similar to [63]. Given the left stereo-rectified image I_L^{GT}, the model produces the predicted left and right disparity maps D_L^{pred} and D_R^{pred}. Since the disparity map defines the displacement of pixels from one camera frame to its corresponding stereo frame, a warping function can be applied to these disparities and the opposite images to regenerate the corresponding stereo frames I_R^{Pred} and I_L^{Pred}. The intuition is that the generation of one image from the other in a stereo pair enforces the network to learn some aspect of the 3D shape of the scene. Hence, even with monocular input, the network intrinsically learns to develop both disparity maps. The loss function thus, focuses on the reconstruction error between the predicted and the original images which is used as ground truth. Standard loss components which have been used for self-supervised disparity regression include the image reconstruction loss L_{recon}, the disparity smoothness loss L_{smooth} and the left-right disparity consistency loss L_{LRcons} [63] as shown below:

$$L_{ssim} = \delta_1 L_{recon} + \delta_2 L_{smooth} + \delta_3 L_{LRcons} \tag{4.1}$$

The above loss terms are performed on both the generated left and right images. As stated above, the image reconstruction loss is crucial for achieving high performance in self-supervised depth estimation methods. [63] in particular uses the Multi-Scale Structural Similarity Index (MS-SSIM) [231] jointly with a L1 loss term as their L_{recon}. In our framework, since our model outputs disparity maps at 4 different scales, the MS-SSIM loss enables the model to hone the disparity output gradually as size increases. Additionally, it enables the reduction of artefacts being formed at the final output (at the largest scale).

Adversarial loss based on Generative Adversarial Networks (GANs), have been employed for depth estimation for computing the photometric error [166][5]. To compute this loss, a generator G and a discriminator D is required. G is the proposed model that generates the disparity maps and thereby the predicted images via warping. D is a binary classifier that differentiates between ground truth images and the predicted images. The aim is to train G and D both to the point that D can no longer differentiate between ground truth and the predicted images generated by G. This formulation doesn't account for the multi-scale penalty. The multi-scale penalty is vital, since standard adversarial losses tend not to specifically penalise for artifacts in the synthesized image i.e. on the border regions or on the edges of objects where the depths are always not sharp enough to generate realistic warp colors. Adversarial loss alone fails to tackle this issue.

In this work, in addition to MS-SSIM for L_{recon}, we propose a novel loss function based on multi-scale adversarial learning and perceptual loss. Our loss function allows for penalising poor quality generations at their respective scales. The naive approach to achieving this, would involve having a discriminator D dedicated to each scale. However, this would increase computation time and memory, requiring 5 networks being trained in parallel (G and 4 Ds, one for each scale). This also has the high risk of loss explosion, leading to G potentially not converging, or in some cases diverging. Instead, we define one D network inspired by [100], and unlike other methods, where D is some arbitrary sequential convolutional network, in our work, D is similar to the specification of the G encoder. Furthermore, we concatenate the output at each scale with the intermediate layers of D, enabling each block to learn features dedicated for the input at its respective scale. Illustrated in Fig. 4.4.

The multi-scale GANs loss for both G and D we experiment with is Least Squares GANs loss [140] (though any other type of GANs loss can be used):

$$L_{adv}^{D} = \frac{1}{2}[(D(I^{GT}) - 1)^2] + [(D(G(I_L^{GT})))^2] \tag{4.2}$$

$$L_{adv}^{G} = \frac{1}{2}[(D(G(I_L^{GT}))) - 1)^2] \tag{4.3}$$

D takes both the left and right images concatenated as input. I is the list of concatenated images for each of the 4 scales. The largest scale is parsed in via forward propagation, where the subsequent scales are concatenated with the forward propagating feature map before being fed into

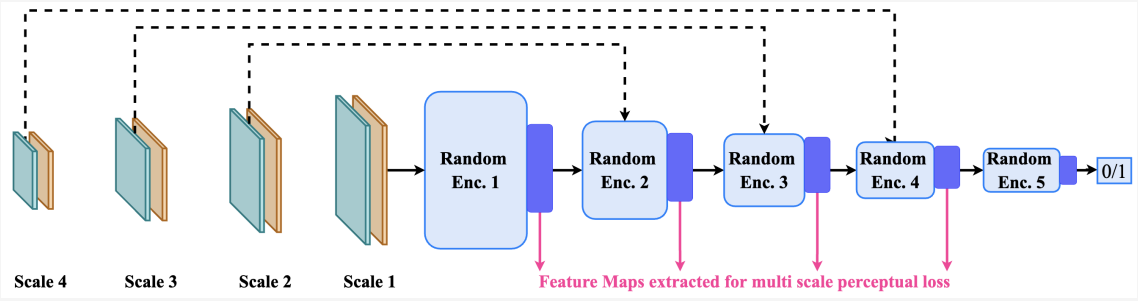


Figure 4.4: Discriminator model. The solid black lines and the dotted lines denote forward propagation and skip connections respectively. The skip connection inputs are concatenated with the forward propagating feature map prior to being processed by the next layer. The pink lines show the extraction of multi scale feature maps.

their respective intermediate layers. This enables the intermediate layers to learn the features dedicated for differentiating real and fake data for that particular scale. Furthermore, [94] showed, high fidelity images can be constructed through perceptual loss functions. It encourages the predicted outputs to have similar feature representations as the ground truth using features extracted from pre-trained networks. We adopt this to a multi scale feature reconstruction loss, by directly using our multi-scale discriminator as the feature extractor. This allows the feature matching to occur at each scale. The discriminator block with large receptive fields i.e. $Enc1$ learns global features such that G generates globally consistent outputs at the largest scale. Whilst, the blocks with smaller receptive fields i.e. $Enc3$ learn local features, ensuring G produces sharper details. This combination enables G to generate outputs with less artefacts, especially in low-textured scenes prevalent in MIS data. We experiment with the feature reconstruction loss (L_{feat}), equation displayed below:

$$L_{feat} = \Sigma_{n=1}^{N=4} ||D_n(D_{n-1}, I_n^{pred}) - D_n(D_{n-1}, I_n^{GT})||_1 \tag{4.4}$$

Where, n is the scale and for the discriminator it's the respective layer number too. Like above, I are the list of images at each of the 4 scales. The final loss function is:

$$\begin{aligned} L_{final} = \delta_1 L_{recon} + \delta_2 L_{smooth} + \\ \delta_3 L_{LRcons} + \delta_4 L_{adv}^G + \delta_5 L_{feat} \end{aligned} \tag{4.5}$$

4.4.3 Implementation Details

The square filter sizes of the convolution operations in the encoder blocks sequentially are [7,5,3,3,3]. The number of blocks and the filter sizes are consistent with the model in [63], to allow fair comparison. With the exception, standard [63] model, comprises of 7 layers of the encoder whereas ours is smaller comprising of 5 layers. The proposed model is trained to for 200 epochs. The input and output image resolutions are 256x512. The loss weightings in Eq. 4.5 are as follows: δ_1,δ_2, δ_3,δ_4 and δ_5 are 0.85, 1.0, 1.0, 1.0 and 0.05 respectively. For the graph generation algorithm WS, the

maximum k nearest neighbours for node selection was 4, the total number of nodes n was 5, and the probability of rewiring an edge was 0.75. For training with L_{feat} the loss calculation starts after 10 epochs, to give the discriminator enough time to become a good classifier. Both the proposed disparity estimation model and the discriminator are trained with the Adaptive Moment Estimation (ADAM) optimizer with a learning rate of 0.0001. The model was trained on MIS data from the *Hamlyn* database [61] where, 34,240 frames were used for training and 7,191 for testing. The models pre-trained on the *Hamlyn* data, were later fine-tuned on *SCARED* [6] for further evaluation. The models were trained on an Nvidia Tesla T4. The training computation time for the proposed model trained with L_{final} is 3 GPU days, whereas training with L_{ssim} is 1.5 GPU days.

4.5 Results & Discussion

4.5.1 Performance Evaluation

For *SCARED* data, we use the same evaluation metrics as defined in their paper [6], which is the mean absolute depth error (in mm) on test datasets 1 and 2. We show comparison with other self-supervised methods submitted to this challenge for both the test datasets presented in the challenge. For *Hamlyn* data, due to lack of ground truth depth, we compare the Structural Similarity Index (SSIM) between the ground truth and the predicted images followed by warping, same as [249] which is used for comparison. In both cases, the original methods proposed by the authors were stereo models whereas our method is a monocular model. Due to lack of availability of monocular models in surgical data we continue with comparison to stereo models on these benchmarks. Furthermore, the performance of the proposed method is qualitatively compared on both the datasets in Fig. 4.5. We also re-train the state of the art monocular models trained via stereo images i.e. [63] and [166] on the *Hamlyn* dataset to enable comparison with monocular models We show qualitative comparison of the monocular models on *Hamlyn* in Fig. 4.6.

4.5.2 MIS Evaluation

As it can be observed from Fig. 4.5, in both the datasets, the proposed model variant with learnable-skip connections (LS) (column 3) generates far less artefacts than the model without (no LS) (column 2). This is further visible on the depth maps in row $B.1$, where the top left corner of the depth map generated by our model with no LS, has a more prominent artefact than the variant with LS. Furthermore, LS also enables development of sharper depth maps, more consistent with the shapes appearing in the scene. This can be observed when comparing both the samples in row A, where the LS variant generates more consistent contours around boundaries of the soft tissues. Training with L_{Final} further enhances the results, by removal of artefacts around edges and boundaries of tools and soft tissues. This is visible in all the depth maps generated by the final proposed model (column 4) for all the samples. We believe the multi-scale adversarial loss is capable of penalising for these corner artefacts that are usually tricky for MS-SSIM. Whilst generating sharp, yet smooth depth maps. Row $A.2$, is particularly a tough challenge for deep learning models due to the reflections present in the scene. As it can be observed, both model variants with and without LS fail to generate sharp depth maps. However, the model variant trained with L_{Final} resolves a high fidelity depth robust to such effects. Showing the importance of multi-scale deep feature losses for photometric error calculation. The superiority of our model and loss function is further displayed in Fig. 4.6, where it is compared against 2 state-of-the-art monodepth models. The final column 'Focus area' highlights in green boxes where the generated depth maps by our model, achieves finer details than the competition. Exhibiting the importance of both the architecture design and multi-scale penalty for adversarial learning, in generating such sharper details. One

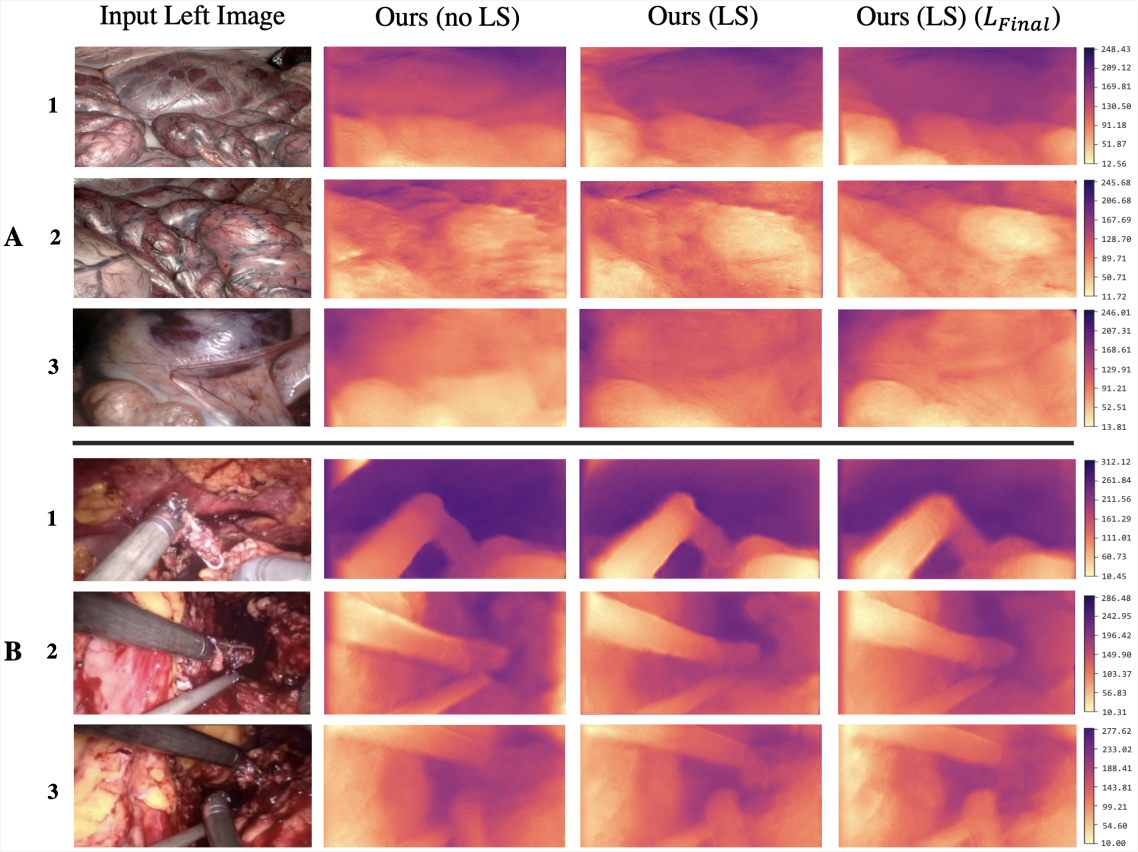


Figure 4.5: Depth maps generated by the proposed model on the *SCARED* and *Hamlyn* test splits shown in rows A and B, respectively. Inclusion of LS signifies presence of learn-able skip connections and $noLS$ without. all models were trained with the standard L_{ssim} loss (Eq. 4.1) unless L_{Final} is specified, which signifies training with the proposed loss function (Eq. 4.5).The colour bar displays the depth range in mm.

can observe that our model, generates sharper depth for anatomical structures present around the tools, which is a challenge for the other 2 methods. Moreover, the image in row 3 includes smoke which occludes some of the soft anatomical structures in the background. As it can be deduced, our model uncovers depth for such regions too. lastly, in row 2 our method generates less artefacts in corner regions of the image, as the multi-scale deep feature loss enforces the model to maintain global and local consistency.

The proposed model is also quantitatively validated and compared to the latest surgical depth estimation benchmark on the *SCARED* datsaset in Tables 4.1 and 4.2. We compare with the self-supervised submissions for a fair comparison albeit, the competing methods are stereo models. Overall the best performing model in test dataset 1 is ours, and despite being monocular our method is comparable to the other stereo models in test dataset 2. We believe this is because, both the stereo methods *'KeXue Fu'* and *'Xiaohong Li'* [6] follow the loss function and model introduced by [63] and [166] though, adapted for stereo. As shown in Fig 4.6, they are both prone to generating artefacts especially near edges and boundaries of soft tissue structures. This is due to both the

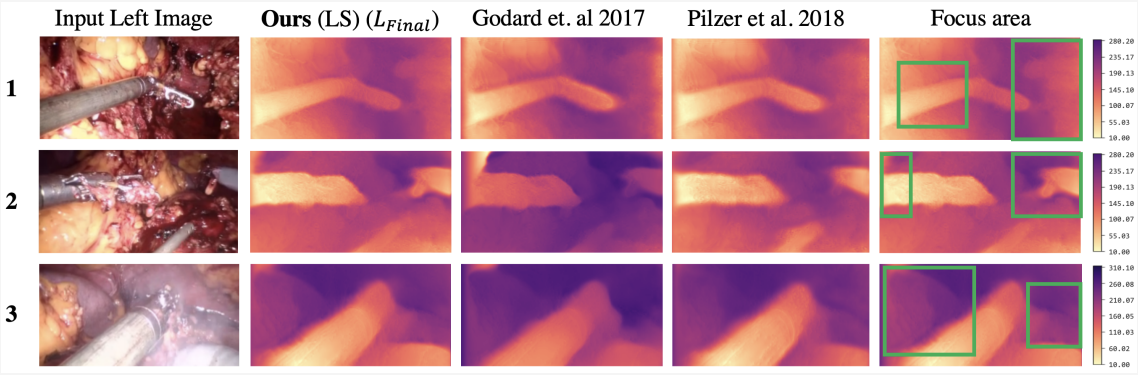


Figure 4.6: Comparison of depth maps generated by our model with [63] and [166] on the *Hamlyn* test split. The last column re-displays the depth produced by our proposed method (column 2) with the green boxes highlighting the key areas where more details are visible in our depth maps. The colour bar displays the depth range in mm.

potential sub-optimal model and the loss function's inability to penalise for high-order inconsistencies in the warped images. This is what we try to address in this work, with a loss function with stronger emphasis on image reconstruction not only via MS-SSIM but also via efficient multi-scale discriminator and deep feature perceptual losses. The proposed model enforces local and global consistency even for monocular depth estimation, whilst providing a more exploratory approach to model design (randomly connected encoder and learn-able skip connections). Furthermore, from the ablation study, one can observe that the performance of all the model variants is comparable with the other stereo competitors. In particular, our proposed method with no learn-able skip connections (no LS) trained on the standard L_{ssim} loss function is comparable to the other stereo models despite being monocular. The only novel component in this variant is the random connections, which further exemplifies the power of innovative topology for neural architecture designs. The addition of learn-able skip connections further improves on these results, due to the extra attention paid to the deep semantic features.

Table 4.1: Showing the mean absolute depth error in mm for test dataset 1 from *SCARED*. The compared methods are the self-supervised methods submitted to this challenge. (M) signifies Monodepth method, and (S) Stereodepth. The presence of L denotes the type of loss function used for training and presence of (no LS) implies no Learn-able skip connections were used in the model. Best results are highlighted in bold.

Method	Keyframe 1	Keyframe 2	Keyframe 3	Keyframe 4	Keyframe 5	Average
Ours L_{final} (M)	34.18 $\pm$ 0.071	27.01 $\pm$ 0.096	**14.82** $\pm$ 0.063	**13.42** $\pm$ 0.055	**12.89** $\pm$ 0.067	**20.46**
Ours L_{ssim} (M)	33.21 $\pm$ 0.069	26.73 $\pm$ 0.099	15.47 $\pm$ 0.066	15.13 $\pm$ 0.058	13.01 $\pm$ 0.066	20.71
Ours L_{ssim} (no LS) (M)	35.02 $\pm$ 0.074	28.10 $\pm$ 0.091	16.89 $\pm$ 0.061	17.76 $\pm$ 0.061	13.26 $\pm$ 0.068	22.206
KeXue Fu (S)	**30.49**	**18.32**	19.73	19.30	16.86	20.94
Xiaohong Li (S)	34.42	20.66	17.84	27.92	13.00	22.77

Considering Table 4.3, our solution supersedes monocular and stereo depth estimation methods

Table 4.2: Showing the mean absolute depth error in mm for test dataset 2 from *SCARED*. The compared methods are the self-supervised methods submitted to this challenge. Where (M) signifies Monodepth method, and (S) Stereodepth. The presence of L denotes the type of loss function used for training and presence of (no LS) implies no Learn-able skip connections were used in the model. Best results are highlighted in bold.

Method	Keyframe 1	Keyframe 2	Keyframe 3	Keyframe 4	Keyframe 5	Average
Ours L_{final} (M)	29.00 $\pm$ 0.061	25.81 $\pm$ 0.092	**18.72** $\pm$ 0.064	**8.90** $\pm$ 0.070	**10.12** $\pm$ 0.044	18.51
Ours L_{ssim} (M)	27.69 $\pm$ 0.059	26.17 $\pm$ 0.091	20.14 $\pm$ 0.061	9.11 $\pm$ 0.068	11.79 $\pm$ 0.054	18.98
Ours L_{ssim} (no LS) (M)	28.25 $\pm$ 0.066	27.36 $\pm$ 0.088	22.83 $\pm$ 0.065	11.69 $\pm$ 0.073	13.51 $\pm$ 0.055	20.73
KeXue Fu (S)	**23.71**	**15.46**	26.43	9.60	10.92	**17.22**
Xiaohong Li (S)	24.58	16.80	29.92	11.37	19.93	20.52

Table 4.3: SSIM index on the reconstructed images generated via the estimated disparity maps on the *Hamlyn* test dataset. Higher value indicates better performance. Where (M) signifies Monodepth method, and (S) Stereodepth. The M in parameter count signifies a million parameters. Methods, A,B,C,D and E are [60], [246], [249], [63] and [166] respectively.

Method	A (S)	B (S)	C (S)	D (M)	E (S)	**Ours** (M)
mean SSIM	0.473	0.547	0.604 $\pm$0.066	0.611 $\pm$0.076	0.633 $\pm$0.079	**0.672 $\pm$0.080**
Parameter Count	–	–	32M	31M	31M	**22M**

including [249] a stereo model. Despite our model comprising of a smaller memory footprint with significantly less parameters for training, it still outperforms the others. This shows the importance of innovative architecture design which doesn't require over-parameterised models. The above results verify that the convolution operation itself is very powerful, such that even a randomly connected network with enough convolution operations is capable of successfully learning a task. This suggests that more focus needs to go on architecture design for optimising monocular depth.

We believe the proposed methodology guides researchers in understanding how to improve monocular depth estimation without reliance on capturing arduous ground truth. To that extent we can further improve the proposed method for clinical deployment. In clinical applications, depth error should be typically in the range of sub-mm accuracy. Our method currently does not achieve this accuracy. However, it is still superior to state-of-the-art methods for monocular surgical scene reconstruction. In our future work, we aim to improve the performance of our model by experimenting with better architecture exploration strategies, which is key for training robust monocular models. More specifically, we will test more sophisticated architecture search strategies that optimise the random connections, such as population based search [60] and further pruning of extra unnecessary parameters and connections [208]. This will provide a better search and selection criteria in place of random connections and self-attention.

4.5.3 Natural Scenes Evaluation

In addition to MIS data, we also evaluate on natural scenes from the Kitti dataset [59]. The model was trained and evaluated on the Kitti Raw dataset with the Eigen Split [46]. In total, 22,600 images were utilised for training and 697 were utilised for testing. The performance evaluation study presented in this work for the Kitti data uses the same metrics as in [58][131][46], to allow comparison with methods published in previous studies. These metrics include Root Mean Square Error (RMSE), Absolute Relative Error (Abs Rel), Square Relative Error (Sq Rel), Root Mean Squared Logarithmic Error (RMSE Log) and accuracy (%). The comparison is performed with the most recent self-supervised single view depth estimation models trained on Kitti data.

Figures 4.7 and 4.8 show qualitatively the improvement the proposed architecture has on the depth outputs. where similar to Fig 4.5 and 4.6 the proposed model with LS and trained with the proposed loss function L_{final} produces sharper and structurally more consistent depth maps. This is reflected in sharper features being visible in depth as highlighted by the green boxes in Fig 4.8 and in Fig 4.7 where the final model resolved depth for distant trees in the background.

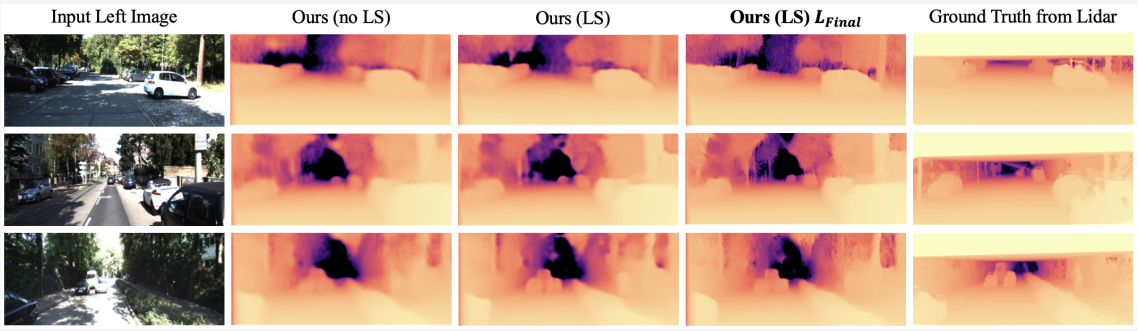


Figure 4.7: Depth maps generated by the proposed model on the *Kitti* test split. Inclusion of LS signifies presence of learn-able skip connections and $noLS$ without. all models were trained with the standard L_{ssim} loss (Eq. 4.1) unless L_{Final} is specified, which signifies training with the proposed loss function (Eq. 4.5).

The proposed model is quantitatively evaluated on the Kitti test split with the latest self-supervised learning methods in Table 4.4. It is clear that the best performing model is our proposed model training with L_{final} which sets the new state-of-the-art (SOTA) for self-supervised monocular depth estimation. Furthermore, even the other variants of our proposed models (not trained with L_{final}) still outperform all the competing methods. We believe this is attributed to the random connections which after training yield superior weight combinations, that allows the model to refine its depth estimates and thereby achieve significant boost in performance. Hence, design of architectures with innovative connections and parsing of attention on skip connection features, between the operations should be given more emphasis, as it is likely to yield better results.

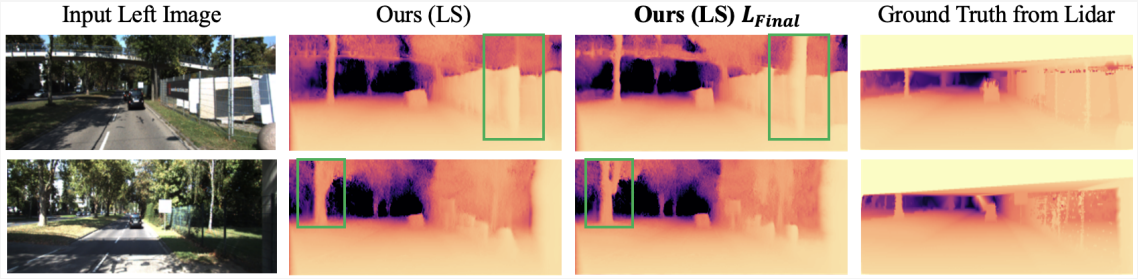


Figure 4.8: Depth maps generated by the proposed model on the *Kitti* test split. Inclusion of LS signifies presence of learn-able skip connections. all models were trained with the standard L_{ssim} loss (Eq. 4.1) unless L_{Final} is specified, which signifies training with the proposed loss function (Eq. 4.5). The green boxes highlighting the key areas where for comparison where our proposed final method resolved greater detail in depth.

Table 4.4: Performance evaluation on the Kitti Raw Dataset for distances up to 80m. All results use crop from [57]. For columns with red headers, lower value is better and for the columns with green headers, higher value is better. All the our proposed architectures are trained with L_{ssim} (Eq. 4.1), unless L_{final} is specified, which denotes training with final proposed loss function (Eq. 4.5). The best performing model variations are highlighted in blue.

Method	Abs Rel	Sq Rel	RMSE	RMSE Log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Our (no LS)	0.097	0.885	4.680	0.199	0.890	0.953	0.976
Ours (LS)	**0.094**	**0.848**	**4.575**	**0.190**	**0.903**	**0.957**	**0.977**
Ours (LS) (L_{final})	**0.089**	**0.797**	**4.467**	**0.185**	**0.905**	**0.958**	**0.978**
[5]	0.118	0.908	4.978	0.210	0.855	0.948	0.976
[168]	0.126	0.961	5.205	0.220	0.835	0.941	0.974
[235]	0.106	0.780	4.695	0.193	0.875	0.958	0.980
[63]	0.148	1.344	5.927	0.247	0.803	0.922	0.964
[165]	0.102	0.802	4.657	0.196	0.882	0.953	0.977

4.6 Conclusions

Both, the neural architecture and the loss function play a vital role in the performance of a neural network. In this work we showed that the biggest factor in optimising monocular depth is the design of the neural network architecture itself and the loss function's capability of computing high-order multi-scale image reconstruction metrics. We proposed the first method to generate randomly connected encoder-decoder networks and a new U-net topology for exploiting hidden semantic features. Our results verify that even a randomly connected network with standard convolution operations but innovative inter-connections can learn the task well. Additionally, we showed that multi-scale penalty in loss functions is vital for generating finer details. We believe that our study will aid in further research of neural network architecture design, and shift from the standard U-net and manual trial and error based approaches to more automated design methods. Our future work will entail experimenting with more loss terms, and exploring more inventive methods of conducting search in the connections space to see how other variations of the network impact the results.

4.6.1 Learnings & Future Work

In this chapter; investigating monocular depth estimation via Deep Learning (DL), equipped me with the essential knowledge on designing depth estimation frameworks. By drawing from the learnings in Chapter 3, I devised a loss function that incorporated perceptual learning and generative modeling, enhancing self-supervised depth estimation. This experience not only allowed me to improve monocular depth estimation techniques but also provided insights into the next steps for extending this work to stereo inputs, which is addressed in the follow-up chapter (5) and also outline some improvements for future work (listed below).

1. Processing higher input resolutions may improve depth: As shown in [163] simply by operating self-supervised depth models at high resolution inputs, improves depth accuracy. Hence, one easy experiment is to input images at 720p or 1080p resolution and retrain the model at that scale. Furthermore architectural changes to the decoder [69] to output high resolution depth maps, can also yield improved performance

2. Jointly estimate camera intrinsics to overcome scale ambiguity: A primary challenge in monocular depth estimation is scale ambiguity. Consequently, a well-trained model may require retraining for different camera setups, as changes in camera intrinsics influence depth perception in single images. One approach, similar to [68], involves estimating both camera intrinsics and depth by leveraging the latent space of the U-Net architecture. This method can enhance the invariance of monocular depth models to various camera configurations, ul-

timately improving their generalisability.

3. Meta Learning / Zero shot generalisation: New methods now employ zero shot learning [172] to improve generalisability of depth. However in this particular paper, the authors train on a proprietary dataset (which is not available for public use). One method to boost performance is to train monocular depth on simulation data, with the same training strategy from [172]. This could greatly enhance generalisability to changing scenes and improve results since ground truth depth will be used as a sharper unambigous learning signal for training. Furthermore, combining this with suggestion 2 mentioned above, will make the method generalisable and high performance to even changing camera scenarios.

Chapter 5

Improving Stereo Depth Estimation via Self-Supervised Contrastive Representation Learning

While pursuing sub-millimeter depth accuracy for surgery and building upon Chapter 4, I explored enhancing the complexity of my solution to accommodate stereo inputs (better suited for surgical scene and improved depth). After investigating traditional Computer Vision (CV) and Deep Learning (DL) solutions in stereo depth estimation, it became evident that supervised learning-based approaches consistently outperformed their self-supervised counterparts. However, supervised approaches rely on exhaustive ground truth data and have limited transfer-ability to new data distributions (as outlined in chapter 4 and in literature). These characteristics limit their use in surgery and for future upgrades (especially if data has to be recollected; which is arduous). The question addressed in this chapter is: can we enable self-supervised learning approaches to learn weight distributions akin to supervised approaches? If so, one can utilise all the unlabeled surgical data and jointly enhance stereo-depth estimation performance without the needing dense ground truth data (which is difficult to attain). Hence, building on my previous research in self-supervised learning I designed the first Contrastive Representation Learning (CRL) method to work on highly similar images like stereo and made it scalable for training on a single GPU (traditional CRL required substantial computation resources and worked well on dissimilar images). CRL allows discretising feature representations for the stereo input such that even when paired with self-supervised learning strategy for depth estimation, DL models can optimise stereo depth estimation with lower ambiguity and errors. This work attained state-of-the-art (SOTA) performance on both surgical and natural scenes showing the power of learning generalisable representations. The insights from this work enabled me to create a follow-on method that achieved sub-millimeter accuracy, discussed in chapter 6.

The work in this chapter was published in the conference proceedings of the 25^{th} international conference on Medical Image Computing and Computer Assisted Intervention (MICCAI) in 2022 [217].

5.1 Overview

Accurate stereo depth estimation is crucial for 3D reconstruction in surgery. Self-supervised approaches are more preferable than supervised approaches when limited data is available for training but they can not learn clear discrete data representations. In this work, we propose a two-phase training procedure which entails: (1) Performing Contrastive Representation Learning (CRL) of left and right views to learn discrete stereo features (2) Utilising the trained CRL model to learn disparity via self-supervised training based on the photometric loss. For efficient and scalable CRL training on stereo images we introduce a momentum pseudo-supervised contrastive loss. Qualitative and quantitative performance evaluation on minimally invasive surgery and autonomous driving data shows that our approach achieves higher image reconstruction score and lower depth error when compared to state-of-the-art self-supervised models. This verifies that contrastive learning is effective in optimising stereo-depth estimation with self-supervised models.

Brief video explaining the work: 5 min video

5.2 Introduction

Depth estimation holds substantial importance in a wide range of applications in Minimally Invasive Surgery (MIS) ranging from soft tissue 3D reconstruction, surgical robot navigation to augmented reality. One method of perceiving depth is with a stereo camera by estimating the horizontal displacement (disparity) from the left image pixels to the corresponding right. End-to-end deep learning stereo reconstruction methods, have become state of the art. They are typically categorised as supervised and self-supervised.

Stereo reconstruction models trained via supervised learning directly learn to regress the disparity values relying on ground truth data [129][70][256][210].These models are typically trained on synthetic data [141] followed by fine-tuning on real scenes, to deal with limited ground truth availability. However, there is an inherent discrepancy between the two data sources, especially in surgical applications where scarce ground truth data, poorer image quality, limited space, shorter depth distance and a myriad of scene variations, can cause model performance to drop. For self-supervised methods, the training data is abundant since, only images and no ground truth is required. Therefore, an end-to-end self-supervised approach capable of harnessing diverse real data, can generate high-quality results for any scenario where ground truth data is limited. Self-supervised stereo reconstruction methods are trained to conduct novel view synthesis thus, optimising the photometric re-projection error [165][228][264][214]. This error however, can be optimised by a wide range of divergent disparity values resulting in inconsistent geometry. This is because self-supervised methods can not learn discrete feature representations between the stereo views.

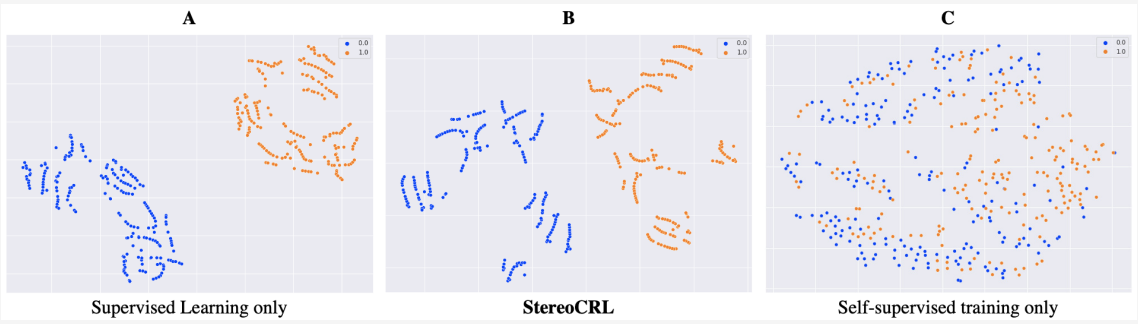


Figure 5.1: T-SNE visualisation of model embeddings on Kitti2015 test data (200 images). Labels 0 and 1 signify embeddings of left and right views, respectively. (A) generated from pre-trained [129], (B) our proposed StereoCRL trained with Eq. 1&2 (C) our model variant trained only via self-supervision Eq.2, on Kitti raw split (42,382 stereo frames).

As shown in Fig. 5.1(A), supervised methods have a unique property of learning clear discrete feature representations for the stereo views (shown by the distance between left and right embeddings) whilst still learning similarities between the views (shown by the similar shape of the feature representations). Self-supervised methods are unable to reach that resolution in the feature

space as shown in Fig. 5.1(C), learning a mixed embedding between left and right views and thus a confused disparity.

The question we address in this paper is, how can we make self-supervised approaches learn discrete representations similar to supervised methods, without relying on ground-truth? To achieve this, we propose *StereoCRL* a two-stage training process which entails (1) training an encoder via CRL [28], to generate distinct feature embeddings for left and right views, followed by (2) self-supervised training for disparity estimation.

5.2.1 Contributions

Our main contributions are the following:

1. We design a procedure for training stereo depth models based on joint contrastive representation and self-supervised learning. To the best of our knowledge, this is the first CRL methodology applied for stereo depth estimation.

2. We introduce a new CRL method by defining a momentum pseudo-supervised contrastive loss for efficiently scaling and training our encoder model.

3. We pair our CRL-trained encoders with a decoder architecture based on cost volume feature aggregation for learning high quality disparity maps.

4. We exhibit that our CRL methodology can boost performance of models trained simply with traditional image re-projection error.

Existing CRL loss functions [28][74] require high computational resources to learn discriminative representations between highly similar images. Our proposed CRL method is tailored for learning differences between highly similar images such as stereo pairs, and is scalable even on a single GPU hardware. The performance of our StereoCRLhas been evaluated on 2 MIS datasets *Hamlym* [61] and *SCARED* [6] and non-surgical scenes from Kitti2015 [146].

5.3 Related Works

5.3.1 Supervised Stereo Depth Estimation

In supervised stereo depth estimation, all approaches learn to directly regress the ground truth disparity values. Initial approaches, utilised convolutional networks to perform stereo matching. Essentially, replacing the hand crafted features used in traditional stereo-matching algorithms with deep learning features. Recently, fully end-to-end methods have been proposed that replace this stereo matching pipeline and instead focus on designing the deep learning model architectures. One of the popular approaches involved using a combination of a convolutional backbone, with spatial pyramid pooling module and cost volume aggregation which leads to superior results [70]. Some methods propose more sophisticated ways of cost volume regularisation [256]. [210] proposed a new model architecture without explicitly building a cost volume. [129] introduced a new architecture inspired by optical flow models and recurrent neural networks. However, in all supervised learning methods ground truth is a necessity for training which is not always feasible to collect. Especially in the case for surgery where ground truth availability is limited and domain adaptation is a key issue.

5.3.2 Self-Supervised Stereo Depth Estimation

In self-supervised stereo depth estimation, no ground truth is required. The problem is formulated as novel view synthesis, hence, the learning signal is based on photometric re-projection error. [63] introduced, a stereo based re-projection error, for training a monocular model. This has been adopted and enhanced in many proposed methods for stereo depth estimation. [165] extended [63] framework to stereo, by introducing a new model architecture and experimenting with adversarial training with cycle consistency for the photometric loss. [228] trained a joint self-supervised optical flow and stereo depth estimating model in a multi-task learning approach. Additionally, [264] proposed a new model architecture based on cost volume generation and Total Generalized Variation (TGV) term as regularisation for the photometric error. [214] took a different approach, by performing supervised pre-training of a model followed by self-supervised continuous adaptation. However, a persisting issue with photometric error is that, it can be optimised by a range of incorrect disparity values. This inhibits end-to-end self-supervised model performance. Thus, in this work we explore a CRL as a pretext task to improve end-to-end self-supervised learning.

5.4 Methods

5.4.1 Stereo Contrastive Representation Learning

The premise of contrastive learning is to learn distinctiveness in the data representation. It involves training an encoder model to be capable of recognising similarities in scenes that are alike and differences in dissimilar scenes. The aim of CRL is to ensure the encoder learns general features such that its embeddings for similar objects (positive samples) are grouped closer together in the embedding space and separated from dissimilar objects (negative samples). In our proposed StereoCRL, positive and negative samples represent the left and right images, respectively [28]. Thus, our encoder model must learn discrete feature representations for the left and right images, such that for the same view the representations are close together and for the corresponding stereo view farther apart, as shown in Fig. 5.1.

To design an efficient CRL algorithm we must overcome the following challenges. *(1) Limited Resources:* CRL algorithms like SimCLR [28], require the encoder to observe vast number of negative samples at each training iteration (i.e. up to 4000 samples per batch). This is to ensure the embeddings for each positive sample are compared with enough divergent negative samples at each gradient update. *(2) High scene similarity:* most existing CRL algorithms are initially trained on classification tasks where, there is enough variation between objects in positive and negative samples. However, in the case of stereo images, the scene is extremely similar hence, optimising their embeddings to be far apart will require an additional learning signal.

Consider SimCLR [28] method adapted for stereo images is illustrated in Fig. 5.2(A). Each image is first parsed into a data augmentation module that performs the following set of transformations randomly: (1) horizontal flipping, (2) colour distortion, (3) Gaussian blurring and (4) variations in brightness, luminance (gamma parameter) and saturation. The augmented variants for each batch are processed by the encoder to generate a set of embeddings, h_i, for each image i in the batch. These embeddings are further parsed through a projection model which is a small Multi Layer Perceptron (MLP) G_ϕ which re-projects them into a different latent space followed by L2 normalisation i.e. $z_i = G_\phi(h_i)$. The embeddings z_i for the same view, should maximise cosine similarity. G_ϕ is only used for CRL training and is discarded upon completion.

However, this standard CRL solution suffers from challenge (1) mentioned above. Hence, we advance the CRL process by formulating CRL as a dictionary look up problem as shown in Fig. 5.2(B). Instead of using large batch sizes to store negative samples for training, we use a memory bank (queue) to store the generated embeddings, same as [74]. In our case, the projection network G_ϕ, predicts $q_i = G_\phi(h_i)$ instead of z_i for each i image in the batch. Where q_i is a query embedding of h_i. Every time a batch is processed, a q_i is generated. At the same time, another encoder model entitled *key encoder* processes a batch of negative images and a single positive image to generate

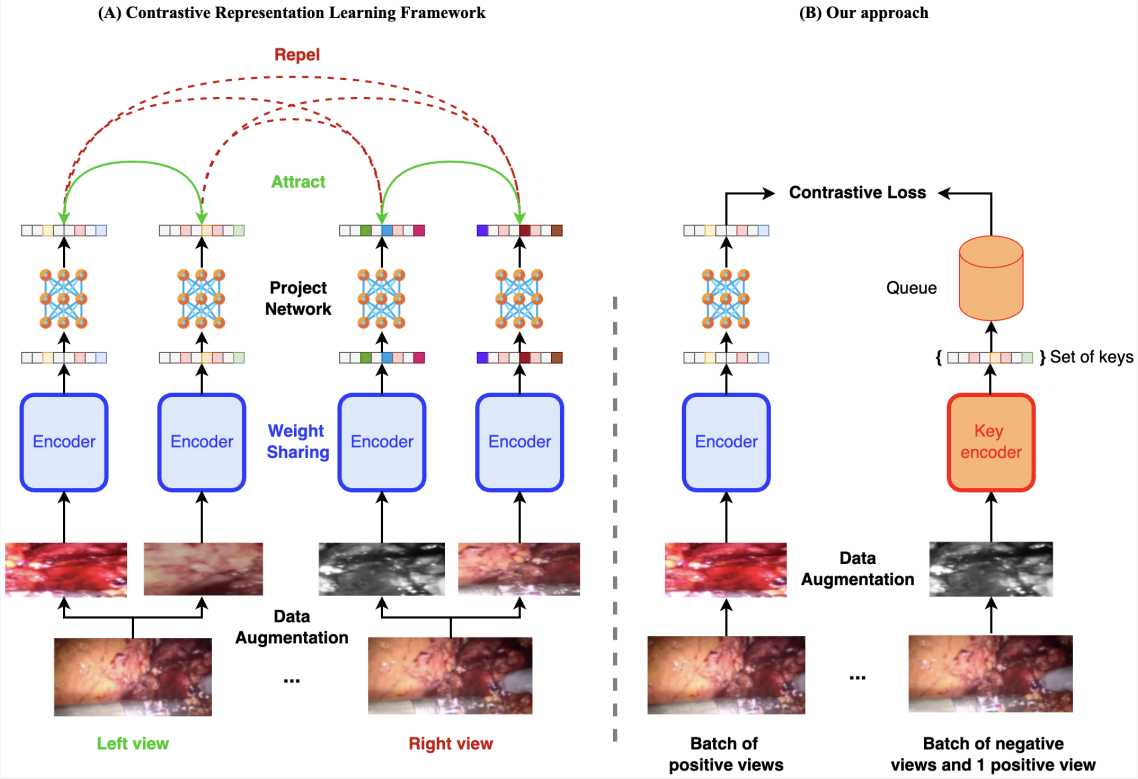


Figure 5.2: The stereo contrastive learning framework. (A) The intuition of CRL and (B) Our proposed CRL approach.

keys k_i^- and k_i^+, respectively. The *key encoder* is not trained, instead its weights are updated using the weights of the other encoder via momentum update [74]. The keys generated are stored in a memory bank, dynamically creating a dictionary of features on the fly. The intuition is that there is a single key k_i^+ that the q_i matches with. Thus, the contrastive loss value will be low when q is similar to k_i^+ and different from the other keys (k_i^-) in the queue. Ultimately, the encoder learns to make q for the same view highly similar, without needing large batches of images, since the loss can be scaled to observe negative samples as query-key vectors.

The above solution still suffers from challenge (2). To overcome this challenge, we define a new loss function to train our encoder based on a supervised variant of the contrastive loss [107]. This loss function leverages ground truth labels as an anchor to enable the encoder to implicitly cluster embeddings of the same class together whilst simultaneously distancing clusters from different classes. To ensure we maintain self-supervision, we create pseudo ground truth labels live during training. Our labels are binary, i.e. 0 and 1 for the left and right images, respectively. This forces the model to simply differentiate between left and right feature embeddings such that they are far apart from one another. We further augment this loss, to enable training with our proposed CRL approach, using query-key vectors. Hence, we introduce the momentum supervised contrastive loss:

$$L_{mo:crl}^{sup} = \Sigma_{i=1}^{2n} \frac{-1}{2|N_i| - 1} \Sigma_{i \in N(y_i)} log \frac{exp(q_i \cdot k_i^+ / \tau)}{\Sigma_{i=0}^{K} exp(q \cdot k_i / \tau)} \tag{5.1}$$

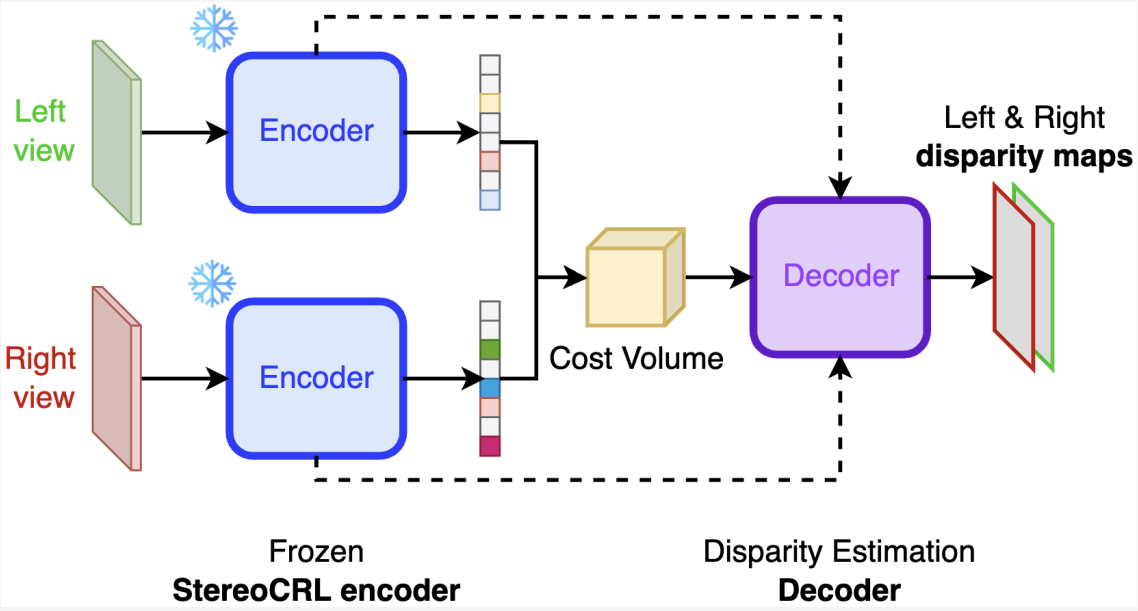


Figure 5.3: StereoCRL model architecture. Solid black and dotted lines denote forward propagation and skip connections respectively.

where, $N = (i, y_i)$ is the number of randomly sampled images in a batch and the corresponding label (y_i), i is the augmented image sample in the batch and τ is a temperature parameter, used for scaling the output in the range of -1 to 1 for calculating cosine similarity. k_i^+ is the single positive key in the memory bank. The sum is conducted over one positive and K negative samples. This loss is effectively the log loss of $K + 1$ samples where a softmax based classifier (G_ϕ) is trying to classify q_i as y_i and k_i^+. Hence, phase (1) of our method, involves pre-training the encoder via Eq. 5.1, prior to the downstream task of disparity estimation. In phase (2) the key encoder is discarded, and a decoder model is trained for self-supervised disparity estimation. The final model is shown in Fig. 5.3 where the trained encoder is frozen and paired with a decoder.

5.4.2 Decoder Architecture

Our decoder comprises of 5 sequential blocks to mirror the encoder, schema shown in Fig. 5.4. Each decoder block utilises 3 convolution layers namely,*UpConv*, *iConv* and *DispPred*. All layers have a filter size of 3x3 except *DispPred* which is 1x1. All convolution layers are followed by normalisation (Batch) and activation (eLU) layers, except for *DispPred* since it outputs direct disparity. Each of the last 4 blocks outputs left and right disparity maps which are increasing spatially by a scale-factor of 2 in ascending order. Much like a *DispNet* [143] architecture, our decoder takes forward propagating feature maps and skip connection feature maps from both left and right view encoders, shown in Fig. 5.3. We redesign the original *DispNet* decoder layer to ensure effective learning from both the incoming features and the skip connection features by generating a cost volume at each scale for multi-scale disparity estimation. The cost volume is used to encode features for disparity by discretising in disparity space and comparing the features along epipolar lines.

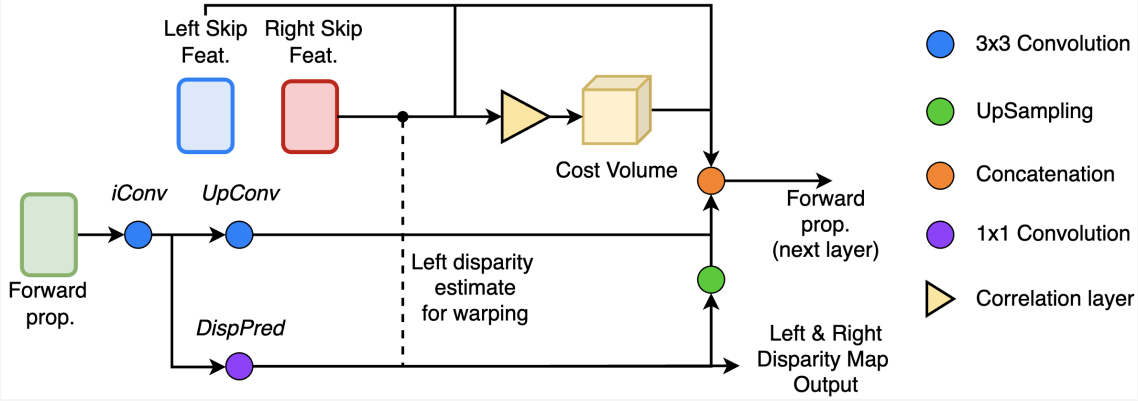


Figure 5.4: Decoder block schema. The solid and dotted lines signify forward propagation and skip connections, respectively.

This overcomes the issue of normalised disparity predicted by DispNet and other self-supervised models [165], [228] which is affected by changes in image scale.

Typically, to process cost volumes, 3D convolution is used. However, this requires high computational cost and limits operations at low resolution. Given our loss is based on photometric error, we require iterative refinement. We opted for an efficient multi-scale cost volume generation same as PWCNet [203] at each decoder block. In particular, the right skip connection feature map is warped using the predicted left disparity, and the resulting warped feature map is correlated with the left skip connection feature map. For our cost volume, we set the search range to [-5, 4] pixels, allowing a sufficient radius for negative and positive disparities. Furthermore, we estimate the cost volume in each decoder block alongside disparity. At each block all the feature maps are increased by a scale factor of 2. Thus, StereoCRL iteratively refines its disparity estimate by adjusting its cost volume weights at each scale.

5.4.3 Self-Supervised Loss

In phase (2) of our method, only the decoder is trained via self-supervised photometric error using stereo video sequences, similar to [63] for disparity estimation. The output predicted left and right disparity maps (D_L^{pred} and D_R^{pred}) from the last 4 layers of the decoder are used to warp the input stereo frames I_L^{GT} and I_R^{GT}. The photometric warping error is defined by utilising a warping function F_{warp} on the input images for generating its corresponding stereo image via the opposite disparity map i.e. $F_{warp}(I_L^{GT}, D_R^{pred})$ generates the predicted right image I_R^{Pred} and vice versa. The loss function therefore, is based on the reconstruction error between the synthesised images and the original image which acts as ground truth. Standard loss components utilised are image reconstruction L_{recon}, disparity smoothness L_{smooth} and left-right disparity consistency L_{LRcons} [63]. This is defined as:

$$L_{pe} = \delta_1 L_{recon} + \delta_2 L_{smooth} + \delta_3 L_{LRcons} \tag{5.2}$$

L_{pe} is performed on both the synthesised left and right images at each scale. L_{recon} is composed of Multi-Scale Structural Similarity Index (MS-SSIM) together with a L1 loss term.

5.4.4 Implementation Details

The encoder architecture used is a ResNet50. For phase (1) of CRL training, the encoder was trained for a maximum of 200 epochs on a combination of the following 3 datasets: *Kitti raw data* containing 42,382 frames, *CityScapes* [37] containing 88,250 frames and *Hamlyn* MIS data containing 34,240 frames. For phase (2), only the decoder was trained via L_{pe} on *Hamlyn* MIS and Kitti raw data, independently. Our maximum disparity range is set to 192. The output dimension of query and key vectors q_i and k_i were 128. The temperature τ was set to 0.07 and the batch-size used in the study was 16 and maximum memory bank capacity was 3000. In L_{pe}, the weightings δ_1,δ_2 and δ_3 were 0.85, 1.0 and 1.0, respectively. The hyperparameters were selected using the original paper sources, for temperature τ [74] and for deltas δ_x [63], respectively. Testing was performed on the SCARED test dataset [6] comprising of 5,637 stereo image pairs without any finetuning (for testing generalisation), Kitti2015 comprising 200 frames (all of which were used only for testing) and *Hamlyn* comprising 7,191 frames. Images were processed at 256×512 resolution. Nvidia Tesla T4 was used for training. CRL and self-supervised training computation time was 3.5 and 1 GPU days, respectively. We also train a variant of StereoCRL only on Eq. 5.2 (no CRL pre-training) for comparison. For this process we randomly initialise weights of the encoder and jointly train it with the decoder.

5.5 Results & Discussion

The performance evaluation was conducted on the *Hamlyn*, SCARED and Kitti 2015 datasets. Due to lack of ground truth in *Hamlyn*, we compare the Structural Similarity Index (SSIM) between the original and the predicted stereo images, same as [249]. Since ground truth is provided in SCARED, we compare mean absolute depth error in mm on test sets 1 and 2 with other self-supervised methods submitted to the original challenge [6]. Since Kitti2015 contains ground truth disparities too, we show end-point-error benchmark.

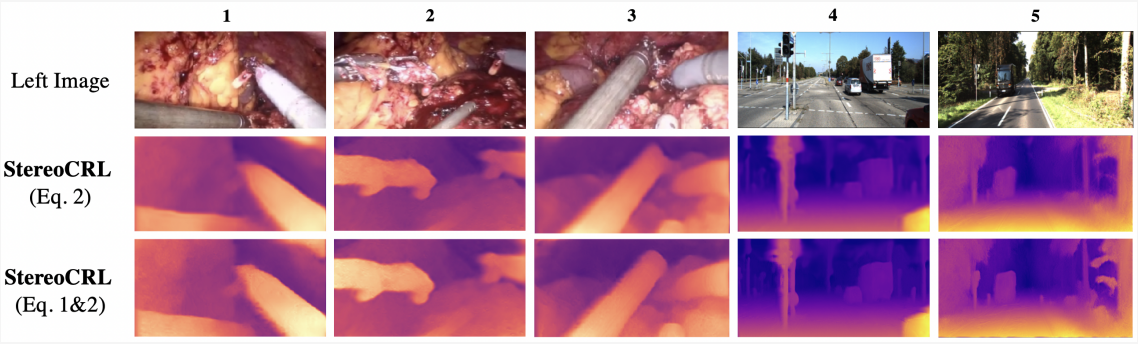


Figure 5.5: Depth maps generated on the *Hamlyn* and the Kitti2015 test split. Inclusion of (Eq.) signifies the type of loss the model was trained with.

Fig. 5.5 compares depth estimated by StereoCRL model variants, i.e. one trained on the proposed approach with Eq. 5.1&5.2 and one trained only via self-supervision with Eq. 5.2. One can observe that StereoCRL Eq. 5.1&5.2 outperforms StereoCRL Eq. 5.2 in all scene types. This is visible from the sharper depth maps generated by StereoCRL Eq. 5.1&5.2. The model variant trained on Eq. 5.2 is unable to resolve depth for the anatomical structures in the background as shown in Fig. 5.5 samples 1 and 3. Furthermore, it results in blurry disparity maps around the edges of robotic tools and anatomical structures. Whereas, the Eq. 5.1&5.2 variant is exceptionally capable of learning different features for such anatomical structures in the background, creating smoother and sharper depth. Same performance is visible when observing non-surgical scene results in samples 4 and 5. The Eq. 5.1&5.2 variant estimates sharper depth around the leaves on the trees (sample 5) and the poles holding the traffic lights (sample 4). The full CRL approach overcomes the issue of mixed feature embeddings that pure self-supervised learning typically generates. This enables the model to learn independent discrete feature representations for both the input views, which are later combined by the decoder for prediction of sharper depth maps with iterative refinement.

The proposed model is also quantitatively validated on the *Hamlyn*, *SCARED* and Kitti2015 datasets in Tables 5.1, 5.2 and 5.3, respectively. In all three cases, our model outperforms traditional stereo depth estimation methods as well as, state-of-the-art (SOTA) self-supervised deep learning methods. In Table 5.1, the high SSIM score verifies that our model generates high fidelity disparity maps and yields better reconstruction accuracy. This exhibits the importance of explicitly mod-

Table 5.1: Performance evaluation on the *Hamlyn*. Best results highlighted in bold.

Methods	ELAS [60]	SPS [246]	V-Siamese [249]	StereoCRL Eq. 2	**StereoCRL** Eq. 1&2
Mean SSIM	47.3	54.7	60.4	80.8	**83.7**
Std. SSIM	0.079	0.092	0.066	0.033	**0.023**

Table 5.2: Mean absolute depth error (mm) on SCARED. Best self-supervised results are highlighted in bold.

Method	Method Type	Test Set 1	Test Set 2
Trevor Zeffiro [6]	Supervised	3.60	3.47
Xiran Zhang [6]	Supervised	6.13	4.39
J.C. Rosenthal [6]	Supervised	3.44	4.05
Zhu Zhanshi [6]	Supervised	9.60	21.20
KeXue Fu[6]	Self-Supervised	20.94	17.22
Xiaohong Li[6]	Self-Supervised	22.77	20.52
StereoCRL Eq. 2	Self-Supervised	19.81	16.32
StereoCRL Eq. 1&2	Self-Supervised	**16.91**	**11.10**

elling stereo features within self-supervised models. The same can be inferred from the results in Tables 5.2 and 5.3, which shows that our model improves on the current benchmark. Particularly, Table 5.2 also shows our model's capability in generalising to other laparoscopic stereo frames, since no fine-tuning was performed. In Table 5.3, we outperform other SOTA methods. Additionally, to asses the impact of our proposed CRL loss defined in Eq. 5.1, we performed an ablation study by training 2 new variants of StereoCRL using (1) Vanilla CRL loss [28] and (2) Vanilla MoCo loss [74]. Both variants are later finetuned for disparity estimation via Eq. 5.2. Variant 1 achieved average depth error of 24.5 and 21.4 mm on SCARED datasets 1 and 2, respectively. Variant 2 achieves 23.2 and 23.6 mm on the same datasets. Both variants perform worse than the proposed StereoCRL method, which achieves 16.9 and 11.1 mm error, respectively. The results of the new variants are like a model trained only on photometric loss. This further verifies StereoCRL's ability in learning discriminative representations.

Table 5.3: Evaluation Results on Kitti2015 Benchmark. Measuring % of disparity outliers (D1). Where 'all' signifies outliers averaged over all ground truth pixels image. Best results highligted in bold.

Method	Method Type	D1-all
SsSMnet [264]	Self-Supervised	3.57
MADNet [214]	Self-Supervised	4.66
UnOS [228]	Self-Supervised	5.58
StereoCRL Eq. 2	Self-Supervised	4.91
StereoCRL Eq. 1&2	Self-Supervised	**3.55**

5.6 Conclusions

Supervised learning based depth estimation approaches typically outperform self-supervised counterparts. However, training a model to descritise the stereo input images into distinct feature representations is certainly the way forward to improve self-supervised training. We show that our approach despite being end-to-end self-supervised, is capable of learning generalisable feature representations without reliance on expensive ground truth on surgical data. This is due to CRL boosting the model's feature representations. Our future work will entail experimenting with additional loss terms such as occlusion masking, and experimenting our contrastive learning approach paired with a supervised model to explore if generalisation can be improved.

5.6.1 Learnings & Future Work

Through my work on this chapter, I gained a deeper understanding of the crucial factors for enhancing stereo depth performance in Deep Learning (DL) models. This accomplishment not only showcased the effectiveness of the methods employed but also laid the foundation for the subsequent chapter (6) where I develop a framework that attains sub-millimeter depth accuracy. In this chapter, I discovered self-supervised learning is effective for attaining generalisable stereo input representations but suffers from ambiguity due to photometric re-projection error. Supervised learning methods, on the other hand, provide sharper learning signal. So it raises the question is self-supervised learning really even needed for stereo depth estimation downstream once generalisable features have been learnt? I explore this in the next chapter, where I propose a joint self-supervised and supervised learning framework for achieving generalisable stereo depth with sub-millimeter accuracy. Nevertheless, I have following suggestions on how self-supervision should be employed in stereo depth estimation for future.

1. Live adaptation: Instead of using self-supervised learning as a baseline training method, use it to update the model online in real-time. Since, photometric re-projection error is self-supervised one can perform test-time training to continuously improve results in real world settings.

2. Data distribution drift in deployable pipelines: One key issue when deploying models in the real world is data-drift. This is where the distribution of real-world data is significantly different from training and this causes model estimated to fail. The photometric re-projection metric can be used as both a way to detect drift as well as a way to overcome it via test-time training, suggested in (1).

Chapter 6

Generalisable Stereo Depth Estimation via Masked Image Modelling

In this work, I address and achieve the challenge of obtaining sub-millimeter accuracy in depth estimation for Minimally Invasive Surgery (MIS) (problem #2 outlined in Introduction). Building on the findings from Chapter 5, it became clear that self-supervised learning helped learn generalisable feature representations for stereo inputs. However, the obstacle to attaining sub-millimeter accuracy is primarily the self-supervised loss function itself. This loss is a photometric re-projection error, which can be optimised for a broad range of disparity values, leading to inconsistent geometry. Furthermore, it is scale ambiguous which inherently distorts the relative depth information. Hence, supervised learning is vital, as the ground truth training data provides an unambiguous & sharp signal for accurate depth representations. The question addressed in this chapter is: Is it possible to achieve best of both worlds i.e. accurate and unambiguous depth (via supervised learning) that can also adapt and generalise to varying scenes (via self-supervised learning)? My intuition was that self-supervised learning was only necessary to assist in acquiring generalisable feature representations for the scenes' context. Once achieved, supervised learning can optimise for depth without affecting generalisability, because the loss is based on patch-based comparison between the two views which is not scale ambiguous. I proposed a two-stage approach, beginning with Masked Image Modelling (MIM) for generalisable feature representation learning, followed by supervised learning for depth estimation on simulation data. Simulation data was intentionally chosen due to (1) dense ground truth, and (2) no bias towards natural/surgical scenes. The method achieves sub-millimeter accuracy, setting a new state-of-the-art (SOTA).

This work has been submitted to the conference proceedings of the 26^{th} international conference on Medical Image Computing and Computer Assisted Intervention (MICCAI) in 2023. Additional co-authors (2^{nd}: Chi Xu, and 3^{rd}: Haozheng Xu) aided with the results generation.

6.1 Overview

Generalisable and accurate stereo depth estimation is vital for 3D reconstruction, especially in surgery. Supervised learning methods obtain best performance however, limited ground truth data for surgical scenes limits generalisability. Self-supervised methods don't need ground truth, but suffer from scale ambiguity and incorrect disparity prediction due to inconsistency of photometric loss. In this work, we propose a two-phase training procedure that is generalisable and retains the high performance of supervised methods. It entails: (1) Performing self-supervised representation learning of left and right views via Masked Image Modelling (MIM) to learn generalisable semantic stereo features (2) Utilising the MIM pre-trained model to learn robust depth representation via supervised learning for disparity estimation on synthetic data only. To improve stereo representations learnt via MIM, we introduce perceptual loss terms, which improves the model's stereo representations learnt by explicitly encouraging the learning of higher scene-level features. Qualitative and quantitative performance evaluation on surgical and natural scenes shows that our approach achieves sub-millimeter accuracy and lowest errors respectively, setting a new state-of-the-art (SOTA). Despite not training on surgical nor natural scene data for disparity estimation.

6.2 Introduction

Depth estimation is of paramount importance in Minimally Invasive Surgery (MIS) facilitating, soft tissue 3D reconstruction, surgical robot navigation and augmented reality. A popular method of perceiving depth is with a stereo camera by estimating the horizontal displacement (disparity) from the left image pixels to the corresponding right. End-to-end deep learning stereo reconstruction methods have become SOTA. They are typically categorised as supervised and self-supervised. Supervised learning methods directly predict disparity (regression) by training on ground truth disparity data [129][70][256][210]. They usually undergo training using synthetic data [141], followed by fine-tuning on real-world scenes to overcome the limited availability of ground truth. Despite their popularity, they face several challenges in surgical applications. Discrepancy between training and inference data, limited ground truth data, lower image quality, restricted space, and a multitude of scene variations, are factors that lead to reduced performance.

Self-supervised methods don't require ground truth disparity, making images-only training data abundant. These methods optimise photometric re-projection error through novel view synthesis. [217][165][228][264][214]. Self-supervised methods show potential, but are not as effective as supervised methods. This is because the photometric error can be optimised for a large range of disparity values, resulting in inconsistent geometry. Furthermore, the relative depth information between stereo images is inherently scale ambiguous. Making it challenging to learn robust representation of depth via self-supervision, especially in scenes with occlusions, textureless regions, or repetitive structures, which is prevalent in surgery. Supervised learning is key for robust stereo depth, since training on ground truth provides a strong, unambiguous signal for learning accurate representations. A question emerges though; can we have the best of both worlds i.e. unambiguous depth that is also generalisable to different scenes?

For achieving generalisability, visual representation learning is key. Learning robust feature representations also allows improved downstream performance as shown in [73][12][27]. These techniques utilise image reconstruction as a preliminary task, relying on the principle that by learning to predict patches in a masked image, useful representations about the context of the scene can be obtained for further tasks. This is proven by the enhanced data label efficiency on various standard benchmark datasets. Hence, in this work we experiment with this MIM approach, to train an encoder model to learn stereo feature representations, that can be fine-tuned downstream to encode robust depth feature representations. This endows our model with both generalisable features via MIM and unambigous sharp depth via supervised-learning for disparity estimation.

In this paper we build on [73] and propose *StereoMAE* a two-stage training process which entails (1) training an encoder via MIM, to generate robust feature representations for left and right views, followed by (2) supervised training for disparity estimation. Furthermore, we enhance MIM in (1) by using perceptual similarity learning [94], leading to learned representations that effectively capture

the intricacies of the scene and object boundaries without explicit guidance or manually designed inductive biases.

6.2.1 Contributions

Our contributions are the following:

1. A novel approach for training stereo depth models by combining self-supervised MIM and supervised stereo depth estimation. To the best of our knowledge, this is the first work to apply MIM for stereo depth estimation.

2. We propose a new approach to boost MIM by incorporating perceptual similarity loss term for learning generalisable visual semantic concepts.

3. We present a modular model architecture for combining any pre-trained MIM encoder model with any off-the-shelf decoder to enhance depth estimation.

4. Our joint MIM-supervised approach enhances performance, yielding a generalisable model with sub-millimeter accuracy in surgical depth estimation.

The performance of StereoMAE has been evaluated on MIS data; *SCARED* [6], non-surgical scenes from ETH3D [189] and Middlebury [78].

6.3 Methods

The method is divided into two phases; (1) Pre-training via MIM for enhanced representation learning followed by (2) supervised downstream fine-tuning for stereo depth estimation. The learning framework for (1) is inspired by MAE [73]. Supervised training in (2) is based on RAFT-Stereo [129]. However for (2) any training methodology from the supervised stereo depth literature can be used.

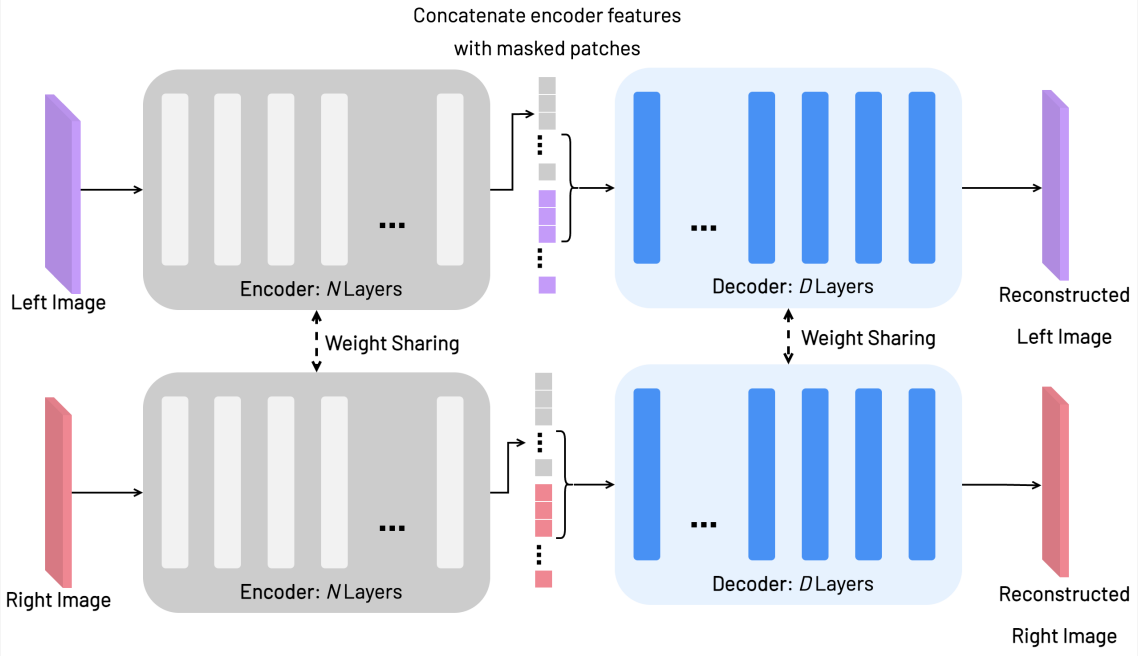


Figure 6.1: Masked Image Modelling (MIM) pipeline for StereoMAE comprising of a weight sharing transformer encoder-decoder model. Where N and D are the total number of transformer layers that make up the corresponding models.

6.3.1 Stereo Masked Image Modelling

MIM involves randomly masking pixels of an input image and training a model to predict the invisible content [73][12][27]. The intuition is that the model learns the representation of the masked pixels by inferring details from the surrounding valid pixels. In the end, if a model is capable of reconstructing the missing content with valid pixels, the model has learnt the context within the image by encoding its semantic features.

We adapt the method in [73], which utilises an auto-encoder model for stereo images as shown in Figure 6.1. In particular, we construct the auto-encoder model, using Vision Transformer (ViT) components [114]. The encoder consists of several layers of ViT-Base (ViT-B), a 12 layer transformer model. The custom decoder is composed of 8 transformer layers. Given the left image as input, it is first resized to 224x448, divided into patches of size 16x32 and 75% of them are randomly masked.

The encoder is fed only the un-masked patches as input, and it generates corresponding features representing the visible scene parts. These features are then concatenated with the patches that were not fed into the encoder and inputted into the decoder to reconstruct the full left image as output. The model is trained via pixel image reconstruction (photometric error). The same is done for the right input image. For the left and right input views, the weights of the encoder and decoder are shared, allowing them to learn joint global stereo representations for reconstructing the scene.

In our work, we do not train the aforementioned model using the Mean Square Error (MSE) loss from the original work [73]. MSE, which compares pixel intensities, is a low-level metric and insufficient for capturing complex structures in an image as it assumes pixel-wise independence. Hence a model's capacity is wasted since high frequency global feature distribution will not be captured. Perceptual similarity, which mimics human visual perception [259], is a better metric for evaluating image similarity and capturing high-level semantic features, essential for model generalization. Hence, we hypothesise, that training MIM with perceptual similarity will enhance the models' output quality and coherence. The focus is on learning high-level relationships between elements in the image, not just low-level details. Thereby reducing the model's vulnerability to masked regions and improving overall semantic content and structure capture

Therefore, to train StereoMAE we utilise a perceptual loss for MIM for learning the required high-level semantic features that represent the scene context (thereby improving its downstream generalisability). Inspired by [94] our perceptual loss comprises of three loss terms: (i) L1 (ii) feature matching and (iii) style transfer losses. L1 loss encourages sparse representations by optimizing absolute intensity values. The feature matching loss compares the feature representations of the reconstructed output and target image, while the style loss measures the difference in statistical distribution of high-level feature activations, capturing image texture and aesthetics. To calculate (ii) and (iii) we utilise a pre-trained feature extractor model i.e. VGG16 (though any arbitrary model can be used).

$$\begin{aligned} L^{I}_{mim} = ||G(I_m) - I||_1 + \delta_f \sum_{j=1}^{J} \frac{1}{N_j} [||\phi^j(G(I_m)) - \phi^j(I)||_1] \\ + \delta_s \sum_{j=1}^{J} \frac{1}{N_j} [||\Psi(\phi^j(G(I_m))) - \Psi(\phi^j(I))||_1] \end{aligned} \tag{6.1}$$

where, I is the original input image, I_m is the masked input image, G is the StereoMAE model, ϕ is the feature extractor model and Ψ is the gram matrix. The loss weights δ_f and δ_s are 0.05 and 40.0, respectively. L_{mim} is calculated for both the left and right views, making the full MIM loss for StereoMAE the following:

$$L_{mim} = L^{I_L}_{mim} + L^{I_R}_{mim} \tag{6.2}$$

where, I_L and I_R are the left and right input views, respectively. MIM training was performed on a mixture of real natural and synthetic scenes (no surgical scenes were used).

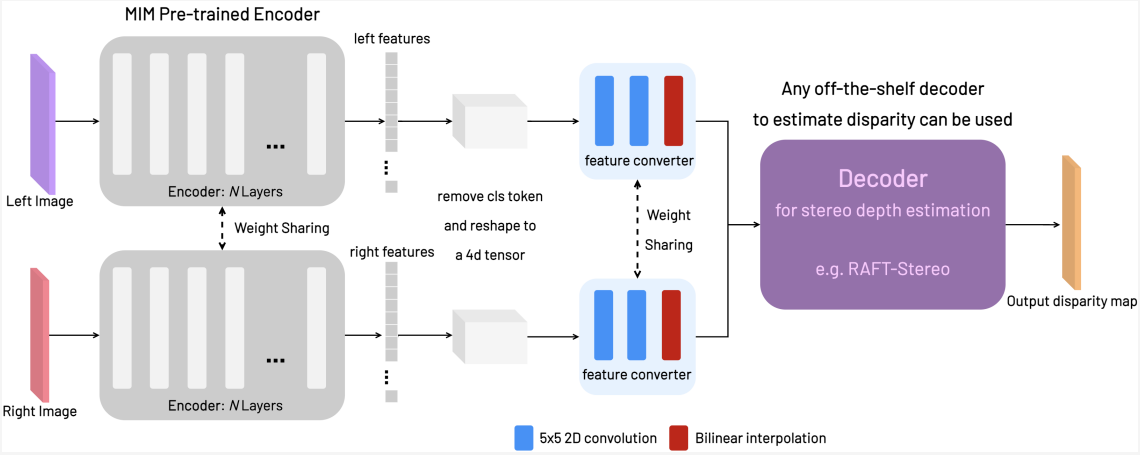


Figure 6.2: The downstream process of supervised stereo depth estimation training. Where the feature extractor encoders pre-trained from MIM, are packaged with an off-the-shelf decoder for depth estimation. Any decoder of user's choice can be used.

6.3.2 Supervised Downstream Fine-tuning

Once the encoders have been pre-trained via MIM in phase (1) we finetune them for disparity estimation. The full model architecture for downstream training is displayed in Figure 6.2. The architecture is designed to be modular such that the pre-trained encoders can be combined with any off-the-shelf decoder of the user's choice. Thereby enabling a smooth transition from MIM to disparity estimation without further modifications. The pre-trained ViT encoders generate features for the (un-masked) left and right input images, and output a 3D tensor. The features have to be reshaped into a 4D tensor for it to be processed by the decoders which are typically composed of convolution layers. However, once reshaped, the positional encodings of the features change. To ensure the decoder focuses on the necessary elements of the feature map, we parse the feature map through a *Feature Converter* block; its purpose is to learn the transfer of MIM-trained features to stereo-disparity features and select the relevant elements needed for the decoder. It comprises of two 5x5 2D convolution layers and a bilinear interpolation module that resizes the tensor to a scale of choice. The scale depends on the type of decoder used. In our experiments we utilise the RAFT-Stereo decoder [129].

To train the final model we utilised a supervised-learning loss comprised of the L1 error between the predicted and ground truth disparity. The RAFT-Stereo decoder generates a sequence of predictions (from its multi-level GRU networks), hence our L1 error is computed over the full sequence of predictions, $[d_p^1, ..., d_p^N]$, with exponentially increasing weights. If a different decoder that outputs a single disparity map is generated then, only the final single output d_p would have been used for the

loss calculation. Considering ground truth disparity is defined as d_{gt}, the loss is defined as

$$L_{sup} = \sum_{i=1}^{Y} \gamma^{Y-i} ||d_{gt} - d_p^i||_1 \tag{6.3}$$

where, $\gamma = 0.9$, is the weighting factor for the loss calculated for each scale and Y is the total number of scales. The supervised training for disparity estimation was only conducted on synthetic scenes (no surgical or natural scenes were used).

6.3.3 Implementation Details

For MIM training, we use the ViT-B architecture [73], trained for 150 epochs on a combination of the following datasets [37][142][215][226][20] (a total of 411,942 stereo image pairs). The input patch size is fixed to 16x32 and we mask 75% of input patches during training. The ViT-B encoder, comprises of 12 transformer layers (N), each with 12 self-attention heads and the final hidden dimension output is of size 768. The decoder architecture comprises of 8 transformer layers (D) each with 16 self-attention heads and the hidden dimension output of size 512. The data augmentation strategies include (1) resizing / cropping image to 224x448, (2) adding random distracting shapes like rectangles of varying size and (3) random colour jittering. We train with a batch size of 8 on 4-Nvidia Tesla GPUs. Adaptive Moment Estimation (ADAM) optimizer with a learning rate of 0.00015, and weight decay of 0.05 (cosine strategy), 40 warm-up epochs and the momentum parameters β_1 and β_2 are 0.9 and 0.95 respectively were used. For downstream training, the pre-trained ViT-B model is used as feature extractor with the RAFT-Stereo decoder architecture, for disparity estimation. The same training strategy as [129] was utilised, though since our model is modular, it can be combined with any downstream stereo-decoder model and training method. For the downstream training of disparity estimation, only the Sceneflow training split [142] was used (a combination of FlyingThings, Monkaa and Driving). Hence, only synthetic data was used for training.

6.4 Results & Discussion

We evaluated the performance of our model on both surgical and natural scenes, despite not training on them. The following datasets were used for testing; SCARED, ETH3D, and Middlebury datasets. To compare our results with other methods in the literature, we calculate the standard metrics for stereo disparity evaluation i.e. average End-Point-Error (EPE) and the percent of pixels with EPE greater than a specified threshold (0.5, 1.0 and 2.0). Specifically for SCARED, we also calculate the mean absolute depth error in mm on test sets 1 and 2. We also exhibit qualitative performance of our MIM pre-training and downstream disparity estimation in Figures 6.3 and 6.4.

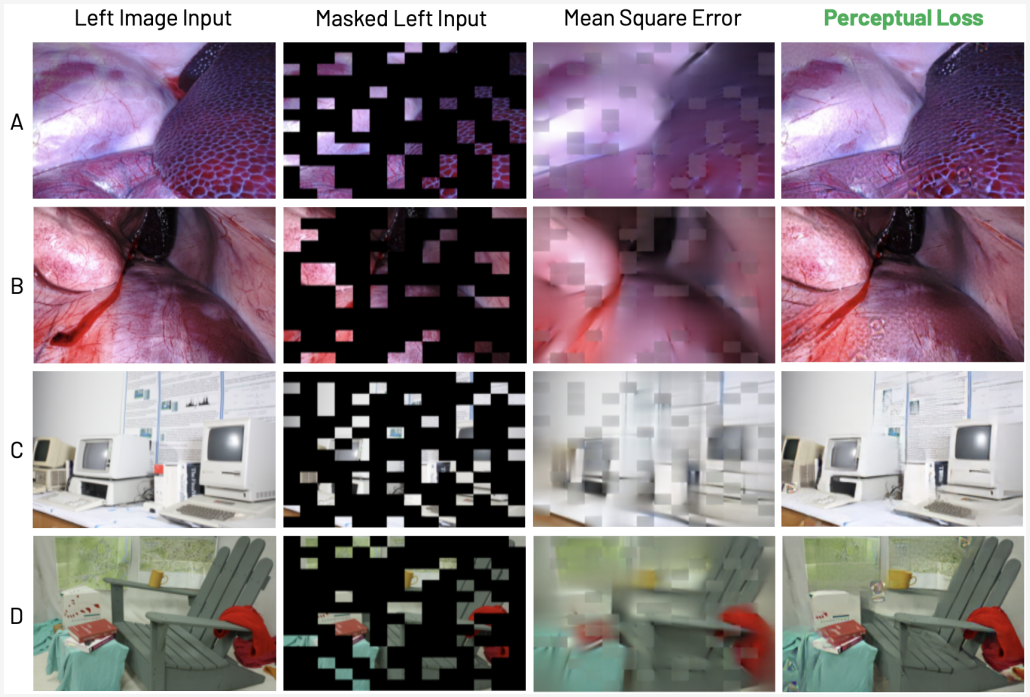


Figure 6.3: Reconstructions of the left images on (A-B) SCARED and (C-D) Middlebury samples. Results of MIM pre-trained StereoMAE via Mean Square Error loss [73] and the proposed perceptual loss shown in 3^{rd} and 4^{th} columns, respectively. The inputs were masked at 75% mask-to-image ratio. StereoMAE was not trained on these datasets.

Figure 6.3 compares the reconstructed outputs of StereoMAE trained via the MSE loss from [73] and the proposed perceptual loss, on unseen datasets. It can be observed that StereoMAE trained with perceptual loss outperforms MSE in all scenes. When using perceptual loss, we observe significant increase in the fidelity of the reconstructed patches with finer textural and structural details. Exhibiting perceptual loss aids the model in learning higher-level feature representations that can generalise to any scene type despite never having observed surgical scenes. When finetuned for disparity estimation, StereoMAE visibly generates sharper depth maps despite only being trained on synthetic data, as shown in Figure 6.4. Specifically, in sample A, StereoMAE generates the fine

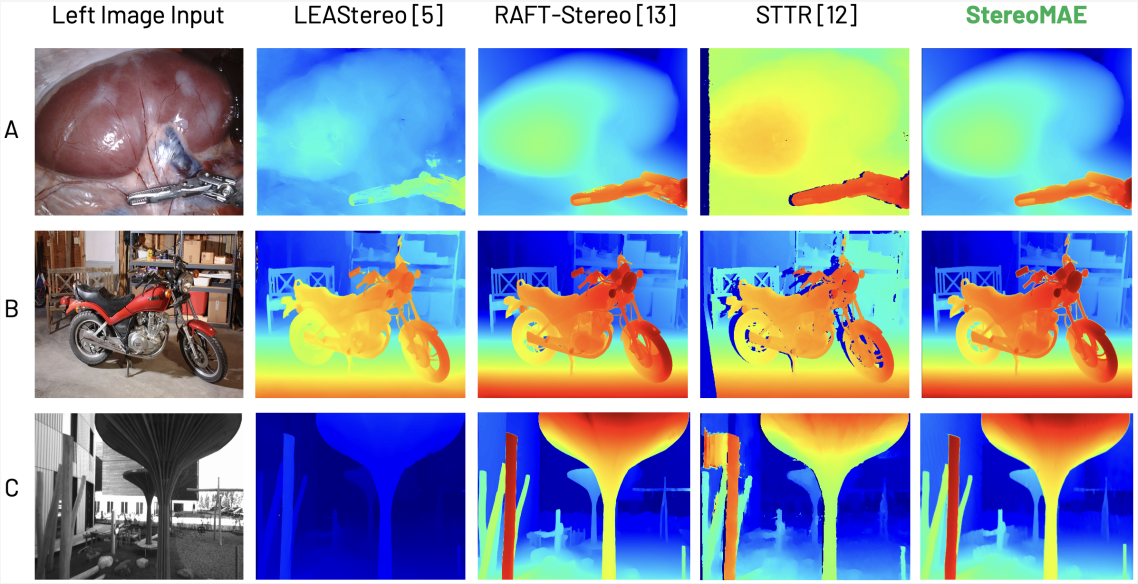


Figure 6.4: Left disparity outputs on (A) SCARED, (B) Middlebury and (C) ETH3D samples. All models were trained only on synthetic data.

Table 6.1: Results of generalisation without fine-tuning on ETH3D, Middlebury, and SCARED datasets. Lower is better for all metrics. Method highlighted in bold is best.

Method	ETH3D				Middlebury 2014 (H)				SCARED Small	
	EPE (pixel)	0.5 px (%)	1 px (%)	2 px (%)	EPE (pixel)	0.5 px (%)	1 px (%)	2 px (%)	EPE (pixel)	3 px (%)
LEAStereo [32]	0.63	22.50	8.54	3.94	3.20	49.44	28.51	16.03	22.62	27.07
STTR [125]	1.29	30.37	11.33	4.85	2.22	40.30	22.00	12.21	6.03	9.52
RAFT-Stereo [129]	0.27	8.10	2.33	0.96	1.12	24.28	14.02	8.43	1.16	4.59
StereoMAE	**0.22**	**6.77**	**2.10**	**0.78**	**0.98**	**22.14**	**11.33**	**7.39**	**0.96**	**3.94**

details on the robotic tool, whereas other methods either fail to achieve the correct disparity range or miss fine details around the edges. Similarly in B and C, StereoMAE generates less holes around the edges, as can be seen in the wheels / handle of the bike in B and the structures in the background in C. Hence, by learning a generalisable feature distribution for stereo image representations and disparity estimation, without any training on datasets from Figure 6.4, StereoMAE creates smooth yet fine disparities.

The quantitative evaluation in Tables 6.1 and 6.2 also shows our model outperforming previous state-of-the-art stereo depth estimation models in Stereo-benchmark datasets. StereoMAE achieves lower EPE across the board on all datasets, despite not being trained on them. Furthermore, as shown in Table 6.2, StereoMAE also sets a new record on surgical scenes, achieving sub-millimeter accuracy in depth estimation. This exhibits the importance of visual representation learning for developing generalisable feature distributions, and still retaining the fine details of disparity estimation from supervised learning. Enabling generalisability to surgical images, without training on

Table 6.2: Mean absolute depth error (mm) on SCARED. Best results are highlighted in bold. None of the models were fine-tuned on SCARED, inference only.

Method	SCARED 2019		SCARED Small
	Testdata 1	Testdata 2	Small 19 [125]
	MAE (mm)	MAE (mm)	MAE (mm)
LEAStereo [32]	3.82	4.51	1.48
STTR [125]	4.14	5.91	11.31
RAFT-Stereo [129]	3.74	4.28	1.01
StereoMAE	**1.66**	**2.01**	**0.92**

them.

6.5 Conclusions

Accurate and generalisable stereo depth estimation is crucial for surgical applications. Supervised learning is effective but limited by the scarcity of ground truth data in surgical settings, restricting their generalisability. Self-supervised methods have issues with scale ambiguity and inaccurate disparity prediction. Our StereoMAE approach demonstrates that by integrating MIM for feature representation learning with supervised depth estimation, achieves the benefits of both methods. Despite training only on synthetic data, StereoMAE shows generalisability to surgical and natural scenes, achieving sub-millimeter accuracy in surgical scenes. This is attributed to robust feature representations learned during MIM pre-training, which enhances the model's performance for subsequent tasks. Future work will explore alternative designs and loss terms to enhance MIM and improve stereo depth estimation.

6.5.1 Learnings & Future Work

This chapter demonstrates that self-supervision primarily improves model representations and generalisability. Ultimately, supervised learning is adequate for stereo depth estimation without impacting the pre-trained model's generalisability. Provided the pre-training strategy yields generalisable results initially. Additionally, I showed that training on simulation data effectively generalises and transfers the model to real-world scenes, albeit natural or surgical. The sharp learning signal from dense simulated ground truth combined with generalisable features from MIM self-supervised training, I achieved sub-millimeter stereo depth accuracy in surgical scenes. Consequently, further improvements to this work can be explored in the following areas.

1. Improve model with vector quantisation for improved discretization of features: In this work I improved MIM by training with perceptual loss, which improved the model's reconstruction quality and thereby learning deeper semantic context of the scene. [155] introduced vector quantisation as a way to decretise features based on the context, allowing a greater separation between objects and context in the scene in the latent space. It has proven to improve reconstruction quality too. This could greatly enhance the model in both MIM pre-training phase and downstream stereo depth estimation phase. I have created a sample model architecture of this shown in Figs 6.5 and 6.6 making experimentation and design easy.

2. Domain randomisation in simulation data for robustness: Only simulation data was used for training downstream stereo depth estimation. Hence, one easy improvement is to try simulations with tougher samples such as domain randomisation [213] and adversarial attacks [137]. This will make the model robust to distractor objects and data drift between simulated and real world scenes, thereby also improving generalisability.

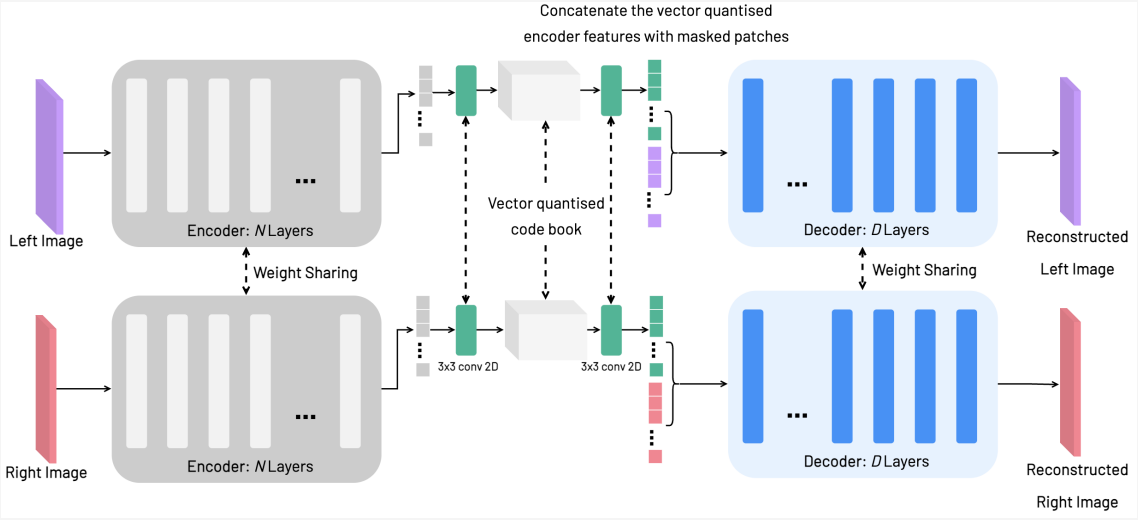


Figure 6.5: Masked Image Modelling (MIM) pipeline for StereoMAE with Vector Quantisation module [155]. Where N and D are the total number of transformer layers that make up the corresponding models.

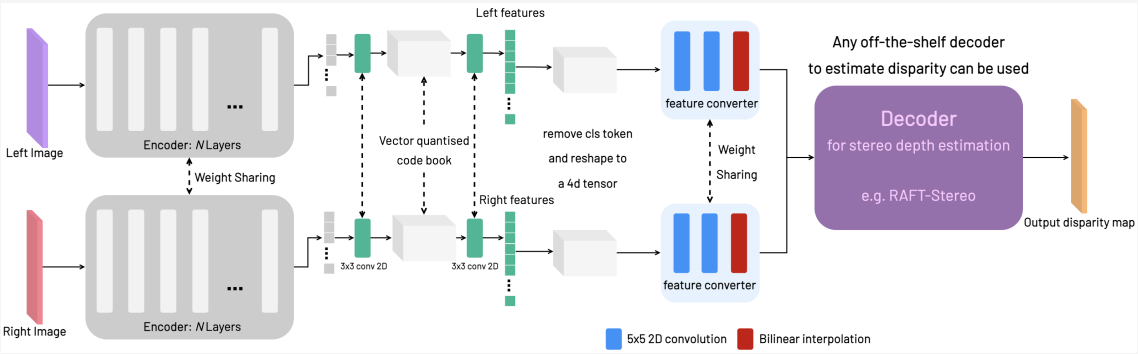


Figure 6.6: The downstream process of supervised stereo depth estimation training using the pre-trained StereoMAE encoder with Vector Quantisation module.

Chapter 7

Boosting Depth with Temporal-Geometric Consistency

In this chapter I address the challenges #3 and #4 (outlined in Introduction): Obtaining temporal consistency in depth maps whilst being robust to dynamic motion and occlusions. This chapter combines all the findings and research conducted in preceding chapters 3-6. Chapters 4-6 describe my journey en-route to attaining sub-millimeter depth accuracy. A key limitation thus far for the depth estimation models, are that they estimate depth individually per-frame in a video. When these depth maps (per-frame) are visualised in a sequence as a video; they flicker, which implies temporally inconsistent geometry. This is further exacerbated due to dynamic motion, deforming anatomical structures, occlusions and more. Solving this is key, for adopting Mixed Reality (MR) and autonomous robotics in applied surgery. In this chapter I expand on chapter 3's findings and use generative modelling to train a model to learn video temporal dynamics and chapter 6 for creating a new type of loss function that enforces multi-view geometric constraints; ensuring predicted geometry is consistent in a sequence. To explore a new pipeline, I focused on a monocular video setup, this simplified model design and allowed for flexibility in exploring new training strategies. The proposed approach establishes a foundation for achieving consistency in depth estimates in Deep Learning (DL) models and is compatible with off-the-shelf depth estimation models. Therefore, enabling future expansion to stereo / multi-view input models, where the same pipeline can be used. In particular the method designed comprises of a 3D-Unet trained via diffusion for learning video dynamics and a Multi Layer Perceptron (MLP) for estimation 3D scene-flow. The innovative loss function jointly optimises both to ensure depth video consistency and enforces multi-view geometry constraints via scene-flow for robustness to dynamic motion. This approach learns video dynamics and maintains depth consistency for objects with high dynamic motion despite camera movement and obfuscation by occlusions. It serves as a precursor for future work, where consistent depth can be adapted for obtaining deformable depth for soft-tissue structures in Minimally Invasive Surgery (MIS).

The work in this chapter is the amalgamation of learnings from chapter 3-6 and has been submitted to the IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) in April 2023.

7.1 Overview

Accurate 3D scene reconstruction from video data requires both temporal and geometric consistency in depth estimation. However, this is difficult to attain due to dynamic changes in scenes such as object and camera motion. Current Structure from Motion (SfM) methods lack the ability to learn the temporal dynamics of a video and result in flickering and geometrically inconsistent depth maps. In this paper, we propose a self-supervised learning algorithm that enforces geometric and temporal consistency in monocular depth estimation, by jointly optimising learning of the camera pose and 3D scene-flow. For learning temporal dynamics, we introduce a denoising diffusion model trained via a novel diffusion loss that iteratively refines depth estimation enforcing multi-view geometric constraints and forward-backward depth consistency. For geometric consistency, we introduce an auxiliary multi-layer perceptron for estimating scene-flow as a continuous differentiable representation to embed local motion priors in 3D. Our method does not require any pre-processing such as using COLMAP to estimate camera pose nor relies on future frames in a video sequence thereby, making it deployable for online real-world use. Qualitative and quantitative performance evaluation on surgical, natural and synthetic scenes shows that our approach sets a new state-of-the-art (SOTA) in both temporal consistency and spatial accuracy in depth estimation.

Video result 1 (deforming sweater test): 21 second video
Video result 2 (moving camera test): 24 second video

Note (for video results above): The proposed approach is untrained on images and videos from my iPhone 13 Pro Max, yet it successfully generalises to new camera intrinsics and extrinsics, yielding sharp depth and less flicker compared to current SOTA methods. This is evident in Video 1, where it reduces flicker artifacts even with a deforming sweater. In Video 2, the phone's front camera is used and, the method demonstrates significant consistency for closer objects; however, an improved off-the-shelf monocular depth model could better resolve depth for objects at greater distances.

7.2 Introduction

The human visual system is capable of perceiving the world in three dimensions by inferring depth from 2D images. However, training machines to achieve similar depth perception capabilities has proven to be a complex and unconstrained problem. Depth perception is fundamental in computer vision and has far-reaching implications for various technological advancements, such as 3D graphics, video stabilization, augmented and virtual reality, robotics, autonomous driving, and other related applications. In Minimally Invasive Surgery (MIS), depth perception allows improved visualisation of the surgical field, robotic navigation, and intraoperative surgical guidance.

Estimating depth from monocular video sequences is an open unconstrained problem. The challenges arise from dynamic moving objects, deformable scenes and camera motion. These factors generate a large space of potential values that can satisfy the depth optimisation strategy, even if the geometry is not consistent. This issue is also prevalent in areas with uniform texture, varying lighting conditions and occlusions. However, these conditions are very common in natural scenes and especially in MIS due to soft-tissue anatomical structures. Current optimisation strategies thus, yield inconsistent depth that fail to adhere to the underlying geometry. Therefore, novel depth estimation models must incorporate prior knowledge of the geometric characteristics and the scene dynamics to adapt to complex scenarios.

Traditional SfM approaches [207] use sparse data representations combined with dense Multi View Stereo (MVS) techniques to find correspondences along epipolar lines, leading to accurate reconstructions when matches are precise. These methods however often produce imprecise / noisy matches due to the aforementioned issues causing drop pixels (holes in the depth maps) and inconsistent depth, requiring regularisation via heuristic smoothness-priors to refine results. Recent deep learning based methods for single-image monocular depth estimation [53, 64, 69, 171, 172, 216, 234, 245, 250] have achieved great performance boost, overcoming these issues. However, they suffer from temporal flickering and inconsistent geometry when per-image depth maps are viewed as a video and the results are only plausible up to a certain scale.

Recent deep learning models for depth estimation from monocular videos can be divided into two categories: implicit and explicit learning methods. Implict methods, address the flickering challenge by implicitly enforcing temporal consistency by designing models with recurrent modules [161, 257]. These methods do not explicitly enforce geometric consistency nor utilise geometric priors. Instead, they rely heavily on learning the data distribution they were trained on. Hence, they suffer from flickering when the inferred videos are divergent from the training distribution. Explicit methods, explicitly enforce temporal consistency by adopting multi-view constraints to learn geometric priors [124, 136, 257], optimising for depth using additional auxiliary information like pose and optical flow. However, these methods require accurate camera parameters and pose to begin with, which is challenging to compute and often fails due to the challenges mentioned earlier. Ad-

ditionally, the presence of dynamically moving and deformable objects or epipolar-aligned motion also result in poor 3D reconstruction performance.

Furthermore, explicit methods require test-time refinement, where the model is constantly trained on new scenes whilst performing inference to allow continuous learning and adaptation [22, 30, 119, 136, 144, 261]. Test-time refinement typically requires substantial computational resources and access to future frames for optimisation. This limits their live use.

Deformable SLAM methods [55, 121, 122, 134, 200] also produce inconsistent results, since they rely on keypoints detected by feature extractor models that are dependent on the implicit data distribution learnt and on accurate camera pose which is difficult to attain.

Hence, a question emerges: can we enforce geometric and temporal consistency even for dynamic scenes? This is possible by simultaneously optimising multiple models to learn temporal dynamics of a scene including camera pose and scene flow as they can mutually enhance each other to overcome the aforementioned challenges. Current depth estimation methods have focused either on implicit or explicit learning. In this work, we propose the first unified approach that combines implicit and explicit learning. We propose to implicitly learn scene motion between frames and explicitly enforce geometric constraints to refine depth for temporal and geometric consistency even in challenging scenarios with moving objects, camera motion, and occlusions.

This paper proposes a novel self-supervised approach for monocular depth estimation. In particular, given initial depth estimates, three surrogate models are trained simultaneously to estimate the 3D scene flow, the camera pose, and refine the initial depth maps, respectively. Advancing the SOTA, the proposed method does not require any pre-processing steps (such as COLMAP inference) to obtain accurate camera pose estimates, thereby enabling inference on videos in the wild. Furthermore, since the surrogate models learn the features for enforcing geometric and temporal consistency during training, the proposed method does not rely on test-time refinement. In our case, test-time training can be performed optionally since our optimisation strategy is self-supervised. Our method therefore, can be used for live applications.

7.2.1 Contributions

Our key contributions are the following:

1. We used for the first time video diffusion to implicitly learn temporal scene dynamics and generate temporally consistent depth maps.

2. We combine the above implicit learning with a loss function that enforces multiview geometric constraints to explicitly refine depth for geometric consistency.

3. We designed a novel depth refinement model the so called, Temporal Consistency Enforcement Network (TCENet) based on the 3D-Unet architecture, augmented with a spatio temporal attention mechanism.

4. We defined a new diffusion loss function to train the TCENet model to generate depth maps with both high and low frequency geometric details.

5. We propose an approach to estimate dense 3D scene flow. For this purpose, a video is represented as a temporal sequence of volumes. A multi-layer preceptron the so called, FlowMLP is trained to predict scene flow as a continuous probability distribution across time.

In our work, we assess temporal coherence in depth by using optical flow warping and scene reconstruction. Thereby showing that by addressing temporal consistency in depth makes model outputs robust to occlusions as well as, simultaneous scene and camera motion. We validated our method qualitatively on the DAVIS [21], MPI-Sintel [20] and SCARED [6] datasets to evaluate its performance across divergent scene types namely, natural, synthetic and medical scenes, respectively. Furthermore, we validated quantitatively the spatial accuracy of our depth maps on the Kitti eigen split [59] and the temporal consistency via optical flow warping on both DAVIS and MPI-Sintel.

7.3 Related Works

7.3.1 Single Image Depth

In the literature, single-image depth estimation deep learning methods are classified into two categories namely, supervised and self-supervised. Supervised methods, attain the best performance as these methods learn to directly regress the output depth values provided a large dataset with corresponding ground truth depth maps are available [46, 53, 172, 239, 245]. Most innovation in this area involves either improving the learning capability of feature extraction models or innovative loss functions for conducting regression. However, their dependence on high quality, ground truth depth during training, limits their generalisability and is subject to availability. Furthermore, changes in the range of depth values of a scene requires re-training, making generalisability to different scenes arduous. For example, since the depth range in MIS (mm) is substantially different from that in natural scenes (m), the adaptability of these depth estimation methods is hindered and transfer learning will have limited effect [251].

Self-supervised techniques for depth prediction optimise photometric consistency, utilizing either stereo images or monocular videos [57, 63, 64, 69, 92, 166, 167, 216, 265]. However, the photometric loss can be minimised even when the geometry is inconsistent, particularly in regions with poor texture. Moreover, such methods struggle to perform well for video frames that are temporally distant due to significant scene appearance changes.

Both supervised and self-supervised methods, estimate depth independently, frame-by-frame. This approach leads to inconsistent estimates over time with flicker artefacts in the resulting depth maps. Moreover, since the epipolar constraint is not applicable in single-view prediction, these methods result in lower accuracy compared to MVS [89, 118, 248] methods for stationary areas.

7.3.2 Multi-View Depth

Multi-view depth methods are classified into two categories namely, MVS and depth from video. MVS involves estimating depth using multiple images taken from arbitrary viewpoints [85, 118, 247]. These methods employ traditional geometry-based principles such as cost aggregation and plane-sweep volume generation which is constructed by warping frames to a reference viewpoint. However, they rely on the assumption of a static scene to apply epipolar geometry constraints during training. Hence, dynamic scenes pose a challenge, resulting in inaccurate depth and camera trajectories, and the exclusion of low-confidence pixels which results in depth maps with holes.

Depth from video essentially solves depth either from single frames [64, 69, 130, 160] or multiple frames [16, 211, 234] in a video by enforcing multi-view constrains to solve the geometry for every target frame. These methods however, face significant degradation when applied to fast-changing

scenes with dynamic objects. Also, it becomes difficult to establish temporal relations between frames in the presence of occlusions, motion blur, and poor texture.

7.3.3 Temporal Consistency

Temporal depth inconsistency occurs when the depth estimate for a given pixel is not stable across frames; resulting in drifts that lead to incoherent outputs. A typical problem amongst all monocular depth estimation models. Numerous methods have been proposed to tackle this in RGB-space, including end-to-end video frame synthesis [225], style transfer [26], or via post-processing [120]. Concurrent to our work, Various methods have been proposed to enforce consistency in the depth space. These methods either achieve their goal implicitly [160, 223, 257] or explicitly [115, 124, 136, 261]. Implicit methods, focus on the design of models with recurrent neural networks to embed temporal dynamics in their weights. However, these methods generate inconsistent results if the distribution of the training data is different from the inference data, limiting their generalisability. Furthermore, they necessitate on large datasets to account for different scenes, object and occlusion types. Lastly, since the consistency is enforced implicitly during training, optimisation becomes unpredictable and thereby limits reproducibility.

The fundamental concept behind explicit methods is to incorporate a temporal consistency loss term which is based on geometric constraints and promotes similar depth values across temporal correspondences derived from the input video, either during training or testing. Most methods build on the formulation of [136] where a single-image depth estimation model is fine-tuned by the 3D re-projection of pixels from one image to the other using optical flow. The re-projection photometric error is reduced over the course of fine-tuning during test time, resulting in a flicker-free depth. The method in [261] was built on this work and trained an additional model to predict scene-flow to account for dynamic moving objects. Key limitations in both works is the requirement of precise camera pose needed as input, which are typically difficult to attain with standard SfM as described in Section 1. Inaccurate poses thus, results in a degrading effect on depth over time in a video. [115] overcame this limitation, by jointly estimating camera pose and depth. However, they do not account for dynamic moving objects like [261]. Additionally, these aforementioned methods train the models via test-time refinement, requiring access to future frames. This limits their online use in real-world applications and is computationally demanding as it requires several forward and backward passes of a model to resolve temporal consistency.

Our method in contrast is a combination of both Implicit and Explicit techniques. It prioritises achieving an accurate prediction with **global geometric consistency** which is explicitly enforced by a geometric loss and learns **temporal smoothness** which is implicitly embedded in the model's features. Both characteristics are crucial for video-based applications. Moreover, our approach differs from previous ones in technical aspects. Instead of relying on test-time refinement we train a separate generative model that optimises the 3D deformable alignment of depth such that the depth

maps are geometrically consistent. Once trained, this model can be deployed in the wild. Since our method is self-supervised, test-time refinement can be optionally conducted. Furthermore, our approach also optimises for camera poses and 3D scene-flow, enabling robust photometric re-projection in challenging videos with high dynamic motion. Hence, our approach is capable of producing dense depth maps even when the scene consists of dynamic motion naturally.

7.4 Methods

7.4.1 Overview

The full pipeline of our method is illustrated in Fig. 7.1. The main steps are: (1) Given an input sequence of frames I^S, a depth map D is estimated from a pre-trained off-the-shelf single-image depth estimation model, MiDASv2 [171] for each frame. (2) Simultaneously, an off-the-shelf Convolutional Neural Network (CNN) pose estimation model estimates the camera pose $[R, T]$ (rotation and translation) given a pair of adjacent frames. (3) Then, the Multi Layer Perceptron (MLP) model, FlowMLP is used to predict the 3D scene flow at time $t+1$ taking as input (p_i^I, z_i^D, t). Where p is the i^{th} pixel in image frame I at time t, z is the i^{th} depth value from the depth map D at time t. Lastly, (4) the diffusion model TCENet refines the initial depth maps D^S and generates new temporally and geometrically consistent depth maps D_c^S.

During training, the camera pose model as well as, FlowMLP, and TCENet are jointly trained via the photometric reprojection error, while the single-image depth model remains frozen. TCENet is additionally trained with auxiliary denoising diffusion for learning video temporal dynamics. FlowMLP is additionally trained with an induced optical flow auxiliary loss.

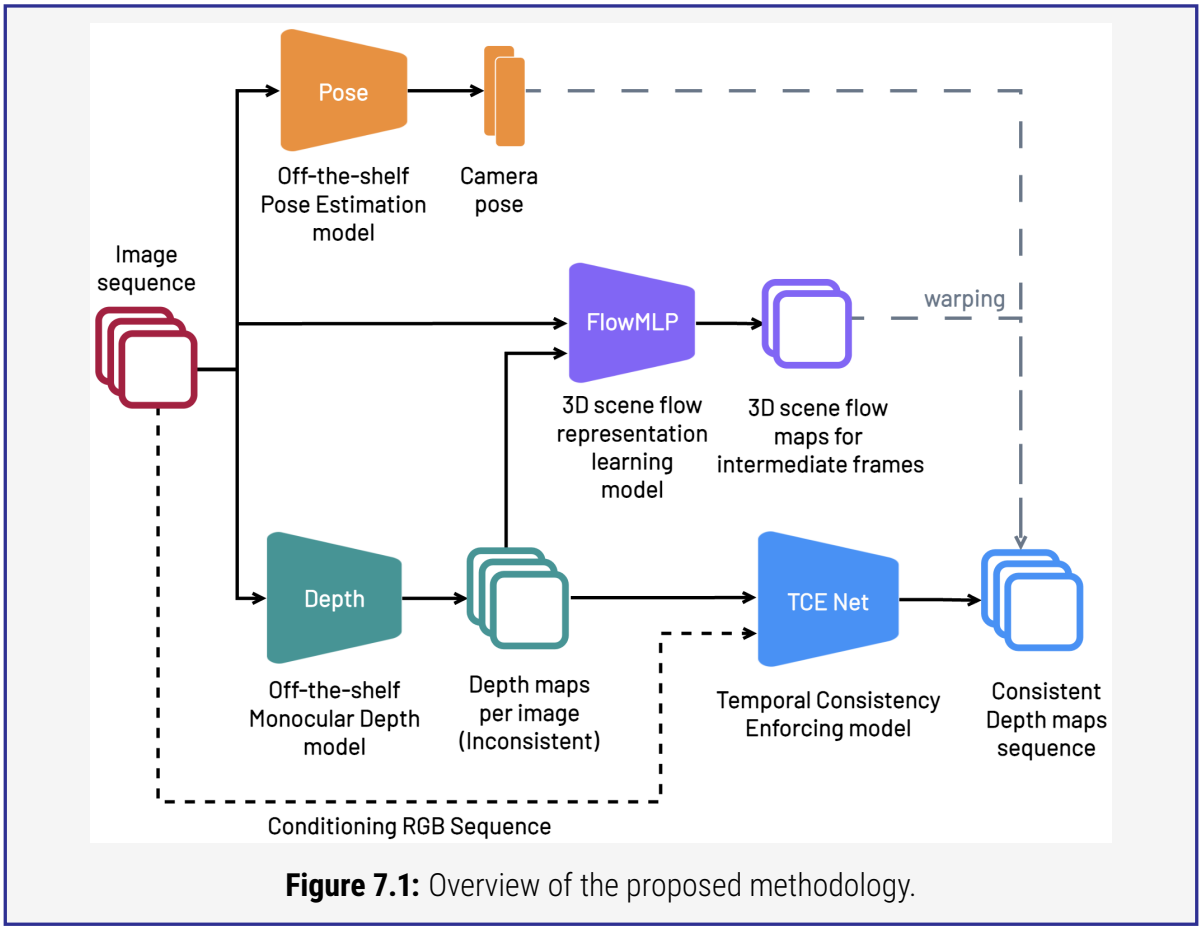


Figure 7.1: Overview of the proposed methodology.

Since, video temporal dynamics and depth frame alignment is embedded in the TCENet's features,

test-time refinement is not a necessity. Hence, once TCENet has been trained, the camera pose and FlowMLP modules can be optionally discarded during inference. Our pipelines which can be utilised in inference, with and without test-time-refinement are displayed in Fig. 7.1 and 7.2, respectively.

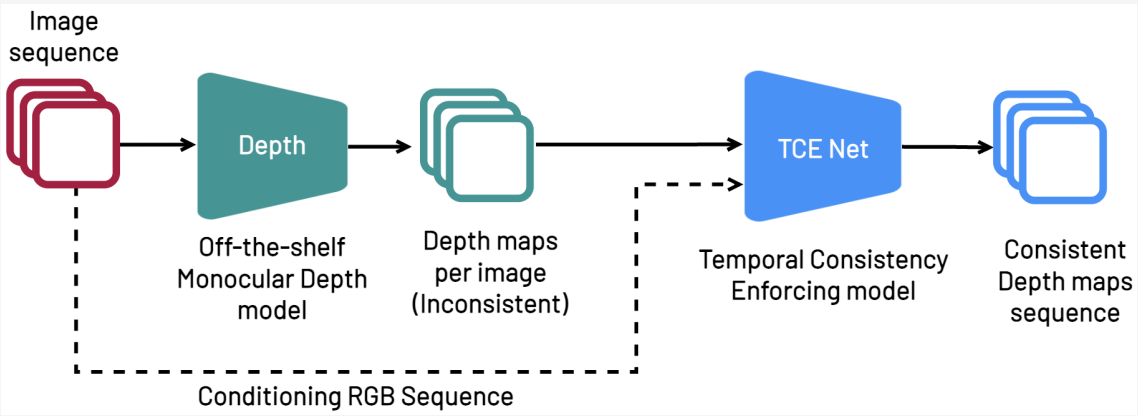


Figure 7.2: Pipeline used for inference without test-time-refinement.

Furthemore, consecutive video frames contain redundant information hence, processing of N sequential frames, will force the model to learn only short term temporal dynamics. To ensure TCENet is robust to scene motion of any range, it must learn from processing both nearby and distant frames. Hence, during training we sample a video sequence as: $[I_{t-(2\times s)}, I_{t-(1\times s)}, I_t]$ where s ($<$8) randomly changes during training iterations. In our method, we process this triplet of frames to train the TCENet. This is the minimum number of frames required to implicitly learn the temporal dynamics via diffusion.

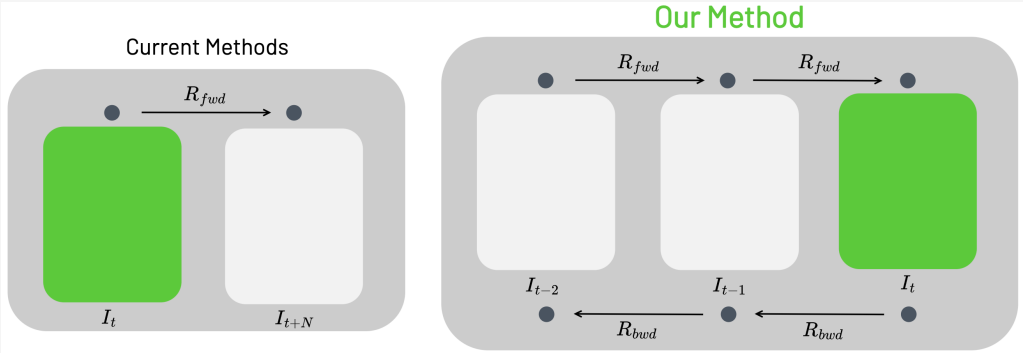


Figure 7.3: Video sampling strategy with reference to the current frame I_t (highlighted in green). R signifies the re-projection function and the subscripts fwd and bwd imply forward and backward re-projection, respectively.

7.4.2 Models

7.4.2.1 Depth, Pose & Optical Flow estimation

Given the triplet frame sequence $[I_i, I_j, I_k]$, we compute (1) **Depth**: for each frame a dense depth map D is computed using the pre-trained monocular depth model (θ_{depth}) proposed in [171]. (2) **Optical flow**: for each frame pair (i,j) and (j, k), we compute dense forward optical flow i.e. $F_{i\to j}$ and $F_{j\to k}$ and backward flow i.e. $F_{j\leftarrow k}$ and $F_{i\leftarrow j}$ using the off-the-shelf pre-trained RAFT model (θ_{flow}) [212]. This is required to perform forward-backward consistency checks and to compute binary occlusion masks $M^F_{i\to j}$, similar to [97]. These masks are created for all frame pairs in both directions. They are used to remove pixels that have forward-backward consistency error greater than 1 pixel. This allows to exclude pixel regions with inaccurate optical flow. The weights of both θ_{depth} and θ_{flow} models are not updated during the training process, they remain frozen. (3) **Camera pose**: we compute relative 6 DoF (degree of freedom) camera pose for each adjacent frame pair both forward i.e. $P_{i\to j}$ and $P_{j\to k}$ and backwards i.e. $P_{j\leftarrow k}$ and $P_{i\leftarrow j}$. For this purpose, we train a small CNN model (θ_{pose}) [31, 68] that estimates $[R, T, K]$, where R, T and K are the rotation matrix parameterised as three Euler angles, the 3D translation vector and the camera intrinsic parameters, respectively, relative to the first input frame. θ_{pose} is composed of 8 2D-convolution layers stacked sequentially and a global pooling layer at the end for generating the final prediction. Regarding the camera intrinsics (K) defined as:

$$K = \begin{pmatrix} f^x & 0 & w/2 \\ 0 & f^y & h/2 \\ 0 & 0 & 1 \end{pmatrix} \tag{7.1}$$

θ_{pose} only predicts the focal length f^x and f^y, along the two optical axes for an image with resolution $h \times w$. We assume that the optical center is at the image center and disregard camera skew and radial distortion. If the camera intrinsics K are known, they can be used rather than estimating them through the camera pose model. This will reduce computational complexity and will expedite the training process.

The optical flow and camera pose models are discarded in inference as they are only used for training TCENet. The three aforementioned models are essential for warping depth bi-directionally to enforce consistency both in the 3D space (using the camera pose) and in the image space (using the optical flow). Our framework is agnostic to the specific architecture utilised for these models. Hence, any arbitrary model of choice can be utilised.

7.4.2.2 FlowMLP: 3D Scene-Flow estimation

Scene-flow is vital for enforcing multi-view geometric constraints for accurate depth estimation of dynamic scenes. State-of-the-art optical flow models like RAFT [212], are prone to errors due to

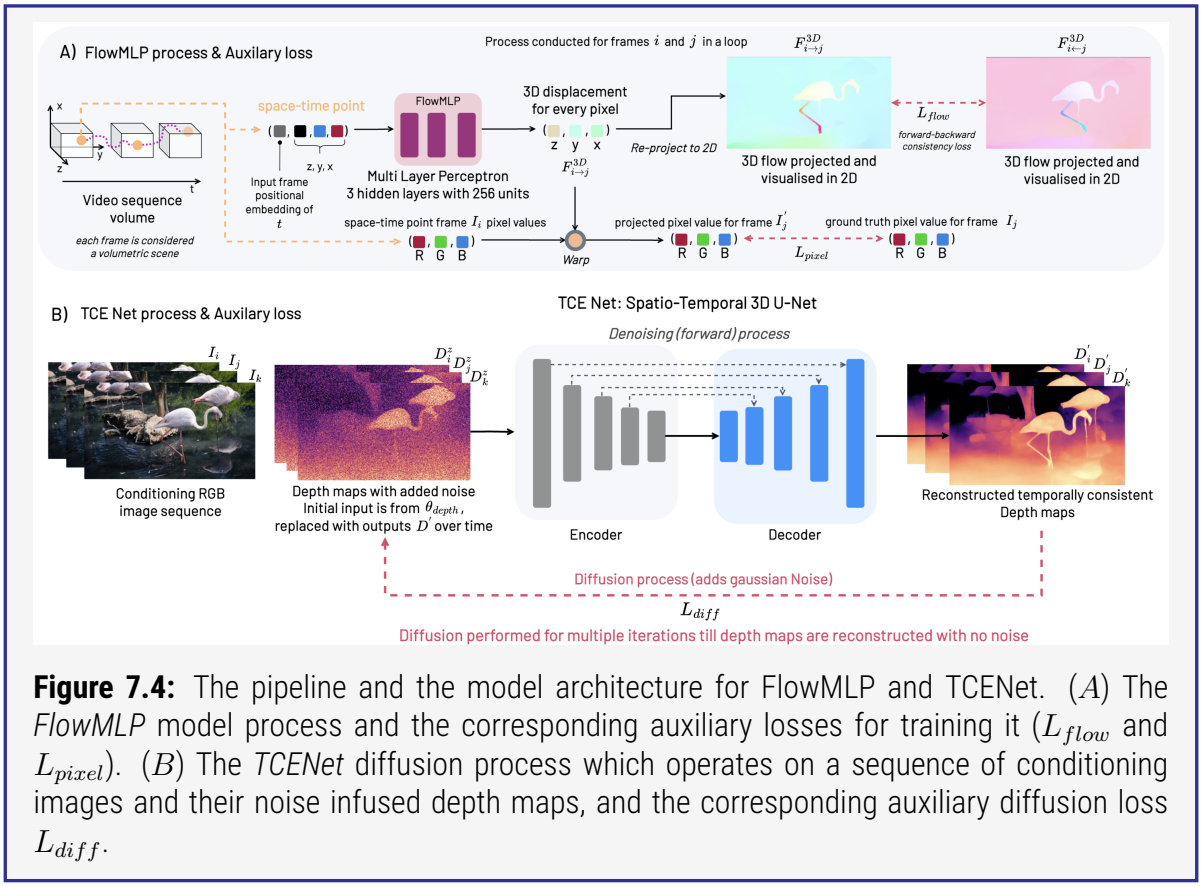


Figure 7.4: The pipeline and the model architecture for FlowMLP and TCENet. (A) The *FlowMLP* model process and the corresponding auxiliary losses for training it (L_{flow} and L_{pixel}). (B) The *TCENet* diffusion process which operates on a sequence of conditioning images and their noise infused depth maps, and the corresponding auxiliary diffusion loss L_{diff}.

factors including scene occlusions and scene illumination variations. Furthermore, the underlying assumption of a constant field of motion does not hold in practise, especially in the presence of dynamic motion.

Hence, inspired by [126, 261] we designed a MLP the so called, FlowMLP, to implicitly learn scene flow representations for estimating precise 3D scene displacements between video frames. This approach is more robust to the aforementioned issues which occur when estimating only 2D scene displacements. The FlowMLP pipeline is displayed in Fig. 7.4A.

In our framework, a video is represented as a temporal sequence of volumes. More specifically, as a collection of 3D data that evolves over time. Each data point consists of an (x, y) position on a 2D image and a depth value (z), at time step (t). As illustrated in the first component of Fig. 7.4(A), every point (x, y, z, t) has its own unique 3D motion path (shown as pink dotted line), which represents how the point moves within the 3D space along time. FlowMLP is a specialised model that has been designed to accurately estimate the 3D displacement for any given scene point.

To improve the representations learnt, we do not use the same architecture as [126, 261]. We modify FlowMLP to improve its performance via **continuous representation learning**. Inspired by [198] we use the sinusoidal representation network (SIREN) with periodic activation functions as our FlowMLP model architecture. Traditional models that use discretised activation i.e. ReLU

[126, 261], can lead to discontinuities in the learned function and thereby loss of information and inaccurate data representations due to the piecewise-linear nature of the function. Our FlowMLP uses periodic activation function (sinusoidal) which learns better implicit neural representations because it allows for the smooth encoding of spatial and temporal information using a continuous geometric function. Furthermore, it enables the network to represent a wide range of frequencies, capturing both high and low-frequency features in the video data.

Our FlowMLP is thus, a SIREN architecture comprising of 3 hidden layers with 256 units each. FlowMLP takes a 4D input (x, y, z, t) and generates a vector of the 3D displacement of the point along the (x, y, z) axis, showing how the point moves from frame t to the next $t + 1$. Prior to parsing inputs through FlowMLP, positional encodings are created for them [148]. This involves mapping the discrete input co-ordinates into a high dimensional space that enables FlowMLP to approximate a high frequency continuous function. The training losses for optimising FlowMLP are discussed in Sec. 7.4.3.2.

7.4.2.3 TCENet: Temporal Consistency Enforcement

In our method, the Temporal Consistency Enforcement Network (TCENet) has been designed to: (1) align depth maps to be temporally consistent, (2) learn the temporal dynamics of a scene and (3) enforce multi-view geometric constraints in the depth predictions. Training this model to enforce temporal consistency overcomes the issue of requiring test-time-refinement. This has the following advantages for our method: (1) not requiring greater computational resources to run in inference, (2) not requiring future frames for optimisation and (3) can be deployed online,for depth estimation of real-world scenes. To ensure that TCENet is generalisable to any scene type, it must learn feature representations that capture the temporal dynamics of a scene. This is accomplished through generative modeling, which seeks to represent the underlying feature distribution of a video.

Recently, diffusion probabilistic models have emerged as a potent class of generative models capable of generating high-quality images [42, 183] and videos [79, 196] with a remarkable level of detail. This is because they learn a series of conditional distributions that progressively transform and reconstruct an input sample corrupted with Gaussian noise into a refined sample that matches the original data distribution as shown in Fig. 7.5. They capture both short-term and long-term data dependencies, including prediction uncertainty.

Our TCENet is therefore a diffusion model that learns temporal scene dynamics as a probabilistic distribution which evolves over time. This makes it robust against unpredictable scene dynamics. The standard architecture for many diffusion models in the literature is a U-Net [182] which comprises of an encoder-decoder architecture with skip connections between the intermediate layers. However, U-Nets are not designed for operating on videos, and discard temporal information. Hence, we propose a 3D U-net variant capable of learning spatio-temporal features. The full pipeline of TCENet is shown in Fig. 7.4(B) and the individual components of the network are

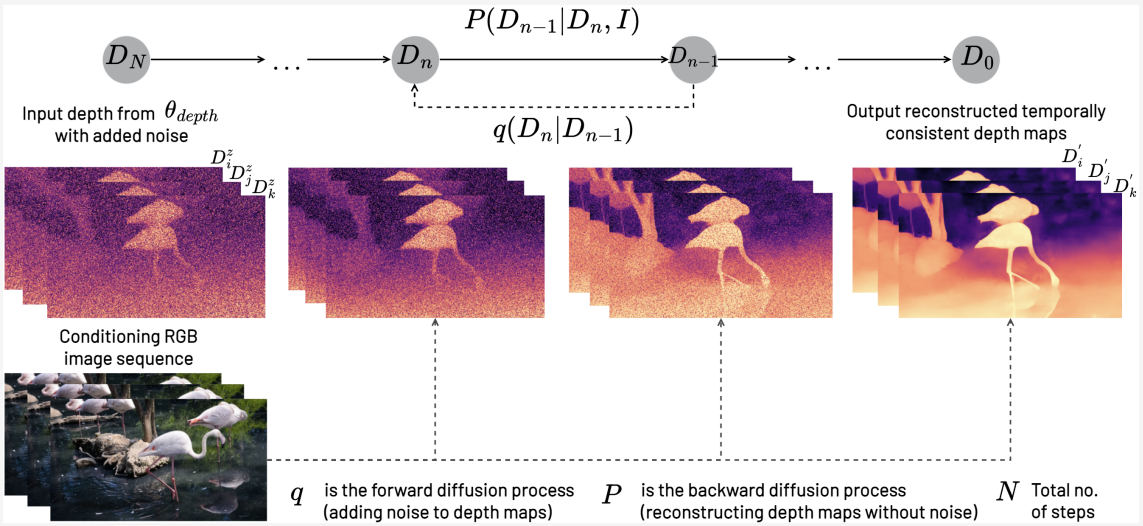


Figure 7.5: The TCENet reverse denoising process.

illustrated in Fig. 7.6.

Firstly, we replace all 2D convolution layers of U-net with 3D convolution layers of filter size 1x3x3. Secondly, to give TCENet greater learning capacity, inspired by [183], every layer in the 3D U-Net is composed of a residual block, as illustrated in Fig. 7.6. We re-design this residual block to encode spatio-temporal features via (1) using 3D convolution layers in place of 2D and (2) adding multi-head spatial and temporal self-attention modules such that the model can focus on key aspects of the feature maps via knowledge distillation, both spatially and temporally. Self-attention is computed as in [220]. However, we reshape the input feature maps to ensure attention is learnt across the required dimension. Consider an input feature map of dimension $b \times c \times n \times h \times w$ where b is batch size, c the number of channels, n the number of frames, h the frame height, and w the frame width. For spatial self-attention, the feature map is reshaped to $(b \cdot n) \times c \times h \times w$, where each frame n in the sequence is processed separately as a spatial sample. For temporal self-attention, the feature map is reshaped to $(b \cdot h \cdot w) \times n \times c$ where features along the n axis are processed for calculating attention over the sequence. The spatial features along h and w axis are treated as separate samples.

7.4.3 Loss Function

In our loss function estimation, we process a triplet frame sequence, $[I_i, I_j, I_k]$, where the loss is computed for adjacent frame pairs i.e. (i, j) and (j, k), and then it is summed. For each pair, the scene flow $F^{3D}_{i\to j}$ and $F^{3D}_{j\to k}$ is estimated using the FlowMLP and the depth maps $[D'_i, D'_j, D'_k]$ are estimated with the TCENet.

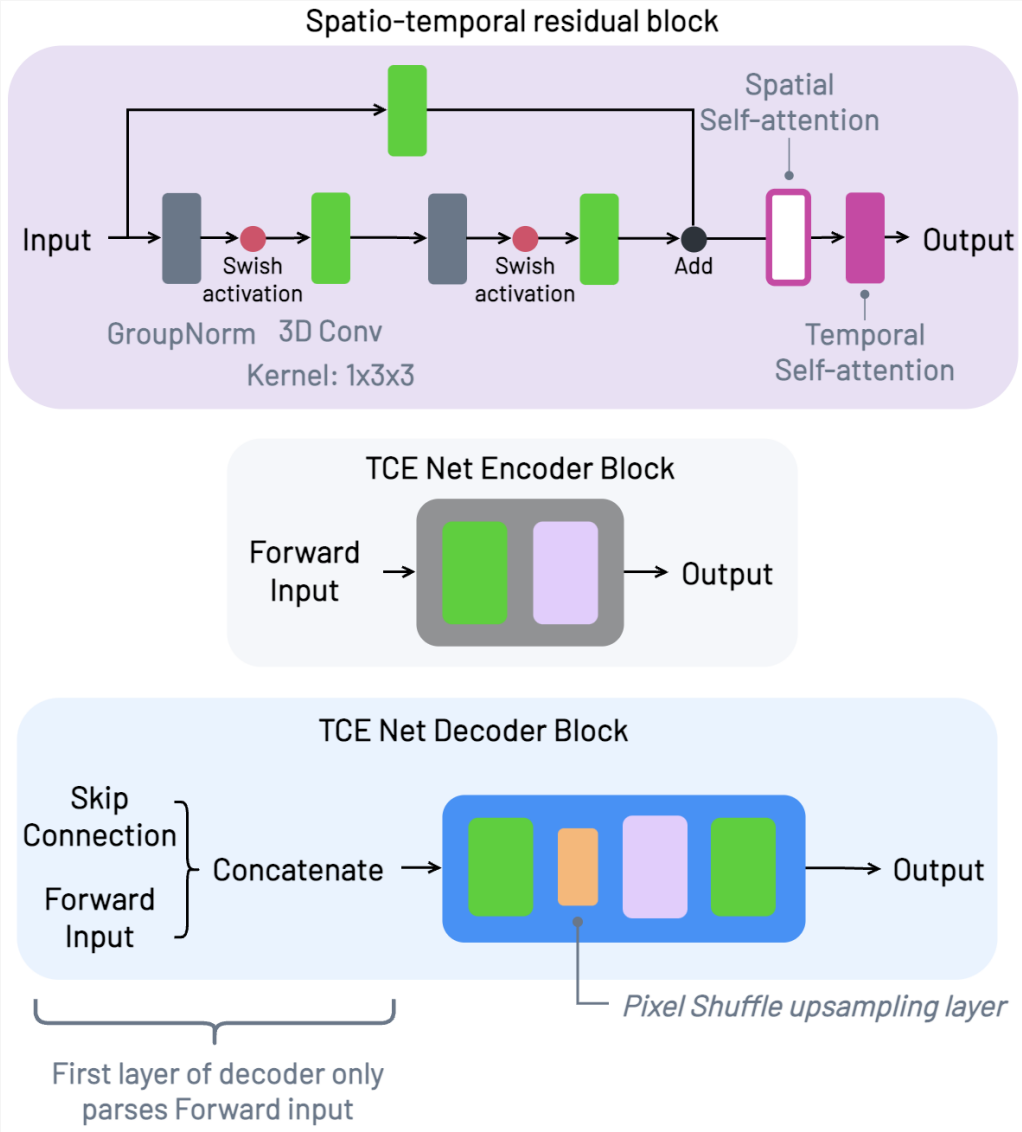


Figure 7.6: The key components of TCENet 3D-Unet architecture.

7.4.3.1 Geometric Loss

The geometric consistency loss, L_{geo}, is based on the principle that the 3D point that corresponds to pixel p_i in frame I_i, when displaced by flow $F^{3D}i \rightarrow j$, should be projected to the corresponding pixel p_j in the adjacent frame. As the re-projection of a pixel from one frame to another requires consistent depth information, using the pixel re-projection error as a loss function enforces the estimated depth maps $D^{'}$ to be geometrically consistent across the video. Hence, our geometric loss is divided into two components namely, the photometric image-space loss L_{pe} and the disparity loss L_{disp}.

Consider a pixel p_i in frame I_i which can be projected to point x_i^p into the 3D co-ordinate system of the ith camera as:

$$x_i^p = R_i^{\top}(D_i^{'}(p_i)K^{-1}\tilde{p_i}) + t_i \tag{7.2}$$

where, $\tilde{p_i}$ is the 2D homogeneous pixel coordinates given as $[p_i^x, p_i^y, 1]^\top$. K, R and t are the camera intrinsics matrix, rotation matrix and translation vector, respectively estimated by the θ_{pose} model. The scene flow field $F_{i\to j}^{3D}$ that defines the 3D displacement of this point towards the next adjacent frame (I_j) in the triplet can be computed via FlowMLP:

$$F_{i\to j}^{3D}(p) = FlowMLP(x_i^p, i) \tag{7.3}$$

This 3D point, x_i^p can thus, be transformed onto the camera co-ordinate system of the next adjacent frame I_j as:

$$x_{i\to j}^p = x_i^p + F_{i\to j}^{3D}(p) \tag{7.4}$$

The above 3D point is then reprojected on the image plane in frame I_j as:

$$p_{i\to j}^{'} = \pi(K_j x_{i\to j}^p) \tag{7.5}$$

where, π signifies the projection operation defined as $\pi([x, y, z]^\top) = [\frac{x}{z}, \frac{y}{z}]^\top$. The objective of L_{pe} is thereby to enforce the position of the reprojected pixels ($p_{i\to j}^{'}$) and the corresponding ground truth pixels in frame I_j to be the same. Since the location of the reprojected pixels depends on the depth maps $D^{'}$, this objective enforces the depth maps to be temporally consistent.

$$L_{pe}^{i\to j}(p_i) = ||p_{i\to j}^{'} - p_{i\to j}||_1 \tag{7.6}$$

where, $p_{i\to j}$ is the pixel correspondence in frame I_j estimated using the 2D optical flow displacement as $p_{i\to j} = p_i + F'_{i\to j}(p_i)$. The 2D optical flow $F'_{i\to j}$ is estimated by the θ_{flow} model

However, to be certain L_{pe} is calculated only for valid pixels where the optical flow prediction is accurate, we utilise the binary occlusion mask $M_{i\to j}$ to select valid pixels where the forward-backward consistency error value is 0 (as discussed in Sec. 7.4.2.1). This ensures only those pixels with consistent flow estimates are considered for our loss. This is mathematically defined as:

$$L_{pe}^{i\to j} = \Sigma_{i,j}\Sigma_{p\in[M_{i\to j}=0]} L_{pe}^{i\to j}(p_i) \tag{7.7}$$

The disparity loss L_{disp} measures the consistency between the inverse depth at point $x_{i\to j}^p$ and the inverse depth at the corresponding point in frame j. This is defined mathematically as:

$$\begin{aligned} D_{i\to j}^{'}(p_i) &= |K_j R_j^\top (x_i^p + F_{i\to j}^{3D}(p) - t_j)|_z \\ L_{disp}^{i\to j}(p_i) &= ||(D_{i\to j}^{'}(p_i))^{-1} - (D_j^{'}(p_{i\to j}))^{-1}|| \\ L_{disp}^{i\to j} &= \Sigma_{i,j}\Sigma_{p\in[M_{i\to j}=0]} L_{disp}^{i\to j}(p_i) \end{aligned} \tag{7.8}$$

Where z is the depth value of a 3D space-point i.e. $z = |[x, y, z]|_z$. The full geometric loss is

defined as follows:

$$L_{geo} = \lambda(L_{disp}^{i\to j} + L_{disp}^{j\to k}) + (L_{pe}^{i\to j} + L_{pe}^{j\to k}) \tag{7.9}$$

where, λ is a weighting coefficient set as $\lambda = 0.1$. The TCENet, FlowMLP and θ_{pose} models are trained jointly with L_{geo}.

7.4.3.2 Scene Flow Auxiliary Loss

For FlowMLP to estimate 3D scene-flow, videos are considered as a temporal sequence of volumes. Each volume and the scene-flow field for each point in it ARE distinct. Hence, FlowMLP requires training on each video sequence separately, which results in unique FlowMLP model weights for each video. This can be a long arduous process if trained with L_{geo} alone. This is because L_{geo} essentially enforces temporal consistency in the 3D space, which can be inherently ambiguous in the presence of occlusions caused by object motion (analogous to 2D optical flow ambiguity). Naively training FlowMLP initialised with random weights from ground up, produces inconsistent flow and depth values which hinders optimisation. Hence, prior to training with L_{geo} we pre-train FlowMLP separately with the two auxiliary losses namely, L_{pixel} and L_{flow}. We call this pre-training, "warm-up" phase which mitigates the aforementioned errors and ensures FlowMLP trains efficiently and performs well post warm-up.

Firstly, L_{pixel}, is the same as the photometric loss L_{pe} in Eq. 7.6. The only exception is that the 3D point x_i^p created in Eq. 7.2 is not created using the depth $D^{'}$ predicted from TCENet. Instead, we use depth D estimated from the frozen θ_{depth} model to create it. Despite the depth D not being temporally consistent nor trained, it is a sufficient prior to train FlowMLP. Then, the steps from Eq. 7.3 to Eq. 7.6 are followed to calculate the photometric error L_{pixel}.

To deal with occluded regions in the scene, we introduce L_{flow}. This loss regularises the scene flow learnt from L_{pixel}, by enforcing forward-backward scene flow consistency. Its aim is to encourage the forward correspondence of the 3D points $x_{i\to j}^p$ and x_j^p, to be consistent with the backward correspondence of the 3D points $x_{i\leftarrow j}^p$ and x_i^p. This enforces forward and backward scene flow consistency, thereby minimising ambiguities caused by occlusion and motion boundaries around moving objects. Similar to Eq. 7.7, the forward-backward flow consistency is calculated for regions where the binary occlusion mask M consists of forward-backward consistency error value of 0. Mathematically, this loss is defined as:

$$\begin{aligned} x_{i\to j}^p &= x_i^p + F_{i\to j}^{3D}(p_i) \\ x_{i\leftarrow j}^p &= x_{i\to j}^p + F_{i\leftarrow j}^{3D}(x_{i\to j}^p) \\ L_{flow} &= \Sigma_{i,j}\Sigma_{p\in[M_{i\to j}=0]}||x_{i\leftarrow j}^{'p} - x_i^p||_1 + \\ &\quad ||x_{i\to j}^{'p} - x_j^p||_1 \end{aligned} \tag{7.10}$$

The full auxiliary loss to pre-train FlowMLP during the warm-up phase is therefore defined as:

$$L_{aux}^{flow} = L_{pixel} + L_{flow} \tag{7.11}$$

This loss aids in the initial stages of training FlowMLP to learn the 3D geometry of the scene correctly rather than relying on L_{geo} alone where initial estimates are noisy and inaccurate. Hence, L_{aux}^{flow} is utilised during the warm-up phase and the initial 5 epochs when conducting training with L_{geo}. When training with L_{geo}, L_{aux}^{flow} linearly decays to zero at which point L_{geo} is the dominant loss function used for training.

7.4.3.3 Diffusion Auxiliary Loss

To enable TCENet to produce consistent depth maps without the need for test-time refinement, it is vital that it learns not only the geometric consistency (enforced by L_{geo}), but also the temporal dynamics of the scene. This can be done by using diffusion to embed the 3D motion priors in the model's weights.

Fig. 7.5 displays the diffusion process for TCENet. Mathematically, TCENet takes as input a sequence of depth maps D from θ_{depth} and a conditioning signal which in our work is a sequence of the corresponding RGB frames I. It estimates a conditional distribution $P(D'|I)$ over the sequence of depth maps. We define the diffusion auxiliary loss as the denoising diffusion function. This is typically used for generative tasks and in our work we re-model it for depth estimation. First, for a number n of diffusion steps, a Normally-distributed noise q is iteratively added to D_{n-1} to obtain D_n, as shown below:

$$q(D_n|D_{n-1}) := \mathcal{N}(D_n|\sqrt{\tilde{\gamma_n}}D_{n-1}, (1-\tilde{\gamma_n})I) \tag{7.12}$$

This is known as the forward process, which follows the Markovian chain. γ_n is the variance following a linear schedule, where $n \in [1, ..., N]$, with N being the total number of steps in the diffusion chain. The backward process involves simply for the TCENet to learn to denoise iteratively, over n steps. This is defined mathematically as:

$$P(D^{'}_{n-1}|D_n, I) := \mathcal{N}(D^{'}_{n-1}; TCENet(D_n, n, I), \sigma_n^2 I) \tag{7.13}$$

where, σ_n^2 denotes the transition variance which remains fixed. Hence, the diffusion focuses on learning the means of the backward process iteratively, i.e., $D^{'}_N \rightarrow D^{'}_{T-\Delta} \rightarrow ... \rightarrow D^{'}_0$, where $D^{'}_0$ is the final set of depth maps. Hence, the diffusion auxiliary loss L_{diff} is simply the L2 distance between the actual noise added and the predicted noise. It optimises a re-weighted variational bound on the maximum likelihood objective, which is defined as:

$$L_{diff} = ||D_{n-1} - TCENet(D_n, n, I)|| \tag{7.14}$$

where, D_{n-1} is calculated from the forward process in Eq. 7.13, by performing the forward pass of TCENet for n steps.

Additionally, in the early stages of training, we aim to improve the spatial quality of depth estimates and prevent the diffusion loss from reproducing the inputs D and converging to a local minimum. To achieve this, we calculate an L1 loss between the generated depth maps $D^{'}$ and the enhanced depth maps obtained using the pre-trained (frozen) MergeNet model [147] as θ_{merge}. The MergeNet model enhances monocular depth by merging high-frequency details from high-resolution depth maps and consistent structure from low-resolution depth maps. Thus, leading to better depth estimations. The L_{depth} loss is then defined as:

$$L_{depth} = ||D^{'} - \theta_{merge}(D)||_1 \tag{7.15}$$

This loss stabilises the diffusion loss and ensures TCENet generates depth maps with both high-frequency details (typically found in edges of distant objects) and low-frequency details (shape and texture of objects near the camera). Furthermore, it ensures that the inputs aren't simply regurgitated by the model. The full diffusion auxiliary loss is defined as:

$$L_{aux}^{depth} = L_{diff} + \beta L_{depth} \tag{7.16}$$

where, β is the weighting factor that has a linear decay from $[1.0, 0.05]$. Hence, TCENet during training is trained on $L_{aux}^{depth} + L_{geo}$, ensuring that the generated depth maps are spatially sharper, temporally consistent and robust to dynamic changes in the scene. Over time, L_{depth} acts as a regulariser and TCENet focuses on the optimisation of L_{geo} and L_{diff}. This improves depth estimation by leveraging the video dynamics, which captures the temporal consistency across frames, and by refining the probabilistic distribution of the depth values.

7.4.4 Implementation Details

The foundational training dataset employed in our work consists of 60 and 23 training video sequences from the DAVIS [21] and the MPI-Sintel [20] databases, respectively. These video sequences include natural scenes with significant dynamic motion and occlusion, captured by cameras in uncontrolled environments. We employ the DAVIS and MPI-Sintel datasets to quantitatively assess the temporal consistency of our model through optical flow backward warping and photometric re-projection error, based on L_{pe} (Eq. 7.7) (discussed further in Sec. 7.5).

Additionally, to gauge the spatial accuracy of the depth maps generated by TCENet, we train our method from scratch on the Kitti raw eigen split [59] database for a fair comparison against other

monocular depth techniques on the Kitti benchmark. Lastly, we also present qualitative results on MIS data, namely the SCARED [6] database, to examine the domain adaptability of our approach in substantially different settings.

Before fully training the proposed pipeline using the geometric loss L_{geo} (Eq. 7.9) we first pre-train both the FlowMLP and TCENet models for stability in optimization. This is because beginning the training from ground up leads to noisy and random estimates for both depth and secen-flow, which hinders the optimization of L_{geo}. Firstly, we pre-train FlowMLP using the auxiliary loss L_{aux}^{flow} on all training video sequences, save the weights, and later reload them for each unique video sequence during the temporal consistency training for fine-tuning. Secondly, we pre-train TCENet using the auxiliary loss L_{aux}^{depth} on training sequences from YouTube-VOS [244] to prevent the diffusion from merely generating noisy depth maps or duplicating the input. The pre-trained TCENet, along with the pre-trained FlowMLP, are subsequently trained further using L_{geo} to ultimately establish temporal and geometric consistency in depth maps throughout the video sequences.

All input images undergo a resize in width equal to 384 pixels, while maintaining the original aspect ratio of the video, adhering to the standard pre-processing method akin to that in [136]. During the pre-training of both FlowMLP and TCENet, we employ the Adam optimizer with a weight decay of 0.05 (cosine strategy), 40 warm-up epochs, and momentum parameters β_1 and β_2 set to 0.9 and 0.95, respectively. The learning rates for FlowMLP and TCENet pre-training are 0.0015 and 0.000015, respectively. For temporal consistency fine-tuning with L_{geo}, the same optimizer configuration is applied to FlowMLP, TCENet, and θ_{pose}.

All experiments were carried out on a single computing instance with 8 Nvidia A100 GPUs. FlowMLP underwent pre-training for 20 epochs per video sequence, taking around 30-40 minutes. Hence, full training time across all video sequence was 2 days. TCENet was pre-trained for a total of 30 epochs on all YouTube-VOS sequences, requiring approximately 3 days. The fine-tuning process using L_{geo} spanned 30 epochs on all DAVIS and MPI-Sintel video sequences and reached convergence in 3 days. In every experiment, TCENet utilised triplet frame inputs, but can accommodate and learn from varying frame quantities. TCENet post-training (during inference), performs at 18 fps on a single Nvidia A100 GPU.

7.5 Results & Discussion

In this section, we present both quantitative and qualitative comparison of our approach with SOTA methods for video consistency in depth estimation. We also compare the spatial accuracy of our generated depth maps with other video based monocular depth estimation methods to validate the improvement in both geometric and temporal depth estimation. Finally, we compute temporal consistency metrics using image reconstruction error based on L_{pe} (Eq. 7.7) on DAVIS and MPI-Sintel datasets. However, the proposed L_{pe} calculates L1 error between the generated pixels and the ground truth, which holds limited semantic significance as it merely compares pixel intensities and is thus inadequate for capturing intricate structures within an image. Consequently, we supplement L1 error reporting by adapting L_{pe} to compute structural similarity index (SSIM) [232], peak signal-to-noise ratio (PSNR) [82], and Inception Score (IS) [184] to compare higher semantic details akin to perceptual similarity metrics. Hence, the temporal consistency metric M_{pe}^{TC} is defined as:

$$M_{pe}^{TC} = metric(I'_{i\to j}, I_{i\to j}) \tag{7.17}$$

where $metric$ is the respective L1, SSIM, PSNR and IS function. I' is the reconstructed output from TCENet and I is the ground truth image frame. The pixel correspondences in frame j are estimated using forward warping with the optical flow estimated from θ_{flow}. All metrics are computed between frame pairs i warped to j, sequentially on all frames of a video sequence.

7.5.1 Qualitative Evaluation

We show qualitative comparison with other SOTA methods on DAVIS and MPI-Sintel in Fig. 7.7 and on MIS data on SCARED in Fig. 7.9. Furthermore, we investigate how the spatial details in the output depth maps generated by TCENet is affected when different θ_{depth} models are used; specifically the default MiDASv2 [171] and LeRes [250] as shown in Fig. 7.8. For all the showcased results, TCENet did not undergo test-time training. Therefore, the displayed outcomes are standard outputs inferred from TCENet using the pipeline illustrated in Fig. 7.2. All video samples in Fig. 7.7 comprise of dynamic moving objects, moving camera. Sample B specifically consists of motion blur as well, making it a challenging sequence for consistency evaluation.

Spatial Quality Evaluation: As shown in Fig.7.7, TCENet produces considerably more precise and sharper depth maps around delicate edges and objects in all video samples. For example, in sample A, the latch on the horse and the hurdle fence being jumped over by the horse are depicted with greater accuracy. Particularly for the hurdle fence, TCENet generates accurate depth information for the gaps within the fence structure, while other models either create blobs for the entire structure or, in the case of θ_{depth}, fail to resolve the depth in that region. A similar observation can be made in sample B, where TCENet renders sharp details on the grass, shrubs, and trees despite the presence of motion blur, while other competing techniques produce blurred and smoothed depth

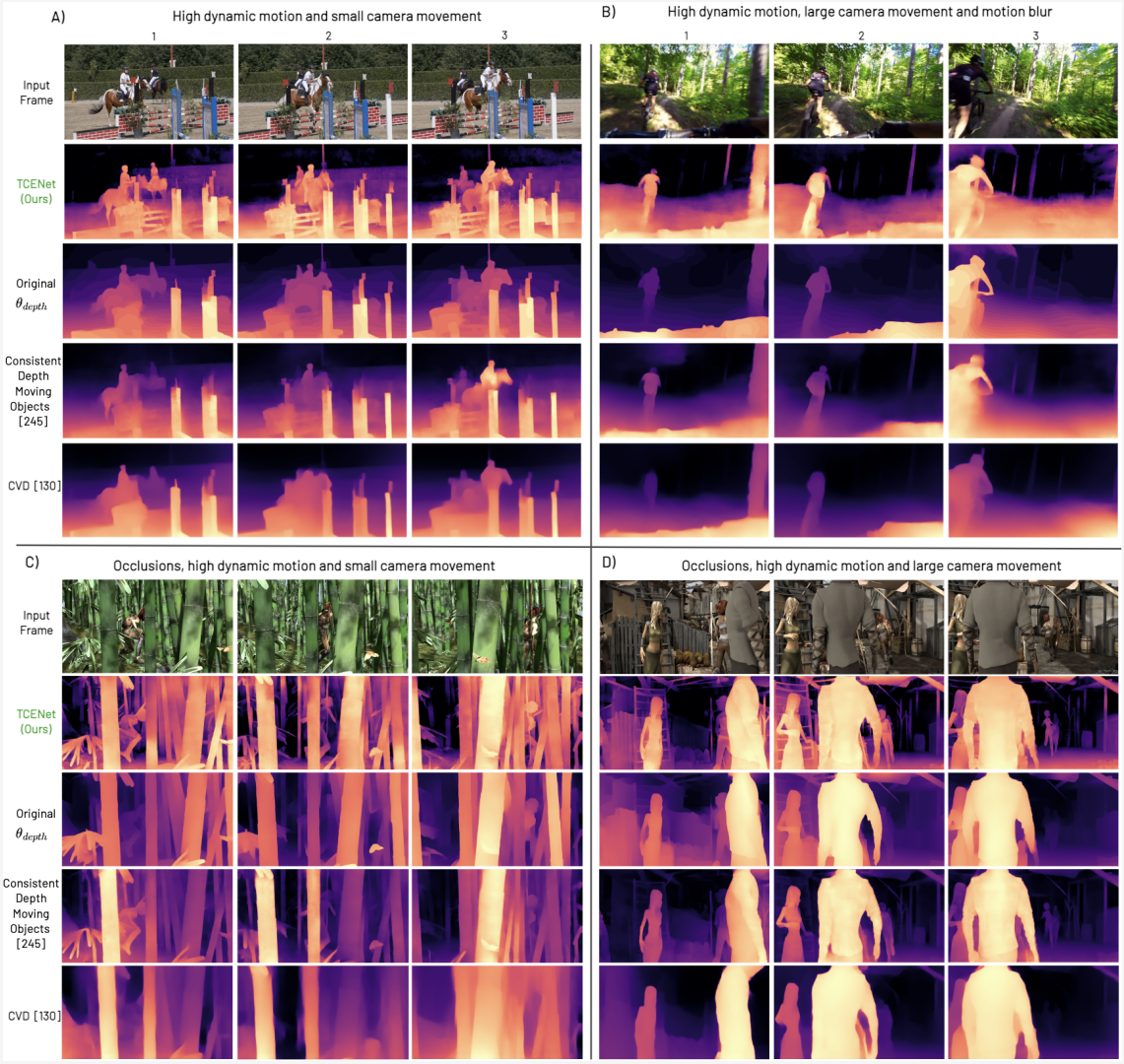


Figure 7.7: Qualitative results comparison on consecutive DAVIS (A, B) and MPI-Sintel (C, D) test frames, emphasizing dynamic motion, occlusions, and camera movement. In each sample, comparisons are displayed with (1) TCE-Net, (2) single-image depth estimation θ_{depth}, which is MiDASv2 [171], (3) [261] and (4) [136]. Note, TCE-Net is deployed without test-time training (frozen), using the pipeline from Fig. 7.2. Whereas [261] and [136] undergo test-time training.

maps. We attribute the enhancement in the spatially consistent of the depth maps and the high-resolution depth estimates for intricate edges and objects to the pre-training of TCENet with the auxiliary loss L_{aux}^{depth} and more specifically, with the L_{depth} component in Eq.7.15. This part of the loss function trains TCENet to estimate depth for both close range and distant objects across multiple scales. In conjunction with the diffusion loss component L_{diff}, TCENet enhances the input depth maps through iterative refinement, achieving high spatial accuracy. Moreover, modifying the input depth maps by utilising an improved θ_{depth}, further elevates the spatial fidelity of the resulting outputs. As demonstrated in the last row of Fig. 7.8, depth maps derived from $\theta_{depth} = LeRes$ provide a more detailed and sharper depth estimation for distant objects. This can be seen specif-

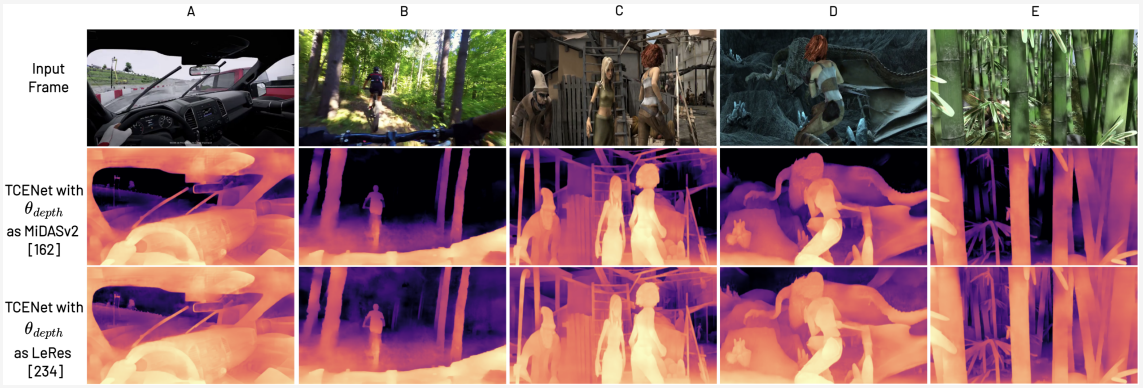


Figure 7.8: Displaying the difference in quality of depth maps produced by TCE Net pre-trained with different θ_{depth} models. Where $A - E$ columns are different samples from the DAVIS and MPI-Sintel test data. first row is outputs of θ_{depth} as MiDASv2 [171] and second row is, LeRes [250].

ically in samples A and C where details of the scene from the car windshield and the poles / infrastructure holding the tent, respectively are clearly visible.

Temporal Consistency Evaluation: Across all video samples featured in Fig. 7.7, it can be observed that for each sequence, TCENet consistently generates depth maps with values within the same range throughout the sequence. In contrast, other methods exhibit flicker artifacts, where the colour map of the depth estimates experiences significant shade variations despite being part of the same sequence. For example, for sample C and the CVD method [136] (last row), frames 1-2 display the depth of the background bamboo sticks as purple (distant from the camera) but abruptly changes to yellow in frame 3 (closer to the camera), even though some bamboo sticks remain far away. Moreover, when faced with occlusions in samples C and D, TCENet delivers sharper and more temporally consistent depth maps compared to other techniques. For instance, in sample C, even though the foreground bamboo sticks obstruct the view of the girl running behind them, TCENet accurately determines the depth for both the girl and the bamboo sticks throughout all frames. In contrast, other models struggle to detect the scene-flow motion of the girl hidden behind the occlusions. As a result, [136] fails to generate on the depth map the girl figure across the frames. Particularly in frame 3, where it creates a hole for her. The model in [261] incorrectly estimates the depth for the occlusions (bamboo sticks) in the foreground in front of the girl, leading to flicker. In frame 2, θ_{depth} misses the girl completely in its generated depth map. In sample D, both the camera (panning to the right) and the object (girl running to the right) are in motion, while being obscured by the person standing in front of the camera. TCE Net accurately tracks the motion of the running girl and the camera, having learned the deep embeddings of the 3D scene flow through the FlowMLP warping. Consequently, it generates consistent depth maps for the running girl despite the occlusions and moving camera. However, other methods exhibit missing components in their depth maps. For example, [261] lacks the girl's upper body in frame 2, and [136] produces a blob, and completely omits the girl in frame 3. We believe TCENet's superior performance is attributed

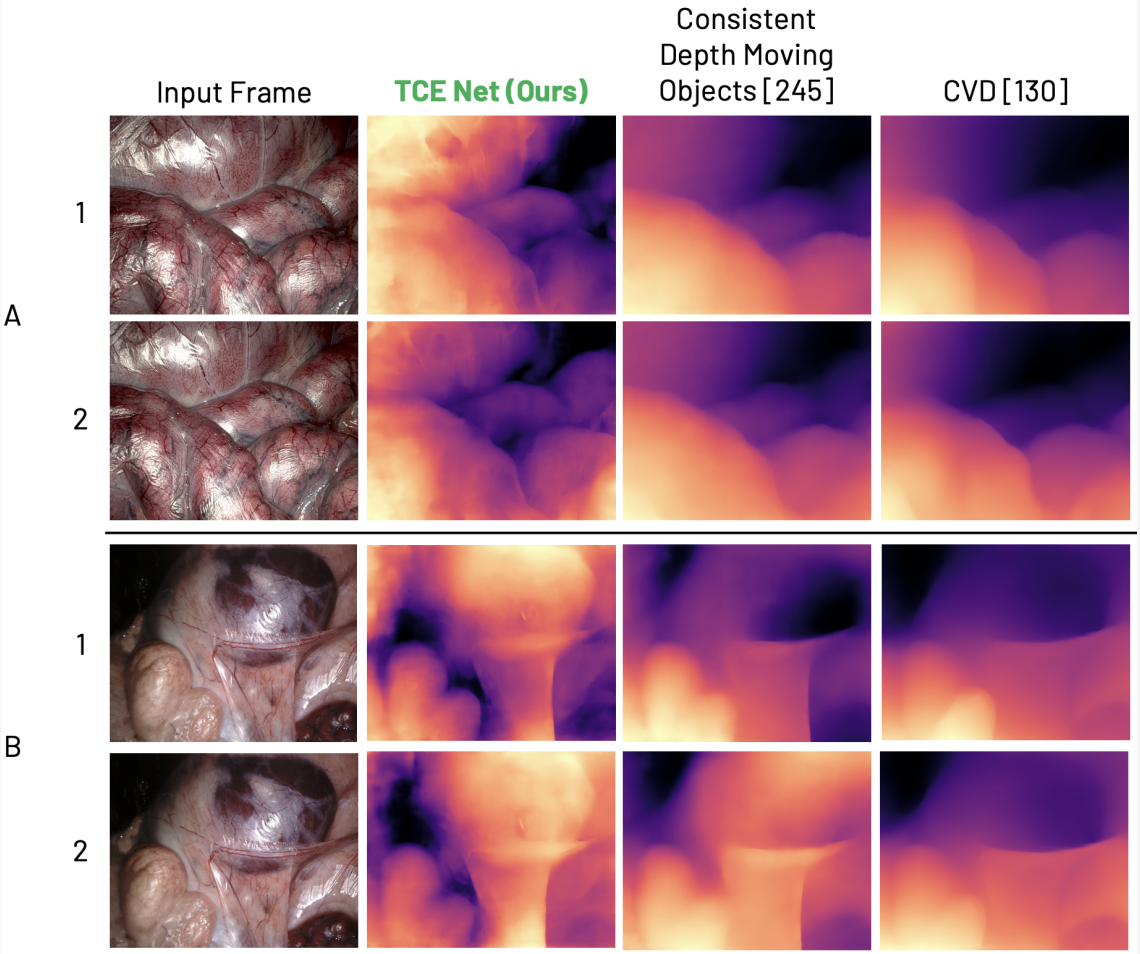


Figure 7.9: Qualitative results comparison on consecutive SCARED (A & B) test frames. In each sample, comparisons are displayed with (1) TCE-Net, (2) [261] and (4) [136]. Note, TCE-Net is deployed without test-time training (frozen), using the pipeline from Fig. 7.2. Whereas [261] and [136] undergo test-time training.

to the final L_{geo} loss which enforces depth to be consistent across the input video sequence based on the forward-backward consistency constraints. Furthermore, the improved FlowMLP's embeddings and scene-flow warping, significantly improves TCENet's understanding of both depth, 3D scene motion and camera pose throughout the video sequence.

In Figure 7.9, we assess the temporal consistency of the models in the MIS domain, despite not training on such data. As can be seen in sample B, [261] yields inconsistent depth estimates between frames 1 and 2, with the primary anatomical structure displaying an irregular color map. Although [136] generates consistent depth, it sacrifices spatial detail, resulting in blurred depth maps and smoothed structures. In contrast, TCENet creates consistent depth maps across both samples despite not being trained on the SCARED database. The spatial performance of TCENet can be further enhanced by training on SCARED or through test-time training during inference.

7.5.2 Quantitative Evaluation

In addition to the qualitative assessment, Tables 7.1 and 7.2 present the quantitative evaluation of the temporal consistency of the compared depth estimation methods based on the M_{pe}^{TC} metrics, for the DAVIS and MPI-Sintel test video sequences, respectively. It is evident in both instances that TCENet significantly outperforms the other methods across both datasets. This improvement can be attributed to the unambiguous flow generated by FlowMLP, which is employed for image warping in Eq. 7.3. The superior performance of FlowMLP can be attributed to its ability to learn the scene-flow as a probabilistic distribution throughout the entire sequence during volumetric training. This offers a more effective image warping signal compared to the use of 2D optical flow maps from θ_{flow} as implemented in [115, 136]. Moreover, TCENet not only learns geometric constraints through image warping, but also gains an understanding of the temporal dynamics of a scene via diffusion, making it resilient to dynamic scene changes. These combined factors significantly enhance the optimisation of L_{geo}, ensuring that TCENet acquires the necessary feature embeddings to detect object and camera movements in a scene and adjust depth accordingly. This observation is further supported by Table 7.2. Moreover, the frame sampling strategy during training (illustrated in Fig. 7.3) which randomly selects close and distant frames, compels TCENet to understand long-term and short-term scene-flow patterns, ultimately strengthening its ability to predict consistent depth outputs, despite not having access to future video frames. Hence, every aspect of the loss function designed and the model components selected, is crucial for maximising temporal and spatial fidelity of the generated depth maps.

Table 7.1: Image reconstruction quality assessment on DAVIS using M_{pe}^{TC}. Red header columns favor lower values, green headers prefer higher. Bold indicates the best model. Methods marked with † utilize LeRes for θ_{depth}, while others use default MiDASv2.

Method	L1	PSNR	SSIM	IS
[136]	0.19	0.23	0.74	8.21
[261]	0.12	0.24	0.81	9.54
TCE Net	**0.11**	**0.31**	**0.87**	**11.13**
TCE Net †	**0.09**	**0.39**	**0.93**	**14.24**

Table. 7.3 shows the spatial depth evaluation on Kitti. This benchmark uses metrics from [58], allowing comparisons to prior methods. The evaluated metrics comprise Root Mean Square Error (RMSE), Absolute Relative Error (Abs Rel), Square Relative Error (Sq Rel), Root Mean Squared Logarithmic Error (RMSE Log), and accuracy (%).

In Table. 7.3, it can be observed, even though TCENet solely processes prior frames relative to the current one, it attains superior performance in many metrics, compared to methods that incorporate future frames. As shown in Fig. 7.7, TCENet produces depth maps with considerably enhanced

Table 7.2: Image reconstruction quality evaluation on MPI-Sintel via M_{pe}^{TC}. Red header columns favor lower values, green headers prefer higher. The best result is highlighted in bold. Methods marked with † utilize LeRes for θ_{depth}, while others use default MiDASv2.

Method	L1	PSNR	SSIM	IS
[136]	0.29	0.31	0.71	7.91
[261]	0.24	0.40	0.77	8.34
TCE Net	**0.13**	**0.49**	**0.83**	**12.43**
TCE Net †	**0.10**	**0.52**	**0.88**	**16.39**

Table 7.3: Spatial performance assessment on Kitti Raw Dataset (depth capped to 80m) using [57] crop. Red header columns prefer lower values, green headers higher. Bold indicates the best model. In the 2^{nd} column, the time step is relative to current frame at time step 0 hence, $-ve$ and $+ve$ values signify preceding and future frames respectively.

Method	No. of Input Frames (time step)	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Ours (TCE Net)	3 (-2, -1, 0)	**0.092**	0.913	**4.442**	**0.170**	**0.909**	**0.968**	0.980
DeepV2D [211]	5 (random sampling)	0.118	0.908	4.978	0.210	0.855	0.948	0.976
ManyDepth [234]	2 (-1, 0)	0.098	0.770	4.459	0.176	0.900	0.965	0.983
GLNet [31]	3 (-1, 0, +1)	0.099	0.796	4.743	0.186	0.884	0.955	0.979
Struct2Depth [22]	3 (-1, 0, +1)	0.109	**0.825**	4.750	0.187	0.874	0.958	**0.983**
CVD [136]	1 (0)	0.130	2.086	4.876	0.205	0.878	0.946	0.970
Wang et. al [222]	2 (-1, 0)	0.106	0.799	4.662	0.187	0.889	0.961	0.982

detail and high spatial accuracy. This is reflected quantitatively in the improved performance on the Kitti dataset. The above performance evaluation has verified that our method surpasses SOTA methods in achieving both higher spatial accuracy and temporal consistency, thanks to the innovative loss function and the comprehensive pipeline that optimises TCENet.

7.6 Conclusions

In this paper, we introduced a self-supervised learning algorithm designed to maintain geometric and temporal consistency in depth estimation, leading to enhanced 3D SfM approaches. Our approach jointly optimizes the learning of scene temporal dynamics through diffusion, incorporating camera pose and 3D scene-flow. This allows our method to overcome challenges posed by dynamic scene changes and occlusions. We pioneered the use of video diffusion for implicit temporal scene dynamic learning, producing temporally consistent depth maps. To maintain geometric consistency, we integrated a loss function enforcing multi-view geometric constraints with the implicit learning approach. We developed the innovative Temporal Consistency Enforcement Network (TCENet), a 3D-Unet-based depth refinement model with a spatio-temporal attention mechanism. Additionally, we devised a diffusion loss function to train TCENet, allowing it to create depth maps with various geometric details. Finally, we introduced a method for dense 3D scene flow estimation using a temporal sequence of volumes and trained a MLP, FlowMLP, to predict scene flow over time. Each of these components were meticulously designed to collaborate in achieving consistent depth maps even amidst dynamic motion and occlusions.

Our method does not necessitate pre-processing (i.e. use of COLMAP for obtaining accurate camera pose), known camera parameters, or processing of future frames in a video sequence. This renders our method suitable for online real-world applications. It sets a new benchmark in both temporal consistency and spatial accuracy for depth map estimation. This research holds the potential to significantly influence various applications dependent on precise 3D SfM, ultimately augmenting their performance and capabilities.

Current limitations of this work include the lengthy training time, which demands significant computational resources, and specific image resolution at which it functions. As demonstrated by other studies [147, 163], training depth estimation models at higher image resolutions leads to improved results. Consequently, our future endeavors will focus on efficiently scaling our pipeline to accommodate higher resolutions, enhancing the diffusion pipeline to decrease computational complexity, and evaluating their effects on temporal consistency.

7.6.1 Learnings & Future Work

This chapter illustrates how achieving temporal consistency in depth can be accomplished by merging multi-view geometric constraints with generative modelling for video representation learning. Where a supplementary model, such as TCE Net, can be trained in collaboration to grasp the temporal dynamics of a video while conforming to the video's 3D geometry. Furthermore, I emphasised that the crucial element for enhancing temporal consistency resides in accurate estimation of scene-flow, which plays a vital role in re-projecting depth from one reference frame to another. By accurately modeling scene-flow, ambiguities in re-projection are reduced, resulting in a more

powerful learning signal for achieving consistent depth. Therefore, the primary factors to enhance in the future involve refining scene-flow estimation and improving the generative model (in this instance, diffusion) as the foundation for obtaining stable temporal representations. Additionally, I also recommend adding further constraints by incorporating stereo frames and using the model from Chapter 6. Further details are shown below:

1. Combine this process with the stereo model proposed in Chapter 6: The current method was designed based on a monocular depth estimation model to simplify experimentation. This foundational approach can now be expanded to accommodate more complex video inputs, such as stereo, where dual multi-view constraints can be employed to achieve both temporal consistency (warping over time) and stereo consistency (warping between corresponding stereo frames). As a result, the model from Chapter 6 can replace the baseline model θ_{depth}, and the TCE Net encoder architecture can be updated to accommodate stereo inputs. By doing so, the method can be extended to stereo models, enhancing consistency across various depth estimation models of preference.

2. Improving scene-flow representations: In our method, the scene-flow estimation model, FlowMLP, is inspired by NeRF [126]. This concept allows us to calculate dense 3D scene-flow fields by treating videos as volumetric inputs. However, a significant limitation is that it demands numerous video frames to be processed simultaneously (not in batches) to comprehend the entire volumetric structure. This constraint necessitates on substantial computational resources for training and may pose scaling challenges for longer videos. A possible improvement, similar to [209], would involve training FlowMLP in batches of video frames (i.e., volumes) and then combining the scene-flow representations from these volume batches into a single, coherent volume. This distributed computation will make scaling efficient to long videos and by enforcing forward-backward consistency checks between the volume batches will also improve scene-flow estimation.

3. Improve temporal generative modelling: The generative modelling method used in this work was diffusion for training TCE Net. This powerful approach enabled TCE Net to learn video temporal dynamics and generalise depth temporal consistency to any video without needing test-time training. However this approach of learning temporal dynamics distribution can be further improved by utilising the latest innovations from generative modelling literature. Such as conducting the denoisining reconstruction process in the latent space [180] or by upgrading the architecture to have scene-flow embeddings as input (as opposed to using them simply to compute the loss) [48].

Chapter 8

Deployment of Deep Learning Models on HoloLens

This chapter introduces a Machine Learning Operations (MLOps) pipeline for deploying deep learning models on the cloud, for use with Mixed Reality (MR) headsets like HoloLens. The challenge of deploying Deep Learning (DL) models post-training is often overlooked in research, despite the fact that GPU-dependent models risk remaining limited to research settings due to the absence of such hardware in mobile devices. While previous chapters addressed the four primary barriers to clinical adoption of MR through innovative DL solutions, this chapter demonstrates the use of cloud computing as a potent approach for deploying DL models on mobile devices via scalable, on-demand resources. This chapter aims to bridge the gap between development and real-world deployment of these models. Offloading these complex calculations to the cloud allows mobile devices to utilise DL models without hardware limitations and enables continuous maintenance with access to updated models, bypassing hardware alterations.

In this work, I harness the power of cloud computing to develop a software architecture capable of deploying DL models for use on any mobile device, including MR. To accommodate the variability of real-world data, I introduce a data version control system that logs newly collected data and indexes it for future model training. Moreover, I employ model version control to track alterations in deployed models, such as variations in weights or model architectures over time. Finally, to guarantee that the work presented is reproducible and deployable on any cloud Virtual Machine (VM), I encapsulate all software within a Docker image, which serves as a virtual Operating System (OS). This Docker image can be easily downloaded on any VM, allowing the proposed software to operate seamlessly, much like a cloud-based app. This MLOps architecture establishes a solid foundation for deploying any models discussed in this thesis or beyond, facilitating their use on mobile devices in both clinical and non-clinical settings and ultimately driving their clinical adoption.

The work in this Chapter was carried out in collaboration with Mr. David Shapira, who assisted in the HoloLens 2 software development.

8.1 Overview

This chapter presents a Machine Learning Operations (MLOps) pipeline that leverages cloud computing to deploy deep learning models on Mixed Reality (MR) headsets. The proposed software architecture addresses the challenge of model deployment after training and development, overcoming the limitations of mobile devices' hardware capacity. The solution incorporates data version control and model version control to accommodate real-world scenarios and track changes in both data and model. The architecture is reproducible and deployable on any cloud Virtual Machine (VM) using a Docker image. For the purpose of this thesis the architecture was deployed on Microsoft Azure cloud and was tested on the HoloLens 2 MR headset. The pipeline operates with the user first launching a HoloLens app of choice where the cameras mounted on the device capture all actions. This video is streamed to the cloud VM where the deployed model processes it, and performs inference on it. The outputs are then streamed back onto the HoloLens for the user to visualise. The current focus was on testing the functionality of the MLOps pipeline through visualization. In the future, the outputs would be used to perform background analysis for enhancing the user's visual understanding of the scene, as opposed to simple direct visualisation. The aim of this study is to introduce a working MLOps architecture that takes developed models and deploys them in the real-world. This can enable testing the clinical impact of such models in the future. This MLOps approach, lays the groundwork for such future model deployment and visual scene understanding on mobile devices in various settings and fosters the clinical adoption of deep learning advancements on MR headsets.

Video result HoloLens test (My room): 32 seconds video

8.2 Introduction

In recent years, the rapid development of Deep Learning (DL) models has led to significant advancements across various domains, from computer vision and natural language processing to healthcare and finance. These models have achieved remarkable results, demonstrating their potential to revolutionise multiple industries. However, despite their impressive performance, the deployment and maintenance of DL models in real-world scenarios remain challenging [176, 190]. MLOps [116] has emerged as a critical discipline that addresses these challenges by streamlining the process of deploying, managing, and maintaining machine learning models throughout their lifecycle. MLOps is a set of practices that combines machine learning, data engineering, and Software Development and Operations (DevOps) principles to enable efficient and reliable production and deployment of machine learning models. By automating and orchestrating the end-to-end machine learning pipeline, MLOps helps ensuring that Machine Learning (ML) / DL models are consistently updated, maintained, and monitored, thereby driving more value from them and the monetary investments in creating them.

Despite its importance, MLOps is often overlooked, as the emphasis in research, tends to be on improving model performance rather than on effective deployment and operationalisation. This oversight can lead to difficulties in transitioning from research to real-world applications, hindering the realisation of Artificial Intelligence (AI) 's full potential. In this chapter, I describe the significance of MLOps in the deployment and maintenance of deep learning models, highlighting the key components that need to be taken into consideration to ensure their successful implementation. These components include data management, model hosting, deployment, monitoring, and communication among cross-functional devices (i.e. MR headset and cloud compute VM). I also delve into the challenges and limitations of the current MLOps landscape, such as scalability, reproducibility, and the complexities of integrating with existing infrastructure and workflows.

In addition, I propose a MLOps architecture specifically designed for this research, which was evaluated using the Microsoft HoloLens 2 by deploying two models simultaneously namely, (1) the TCENet, a temporally consistent monocular depth estimation model from Chapter 7, and (2) the optical flow estimation model, RAFT [212]. These two models were chosen because they provide dense outputs with information for every pixel of the input image, and their high computational complexity makes them an ideal test bed for the proposed MLOps architecture. The proposed MLOps architecture includes data version control and model version control systems to effectively handle real-world applications and keep track of modifications in both data and models. By implementing these version control mechanisms, the architecture allows for seamless adaptation to evolving circumstances while maintaining a clear history of changes. To ensure reproducibility and ease of deployment across various cloud VMs, the architecture utilises a Docker image, which encapsulates all the necessary software components including the third-party libraries needed to run

the code. This approach enables the MLOps solution to be quickly deployed on any cloud VM while maintaining consistency across different environments, significantly simplifying the deployment process.

By comprehensively exploring the MLOps landscape, this chapter aims to raise awareness of its importance and promote its adoption across the research world seeking to harness the full potential of DL models. Research must cater for real world use of such innovative algorithms and thus this work facilitates the delivery of such models to the real world. Additionally, I outline a roadmap for future research and development in MLOps to advance the field further. Through this analysis, I aspire to bridge the gap between the development of cutting-edge deep learning models and their successful deployment and maintenance in real-world applications, ultimately driving more value from AI and Computer Vision (CV) applied in surgery.

8.2.1 Contributions

The main contributions of this chapter are outlined below:

1. To the best of my knowledge, I introduce the first MLOps pipeline enabling seamless integration of multiple DL models in mobile devices like MR, thus expanding their potential use cases and applicability in various fields.

2. Established a reproducible and scalable MLOps pipeline, with version control for both the data and model. Thereby, accommodating for changes occuring in the real world scenario.

3. Designed a cloud-based approach to overcome hardware limitations in MR headsets, making complex DL models accessible on these devices without compromising performance, and enhancing user experience.

4. Laid the groundwork for future research and development in MLOps for MR, opening up new possibilities for innovation in this field and encouraging further exploration of its potential clinical impact.

8.3 Background

8.3.1 MLOps Overview

MLOps is an interdisciplinary field that combines the principles of DevOps, data engineering, and ML. It aims to standardise, automate, and streamline the process of deploying, monitoring, and maintaining ML models in production environments. The need for an MLOps framework arose from the increasing complexity of ML systems and the challenges they presented in deployment and maintenance. Specifically with the advent of DL, models became more sophisticated, and larger in size (large number of parameters) thereby exponentially increasing their computational complexity. The latest models in the literature (at the time of this thesis) scale up to 1 trillion neurons [49]; surpassing the computational complexity of the human brain by a factor of 11. Hence, industry organisations began to realize that the deployment and management of ML models presented unique challenges compared to traditional software development.

The work in [190] highlighted these challenges, emphasizing the importance of considering the hidden technical debt that could accumulate in ML systems if these issues were not addressed. This paper laid the groundwork for the development of MLOps as a distinct discipline dedicated to addressing these issues. The key practices that govern MLOps development are the following:

1. Model Versioning and Management: Establishing version control mechanisms for ML models, enabling seamless tracking, comparison, and rollback of different model versions throughout their life cycle. The same can be done for data to track exactly which samples model performance is good and bad such that, in the future models can be re-trained on bad samples to avoid such out-of-distribution mistakes.
2. Model Deployment: Streamlining the deployment of ML models into production systems, using containerization, Application Programming Interface (API)s, or server-less architectures to expose model predictions and facilitate integration with other applications.
3. Continuous Integration and Deployment (CI/CD): Implementing CI/CD pipelines to automate the integration of new code, data, and models into production systems while ensuring system stability and performance. Particularly achieved with containserization of code for modularity.
4. Monitoring and Maintenance: Continuously monitoring deployed ML models for performance drifts, data anomalies, and resource utilization, and implementing processes for model retraining and redeployment when needed.
5. Security and Governance: Ensuring that sensitive data is properly managed, stored, and processed throughout the ML life cycle, complying with data protection regulations such as General Data Protection Regulation (GDPR), Health Insurance Portability and Accountability Act

(HIPAA), etc. This includes implementing access controls, encryption, and data anonymization techniques to safeguard sensitive information.

In this chapter we propose a MLOps architecture that address points 1-4. Point 5 was not necessary to consider as no sensitive data was processed during the development of this framework. However, this should be addressed in the future.

8.3.2 Benefits of adopting MLOps

As ML and AI continue to make impressive progress across diverse industries, the demand for a systematic and efficient method to manage the life cycle of a ML model grows increasingly important. Implementing MLOps practices when developing ML models offers several advantages. The following are the most significant ones:

- Improved collaboration: MLOps encourages collaboration between data scientists, ML engineers, and IT operations teams, fostering cross-functional communication and reducing the time it takes to bring models to production.
- Faster deployment: By streamlining the development process and automating repetitive tasks, MLOps enables organizations to deploy ML models more quickly and efficiently.
- Enhanced reproducibility: MLOps promotes the use of version control for both code and data, ensuring that models can be easily reproduced, shared, and reviewed by different team members.
- Greater scalability: MLOps provides a structured framework for scaling ML models across various environments and infrastructures, allowing organizations to grow their ML capabilities with ease.
- Better model monitoring and maintenance: With MLOps, organizations can monitor the performance of their ML models in real-time, identifying and addressing issues before they become critical.
- Continuous improvement: MLOps supports continuous integration and continuous delivery (CI/CD) practices, enabling organizations to iteratively improve their models and incorporate new data, features, and techniques as they become available.
- Better compliance and governance: MLOps provides a framework for managing data privacy, security, and regulatory compliance, making it easier for organizations to meet industry-specific requirements and maintain transparency in their ML operations.

8.3.3 Limitations of MLOps

While MLOps has the potential to significantly improve the deployment, management, and maintenance of machine learning models, there are several current limitations that organizations face

when implementing MLOps practices:

- **Scalability**: Managing and deploying machine learning models at scale can be challenging, especially when dealing with large-scale data processing, model training, and real-time inference. As the number of models and their complexity increases, organizations may struggle to scale their MLOps processes effectively.
- **Reproducibility**: Ensuring the reproducibility of model training and deployment processes is essential for consistent results and trustworthy AI systems. However, factors like varying dependencies, software environments, and hardware configurations can introduce inconsistencies that are difficult to track and control.
- **Integration** with existing workflows: Integrating MLOps practices into existing software development and data engineering workflows can be challenging. Organizations may encounter difficulties in adapting their current processes to accommodate the unique requirements of machine learning models, such as versioning of data, models, and code, as well as the need for continuous monitoring and retraining.
- Model monitoring and **drift detection**: Monitoring the performance of deployed models and detecting data drift is crucial to ensure that models remain accurate and relevant. However, setting up effective monitoring systems and maintaining them can be complex and time-consuming.
- **Security and compliance**: Ensuring the security and compliance of machine learning models and data pipelines is crucial, especially in industries with strict regulatory requirements. Implementing robust data privacy, access control, and model auditing mechanisms can be challenging within MLOps workflows.
- **Tooling and infrastructure**: While several tools and platforms exist for different aspects of the MLOps life cycle, there is a lack of comprehensive, end-to-end solutions that can seamlessly integrate all MLOps components. Additionally, organizations may face challenges in selecting the right tools and infrastructure for their specific needs and adapting them to their existing technology stack.

Despite the numerous benefits of MLOps, its implementation can be challenging due to the complexity of the pipeline. Hence, the limitations are also reflections of the benefits it offers. It requires expertise in multiple areas, including data engineering, software development, and machine learning. Additionally, MLOps requires the integration of various tools and frameworks, such as data version control, model version control, Docker, and cloud computing, which can be daunting for non-experts. Furthermore, the deployment and maintenance of machine learning models in production environments require a deep understanding of the underlying hardware and software architecture. Given these challenges, I present a solution that combines all these expertise into one solution to successfully implement an end-to-end machine learning deployment pipeline.

8.3.4 Tools & Frameworks

There are various tools and frameworks available that support the implementation of the MLOps best practices mentioned above, helping practitioners manage the end-to-end ML life cycle more effectively. Some notable tools and frameworks include:

- **MLflow** [254]: MLflow is an open-source platform that provides tools for managing the complete ML lifecycle, including experiment tracking, project packaging, and model deployment. It is primarily used as model version control.
- **TensorFlow Extended (TFX)** [14]: TFX is an end-to-end platform developed by Google for deploying ML pipelines. It includes components for data validation, pre-processing, model training, evaluation, and deployment, as well as support for continuous integration and deployment. It is a general-purpose library that handles all of ML development and deployment. However, it is mostly used for deployment due to its limited support on the development side of the library. Furthermore it is limited to models trained via Tensorflow (hence, models trained on PyTorch cannot be used in this library).
- **Kubeflow** [157]: Kubeflow is an open-source project dedicated to making deployments of ML workflows on Kubernetes simple, portable, and scalable. Where, Kubernetes is an open-source platform for automating the management of applications on distributed computing. It streamlines the process by distributing traffic, auto-scaling resources, enabling rolling updates and rollbacks. This library works well for MLOps on large GPU clusters distributed across multiple devices and is typically used when scaling production on multiple devices. However, it's use on a single device deployment is unnecessary and adds complexity.
- **DvC** (Data Version Control) [117]: DvC is an open-source version control system for managing and logging data in ML projects. It provides tools for data versioning, pipeline management, and experiment tracking and aids in ensuring reproducibility across models.
- **Weights & Biases** [15]: Weights & Biases is a platform that offers experiment tracking, plotting, and evaluation tools. It helps tracking DL models training process and evaluation to streamline tracking of experiments and performance in a centralised dashboard. It is typically used in pre-deployment phase of an ML project. An open source free alternative is TensorBoard [1] which has similar functionality and also works with different DL libraries albeit TensorFlow or PyTorch.

In this research, we specifically employ DVC and MLFlow as our primary version control systems for tracking model performance and data during inference. TFX was unsuitable for this study because all models were developed in PyTorch. As our method was deployed on a single cloud VM instead of distributed cloud clusters, Kubeflow was not necessary. The standard maintenance software provided by Microsoft Azure proved adequate. Finally, TensorBoard and Weights & Biases were explored during R&D, not inference; as the models were not updated at the inference stage.

8.4 Methods

The proposed MLOps software architecture is displayed in Fig. 8.1. In this section I will describe each component and its importance in deploying DL models.

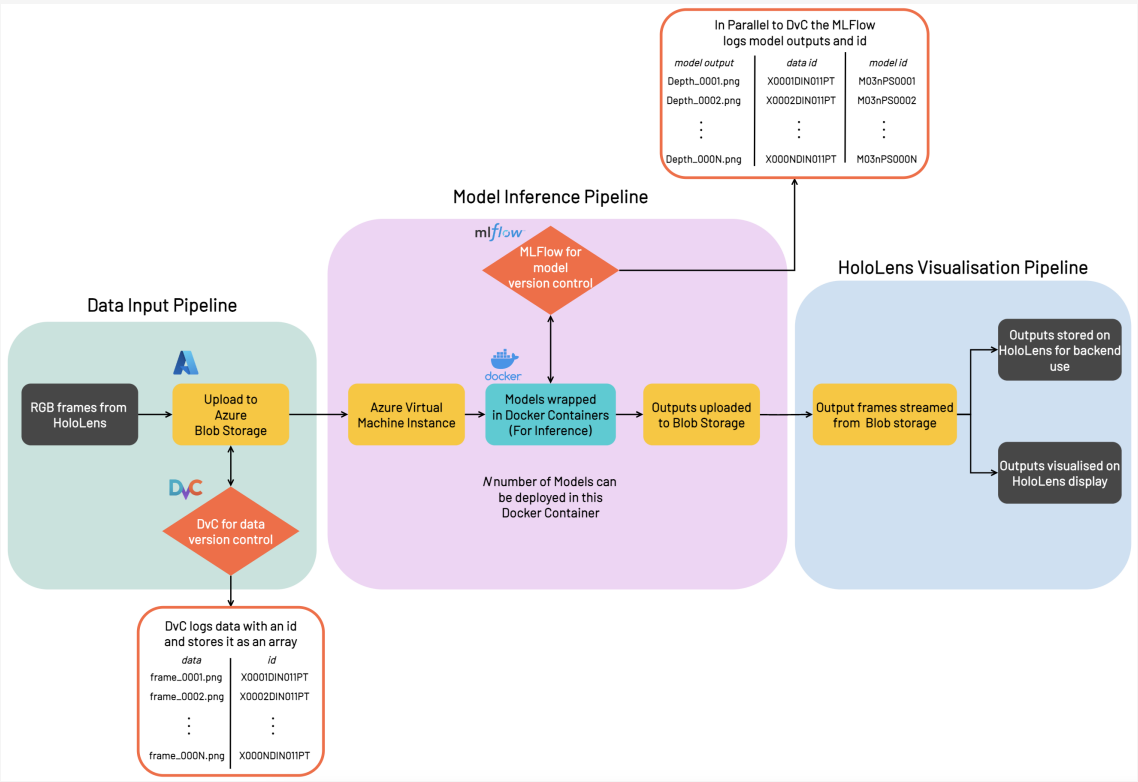


Figure 8.1: The proposed MLOps pipeline for running inference of DL models on images / videos streamed from Microsoft HoloLens 2 mobile device. Black, yellow, blue and orange boxes signify operations related to HoloLens, Microsoft Azure cloud, Docker and version control, respectively.

The architecture above was designed for processing user-recorded videos on a HoloLens 2 headset and processing it via DL models for visualising the outputs. Firstly, the user captures a video with the HoloLens headset, which serves as the input data for subsequent processing. This live video recording is streamed from the HoloLens headset to a designated Azure Blob Storage account frame-by-frame. This cloud-based storage service is particularly designed for handling large volumes of unstructured data. This ensures that the data is securely stored and accessible for further processing. Simultaneously, a VM in Azure is provisioned, which hosts Docker containers with the pre-trained DL models. The Docker containers allow for a consistent and reproducible environment, simplifying the deployment and management of the models. The scripts in this docker container continuously search for the latest frames to process from the Azure blob storage via a communications protocol supported by the azure-blob-storage python library.

The video data from Azure Blob Storage is then fed into the DL models running on the Docker con-

tainers within the VM. The models process the data, performing inference to generate predictions on the input video. The results from the inference process are saved locally (as backup) and simultaneously streamed back to Azure Blob Storage, making them accessible for further analysis downstream or visualization on the HoloLens. The model outputs are thus, retrieved from Azure Blob Storage and visualized on the HoloLens headset, allowing the user to interact with or review the results. Throughout this process, Data Version Control (DvC) and MLflow are used for maintaining version control and tracking the performance of the ML models and change in data respectively. DvC manages data versions and ML pipelines, while MLflow tracks experiment runs, model performance, and metadata. These tools ensure that the ML workflow is reproducible, auditable, and easily manageable. The configuration of the VM used in our study was a cloud cluster comprising of 2 Nvidia Tesla V100 GPUs, 6 core Intel CPU and 112 GBs of RAM. The exact name of the Azure machine is $Standard_NC6s_v3$.

By following this simple light-weight MLOps architecture, anyone can effectively utilise cloud-based resources to process and analyze data from MR headsets, while maintaining version control and ensuring a smooth, efficient workflow. In our experiments, the deep learning models we deploy during inference is the TCENet from Chapter 7 which is a model that outputs temporally consistent monocular depth maps and RAFT [212] which outputs optical flow.

8.4.1 Components

The key building blocks for any MLOps pipeline are: (1) cloud computing, (2) data version control, (3) model version control and (4) docker. The following describes in detail the roles of each of these components and how they come together into a single deployment architecture described above.

Cloud : Utilising cloud computing for MLOps offers numerous benefits, making it an ideal solution for managing the end-to-end ML life cycle. Cloud platforms provide virtually unlimited storage capacity for data, enabling anyone to manage vast volumes of information without the need for extensive on-premises infrastructure. This simplifies data management and access, while also ensuring data redundancy and availability. Additionally, cloud-based VM allow for easy deployment of ML models across diverse environments, ensuring flexibility and rapid scaling. With the ability to provision VMs on-demand, anyone can adjust their compute resources to meet fluctuating needs, optimizing costs and efficiency. Furthermore, cloud providers offer access to powerful GPU resources, which are essential for training and running complex deep learning models. By offloading computationally intensive tasks to the cloud, anyone can leverage the latest GPU technologies without the need for significant upfront investments in hardware.

Data Version Control : Data Version Control (DvC) is an essential tool for overseeing and versioning data sets and machine learning models within MLOps processes. Built upon Git, a widely-used

distributed version control system, DvC extends Git's capabilities to handle large data and model files effectively. By monitoring modifications to data and model artifacts, DvC allows users to maintain a transparent history of their work, making it simple to revert to earlier versions when necessary. Behind the scenes, DvC stores metadata about the data files and models in the Git repository, while the actual files are stored separately, either on local storage or remote storage (such as cloud storage). This method ensures that users benefit from version control without causing the repository to become overwhelmed with large files.

Incorporating DvC in MLOps offers significant advantages, including facilitating collaboration among team members involved in ml projects. DvC enables various team members to work on different versions of data and models simultaneously, while preserving a clear record of alterations. This enhances the reproducibility of experiments and fosters confidence in the results. Furthermore, DvC simplifies the process of sharing data sets and models among teams and across environments, encouraging efficient collaboration and knowledge sharing. By integrating DvC into MLOps workflows, anyone can effectively manage the intricate and iterative nature of ML projects, ensuring harmonious production of reliable and high-quality models.

Model Version Control : Model version control is a process that tracks the changes made to ML or DL models over time, enabling users to understand and manage model evolution. It is a critical component of MLOps that helps manage model versions, dependencies, and configurations, ensuring reproducibility and enabling continuous improvement. Model version control provides several benefits to MLOps, including version tracking, code management, and model deployment. By tracking model versions, users can identify and revert to previous versions easily, ensuring model reproducibility. Model version control also helps manage model dependencies, such as libraries, frameworks, and configurations, ensuring consistent and reliable model performance. Additionally, model version control enables users to manage the deployment of different model versions in various environments, such as testing, staging, and production, ensuring that the correct version of the model is being used in each environment. In the background, model version control uses a combination of version control tools, such as Git, and model packaging formats, such as ONNX, to track model changes and dependencies. Overall, model version control is a vital component of MLOps that helps manage the complex process of developing, testing, and deploying machine learning models, ensuring model reliability and performance.

Docker Docker is a containerization platform that simplifies the process of packaging and deploying software applications, including ML/DL models. Docker provides several benefits for MLOps, including portability, reproducibility, and scalability. It allows users to create a container, which is a self-contained environment that contains all the necessary dependencies, libraries, and configurations for running the application, including DL models. Docker enables users to package and deploy machine learning models easily, ensuring consistent and reliable model performance

across different environments. It also streamlines collaboration, making it easy to share code and models.

In the background, Docker uses a layered file system that enables users to create a lightweight, portable image of the container, which can be easily deployed on any system that supports Docker. Docker images can be versioned and stored in a central repository, making it easy to manage and share containerized applications. Docker also provides tools for managing containers, such as Docker Compose, which enables users to define and manage multi-container applications.

All together a working MLOps pipeline requires these four key components to ensure efficiency, reproducibility, and scalability. Docker provides a consistent and isolated environment for deploying and managing ML models, enhancing portability and reducing environment-related issues. Model version control allows for tracking model changes, performance, and metadata, ensuring easy management and collaboration across teams. Data version control helps manage data changes and ML pipeline configurations, promoting reproducibility and collaboration on data-related tasks. Lastly, cloud computing enables scalable, on-demand resources and streamlined maintenance, making it possible to run complex ML workloads without being constrained by local hardware limitations. Together, these components create a robust, flexible, and efficient MLOps pipeline.

8.5 Results & Discussion

The architecture depicted in Fig.8.1 was assessed using the Microsoft HoloLens 2 for deploying our models. Specifically, the test was carried out as I wore the headset and performed my everyday activities in my living room. The architecture executed depth and optical flow estimation using TCENet and RAFT, respectively. No offline processing or model inference took place, and all results were obtained using the MLOps architecture, with the models running live on Microsoft Azure Cloud. Fig.8.2 presents a sample of processed frames and the corresponding depth and optical flow maps visible on the HoloLens headset. Fig.8.3 showcases the qualitative comparison between the estimated depth maps from our proposed TCENet model (Chapter 7), CVD [136], and Monodepth2 [64]. The video Video Link demonstrates the results of running the three depth estimation models (similar to Fig.8.3) in parallel using the same MLOps architecture. Note that all the results were generated on the HoloLens live, no offline processing was conducted.

From the video, one can observe there is a delay in the depth estimates being visualised and the RGB video frames captured from the HoloLens headset. The proposed MLOps architecture introduces this delay due to the following key factors:

1. **Data transfer latency**: Uploading the recorded video to Azure Blob Storage and retrieving the processed results requires data transfer between the HoloLens headset and the cloud storage. This results in data transfer latency; where the network latency, bandwidth limitations, and large data volumes can contribute to delays in data transfer. The frames recorded by HoloLens 2 are of 2272×1278 resolution, where each frame is composed of 3.3 Mbs of data. To stream this live to cloud storage and to run inference of a model on a remote VM requires high-speed network connection.

2. **Model inference time**: Every DL model has its own unique run time (i.e. the time it takes to do one forward pass). The run time on a cloud-based VM for inference takes longer than run time performed on a local machine. Furthermore, depending on the complexity of the models and the data being processed. Resource constraints on the VM can also impact the inference speed.

3. **Communication overhead**: This is caused by the communication between different components in the architecture. In our case, this would consist of the HoloLens headset, Azure Blob Storage, and the VM, where the network connections (1) across all 3 devices can introduce latency due to time taken to establish connections. Furthermore, the software's own runtime speed can slow down this communication protocol between devices.

From Fig. 8.2 it is evident that all models are performing as expected despite being hosted live, remotely on a cloud VM. Furthermore, it also displays the prowess of TCENet which produces temporally consistent depth across the different frames despite never being trained on data captured

on the HoloLens 2 headset. This displays TCENet's generalisability in both improving depth over the original monocular depth estimation model (MiDASv2 [172]) and on temporal consistency. We further evaluate this generalisability in comparison with other state of the art models like CVD [136] in Fig. 8.3. One can observe that TCENet achieves temporal consistency in depth outputs across the frame sequence, as indicated by the stable color map and reduced flicker. Additionally, TCENet generates sharper depth maps around object contours, such as hands in frames 2-3 and the chair's backrest in frame 8. In contrast, the other models struggle to produce depth in the correct range, maintain consistency in the output color maps (i.e., depth values), or achieve spatial geometric accuracy, resulting in blurry depth around object edges.

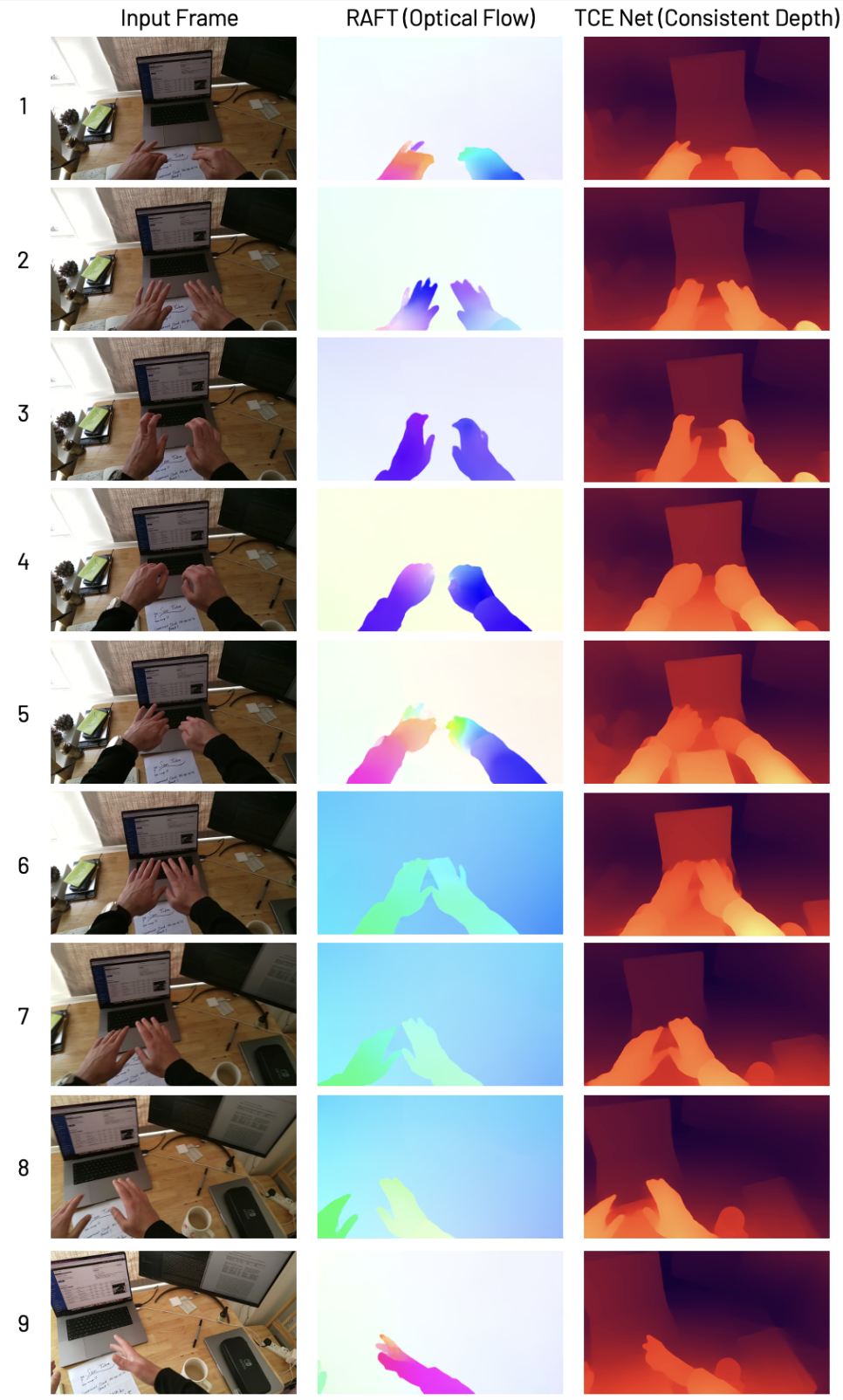


Figure 8.2: The estimated optical flow and depth maps for the corresponding input frames streamed from the HoloLens2 via the proposed MLOps pipeline. Optical flow generated from RAFT [212] and depth generated from TCENet running in parallel on Azure VM in parallel. These output estimates were streamed back to HoloLens 2 device where a screenshot was captured.

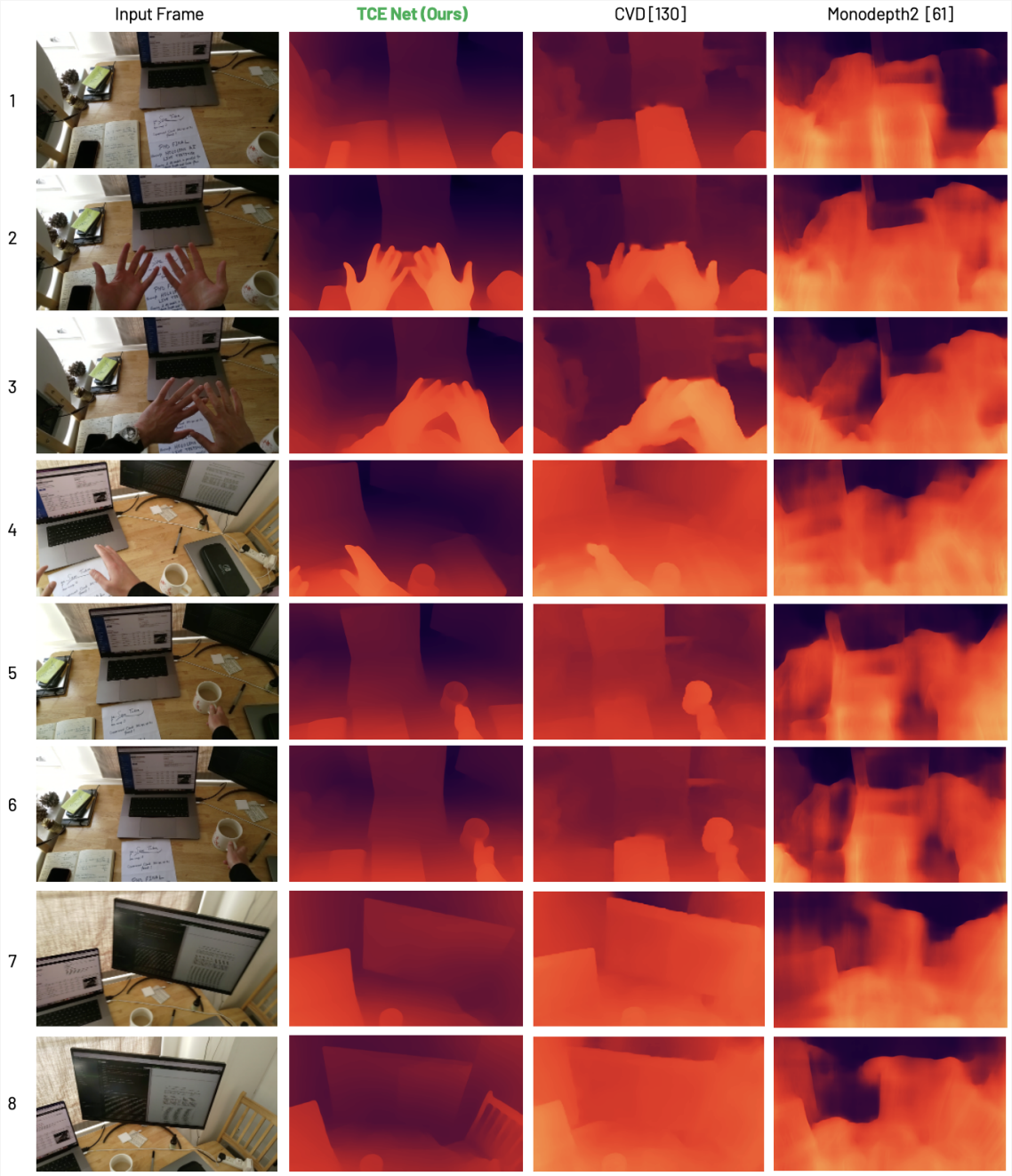


Figure 8.3: The estimated depth maps from 3 methods running concurrently on Azure VM on the corresponding input frames streamed from the HoloLens2 via the proposed MLOps pipeline. The outputs of these models were streamed back to HoloLens 2 device and where a screenshot was captured.

8.6 Conclusion

This chapter presents an MLOps pipeline for deploying deep learning models on the cloud, enabling their use with MR headsets like HoloLens. Addressing the often overlooked challenge of post-training deployment, this pipeline utilises cloud computing to provide scalable, on-demand resources for mobile devices, bridging the gap between development and deployment for real-world applications. By offloading complex computations to the cloud, MR headsets can benefit from deep learning models without hardware limitations and have continuous maintenance with access to updated models. The proposed architecture employs data and model version control, as well as Docker encapsulation, ensuring reproducibility and seamless deployment on any cloud VM. This MLOps architecture lays the foundation for deploying models in both clinical and non-clinical settings, ultimately driving their adoption in various applications.

8.6.1 Learnings & Future Work

In this chapter, I learned about the crucial role that cloud computing plays in MLOps, as it not only offloads computational tasks and enables the deployment of large models but also simplifies scaling MLOps across multiple devices. The variety of tools and frameworks available for MLOps means it's essential for developers to select the ones that best suit their use case to minimize complexity in service design. Additionally, I recognised that security and governance will be a key concern in clinical adoption, which was not addressed in this study. Moving forward, considerable attention must be given to these aspects to ensure safe and compliant deployment in the healthcare domain. Hence, the following are the key areas I recommend further development on for the future:

1. Automated Model Retraining and Continuous Integration: To further improve MLOps, develop a system for automatic model retraining based on changes in data or performance metrics. Continuous integration and deployment pipelines can be implemented to ensure that models are always up-to-date and performant, reducing the need for manual intervention.

2. Enhanced Model Monitoring and Performance Metrics: Establish comprehensive model monitoring solutions that track model performance in real-time, accounting for drifts in data distribution or model degradation. This would enable the identification of issues and trigger model retraining or updates more efficiently, ensuring optimal performance.

3. Robust Security and Governance Measures: Data is the most valuable commodity in the world now. Investing in the development of more robust security and governance mechanisms to protect sensitive data and models must be a priority. This includes access controls, en-

cryption, and data anonymisation techniques, ensuring compliance with regulations and safeguarding the trust of users and stakeholders.

Chapter 9

Conclusions

The primary focus of this thesis is to tackle the key challenges that inhibit wide adoption of Mixed Reality (MR) and robotics in surgery. In particular, the aim is to predict occlusion-aware, geometrically and temporally consistent depth of a scene. This task remains difficult due to scene occlusions, sub-millimeter accuracy requirements and dynamic scene motion. Particular emphasis has been given on using Deep Learning (DL) techniques, focusing on self-supervised representation learning and generative modeling, to overcome the scarcity of ground truth data in surgery and create a generalisable solution. In this research, I developed a custom video generative model to remove occlusions and reconstruct scenes. I designed an advanced multi-scale Generative Adversarial Networks (GANs) framework to achieve state-of-the-art (SOTA) results in monocular depth estimation. This led to a novel stereo depth estimation model that achieves sub-millimeter accuracy in Minimally Invasive Surgery (MIS), bringing significant advancements. I also presented a novel generalisable solution that addresses temporal inconsistency in depth maps through the training of multiple complementary DL models. Furthermore, I introduced a new Machine Learning Operations (MLOps) system to deploy these innovative DL models on MR hardware like the HoloLens.

The impact of this work on clinical Artificial Intelligence (AI) and Computer Vision (CV) development is substantial, as it sets a strong foundation for future innovations in the field. As the field of AI is increasingly embracing self-supervised learning to train DL models with generalisable features, it is an opportune moment to leverage this research work for driving continued progress in the domain. In this chapter, the main conclusions are presented. Generative modeling effectively learns generalisable features that represent underlying data distributions and allows for downstream model refinement. However, supervised learning is essential for dense computer vision tasks like depth estimation due to its sharp, unambiguous learning signal. Therefore, in situations with limited data, such as in MIS, the ideal strategy involves pre-training with generative modeling followed by supervised learning on a smaller dataset for generalisability.

This chapter summarizes the key findings of this research, along with its limitations, and initiates a conversation on potential directions for future research.

9.1 Main Conclusions

Each chapter presented in this thesis address the core challenges in depth estimation in a unique novel way. A common theme that is consistent in the successes of these approaches is generative modelling and representation learning. Recent advancements in Artificial Intelligence (AI) also now utilise these techniques, which have become part of mainstream model development [12, 73, 112]. In this thesis, each content chapter (3-8) presents unique contributions that informs my closing remarks. The following describes the main findings:

Chapter 3: See-Through Vision with Unsupervised Scene Occlusion Reconstruction in Videos

In this chapter, I presented a novel generative modelling approach for reconstructing scenes with occlusions. The core concept that aided in achieving this (amongst other innovations) was a tailor-made loss function, that incorporated adversarial, reconstruction, perceptual, style, and temporal losses, to facilitate unsupervised training with different occlusion-to-image ratios. The resulting model demonstrated invariance to both occlusion and scene types, effectively reconstructing obstructed areas in surgical and natural settings. This marked the first instance of an online video-based method for eliminating occlusions in surgical videos.

This study highlighted the significance of integrating perceptual and adversarial learning to train a model that effectively captures the data's underlying features. This laid the groundwork for enhancing self-supervised learning in later chapters. It became evident that the quality of self-supervised learning relies on crafting a suitable loss function that forces the model to learn deep features hidden in the data, which wouldn't be possible to quantify via supervised losses. Crafting such a loss in this chapter enabled creation of a model that not only learnt temporal dynamics of the video but also reconstructed missing components (occlusions) with temporal and spatial coherence.

Chapter 4: Randomly connected neural networks for self-supervised monocular depth estimation

In many cases like laparoscopic surgeries with monocular views, which currently lack effective computer vision systems. The absence of stereo vision can impede depth perception and cause surgeon fatigue, prompting skilled surgeons to rely on monocular cues for hand-eye coordination. Two key challenges hinder progress in this area: (1) the scarcity of dense ground truth data for surgical settings, limiting deep learning model training, and (2) existing models biased towards natural scenes, which perform poorly in surgical environments even after transfer learning due to distribution drift. To address these challenges, I created a novel neural network architecture for dense monocular disparity estimation, employing random neuron

connections and attention mechanisms for feature extraction. This innovation led to the first randomly connected encoder-decoder structure. Additionally, building on the work of chapter 3, I devised a multi-scale loss function that combines deep feature perceptual and adversarial losses to enhance view synthesis, error calculation, and learning of the input image's underlying 3D distribution.

This research played a crucial role in adapting perceptual and adversarial learning for depth estimation, whilst also enhancing accuracy of monocular depth estimates for surgical applications. Additionally, it also provided me with the valuable insights into potential avenues for extending this work to incorporate stereo inputs, which lead to the innovations in the next chapter.

Chapter 5: Improving Stereo Depth Estimation via Self-Supervised Contrastive Representation Learning

Upon examining other methods in the literature in stereo depth estimation, it became clear that supervised learning-based methods consistently outperform self-supervised ones. Hence, I focused on determining whether self-supervised learning approaches can learn weight distributions similar to those of supervised methods. Thus, building on prior research in self-supervised learning, I devised a stereo depth model training procedure employing joint Contrastive Representation Learning (CRL) and self-supervised learning. This work introduced the first CRL methodology for stereo depth estimation, which involved efficiently training encoder models for subsequent pairing with a decoder architecture to generate high-quality disparity maps. The CRL approach enhances performance in models trained with traditional self-supervised image re-projection error and is scalable and optimized for highly similar images like stereo pairs, even on single GPU hardware.

Through the research conducted in this chapter, I found that self-supervised learning is successful in achieving generalisable stereo input representations but faces ambiguity due to photometric reprojection error. In contrast, supervised learning methods offer a more precise learning signal. This raised the question; once generalisable features have been learned, is self-supervised learning truly necessary for downstream stereo depth estimation? This very question formed the basis of my next chapter where I created a method that attained sub-millimeter accuracy on Minimally Invasive Surgery (MIS) scenes.

Chapter 6: Generalisable Stereo Depth Estimation via Masked Image Modelling

Based on the previous chapter's findings self-supervised learning effectively acquires generalisable feature representations for stereo inputs, but its photometric reprojection error loss

function limits sub-millimeter accuracy. Supervised learning, with its unambiguous and precise depth representations, is crucial. My hypothesis was that self-supervised learning is required solely for attaining generalisable feature representations, while supervised learning can optimize depth without impacting generalisability (in the case of stereo).

Hence, I introduced a pioneering technique, combining self-supervised Masked Image Modelling (MIM) with supervised stereo depth estimation. This constitutes the first application of MIM representation learning for stereo depth estimation. A novel approach was proposed to enhance MIM by incorporating a perceptual learning (presented in chapter 3 & 4) for attaining generalisable visual semantic concepts. A versatile model architecture was introduced, enabling the combination of MIM encoder with any off-the-shelf decoder for improved depth estimation. The joint MIM-supervised method boosts performance, yielding a generalisable model with sub-millimeter surgical depth accuracy.

Chapter 7: Boosting Depth with Temporal-Geometric Consistency

A key limitation of depth estimation models is their per-frame depth estimation, resulting in flickering and temporally inconsistent geometry in video sequences. This issue is exacerbated by dynamic motion, deforming structures, occlusions, and more. Addressing this is vital for adopting Mixed Reality (MR) and autonomous robotics in applied surgery. In this chapter, I build on Chapter 3's findings by using generative modeling to train a model that learns video temporal dynamics and Chapter 6's insights to create a new loss function that enforces multi-view geometric constraints, ensuring consistent predicted geometry in a sequence. Focusing on a monocular video setup simplifies model design and allows flexibility in exploring new training strategies.

Thus I proposed an approach that lays the groundwork for achieving consistency in depth estimates in Deep Learning (DL) models and is compatible with off-the-shelf depth estimation models. This enables future expansion to stereo/multi-view input models using the same pipeline. The method consists of a 3D-Unet trained via diffusion for learning video dynamics and a Multi Layer Perceptron (MLP) for estimating 3D scene-flow. The innovative loss function jointly optimizes both for depth video consistency and enforces multi-view geometry constraints via scene-flow for robustness to dynamic motion. This approach learns video dynamics and maintains depth consistency for objects with high dynamic motion despite camera movement and occlusions. It paves the way for future work adapting consistent depth for deformable depth in soft-tissue structures during MIS.

Chapter 8: Deploying Deep Learning Models on HoloLens

The challenge of deploying DL models after training is often underestimated in research, as GPU-dependent models may be restricted to research settings due to the lack of such hardware in mobile devices. While previous chapters tackled the primary obstacles to the clinical adoption of MR through innovative DL solutions, my goal was to bridge the gap between research and real-world implementation. Consequently, I developed a Machine Learning Operations (MLOps) pipeline for deploying DL models on cloud services like Microsoft Azure, leveraging MR headset data such as HoloLens. Specifically, I designed a software architecture with built-in version control to track changes during usage and facilitate deployment on the cloud, thereby addressing significant computational resource requirements.

Overall, self-supervised learning and primarily generative modeling is a powerful approach that effectively learns generalisable features representing the underlying data distributions, enabling downstream model refinement. As observerd from the work in this thesis, it has been the foundation for all proposed innovation. This technique excels in capturing complex patterns and structures present in the data, which can be beneficial for various applications. However, when it comes to dense computer vision tasks such as depth estimation (and others, like optical flow, segmentation etc.) supervised learning is indispensable due to its precise and unambiguous learning signal, which allows for more accurate predictions.

In scenarios with limited data availability, such as in MIS, combining both generative modeling and supervised learning can lead to improved generalisability and performance. The ideal strategy in these cases involves pre-training the model using generative techniques, allowing it to learn essential features and patterns from the available data. Once the model has been pre-trained, it can be fine-tuned using supervised learning on a significantly smaller dataset, focusing on specific tasks and capturing more subtle details. This smaller dataset can also be created from simulation, which as shown in chapter 6 is sufficient to train a model that generalises to real-world scenes. This two-step process leverages the strengths of both generative and supervised learning approaches, resulting in a more robust and generalisable model that can better handle the challenges associated with limited data in dense computer vision tasks. I strongly believe this research path will be the foundation for the next set of innovations in AI in MIS.

Additionally, self-supervised learning offers significant advantages when it comes to detecting data drift or facilitating online adaptation after a model has been deployed. This capability ensures that the model remains updated and relevant in the face of changing data distributions or evolving real-world conditions. By continuously monitoring and refining the model estimates, generative modeling can help maintain its accuracy and effectiveness over time.

9.2 Barriers to Translation

While the innovations presented in this thesis have demonstrated significant potential, there remain substantial barriers to translating these technologies into widespread clinical practice. One of the primary challenges is the clinical validation of deep learning models in real-world settings. While many models perform exceptionally well under controlled conditions, their robustness in live clinical environments—where factors such as patient variability, tissue deformation, real-time inference and surgical complications introduce significant uncertainties which has yet to be fully tested.

Additionally, integrating these technologies into existing clinical workflows poses another significant challenge. Surgeons are often hesitant to adopt new technologies that require extensive retraining or that disrupt established procedures. There is also an ethical concern regarding the degree of autonomy granted to machines in the operating room, particularly with regard to patient safety. Regulatory bodies, such as the FDA and the MHRA, further complicate the adoption of deep learning-based medical devices due to stringent approval processes that can take years to complete .

To mitigate these barriers, future research must focus on conducting extensive clinical trials to validate these models across a variety of surgical scenarios. Close collaboration with clinicians during the design phase will ensure that the resulting technologies are user-friendly and can be seamlessly integrated into clinical practice. Moreover, regulatory frameworks must evolve to accommodate the rapid development of AI and machine learning in healthcare, balancing the need for rigorous validation with the necessity of bringing life-saving innovations to market more quickly.

In addition, the Machine Learning Operations (MLOps) pipeline introduced in Chapter 8 provides a pathway for continuous improvement and clinical validation of these models. By offloading computationally intensive processes to cloud-based infrastructure, this approach allows for real-time updates and retraining as more clinical data becomes available. This not only improves the robustness and accuracy of the models but also ensures that they evolve alongside changing clinical needs, ultimately improving their chances of being successfully adopted into clinical practice.

9.3 Future Work

In the following description I delineate potential future avenues to advance the research presented in this thesis.

Chapter 3: See-Through Vision with Unsupervised Scene Occlusion Reconstruction in Videos

DL evolves rapidly, with the average model lifetime now just five months before new advancements emerge. Since completing this work, other studies have introduced innovative modules for handling high-resolution inputs on single-image inpainting, such as replacing standard convolution layers with Fourier convolution but adapted for sequences [205]. This could improve inpainting fidelity and make the model lightweight, enabling higher resolution outputs. Additionally, since completing this work, more accurate optical-flow models have been introduced (i.e. RAFT [212]), which could significantly improve results by estimating occlusion deformation with greater precision. Furthermore, instead of parsing flow maps as inputs, explicitly enforcing flow constraints as just a loss function may be sufficient. This can lead to a more lightweight and easy-to-deploy model architectures that focus on input resolution. Reducing the computational complexity of the current model by finding alternatives to 3D convolution, such as combining 2D convolution with Long Short-Term Memory (LSTM) or using a full video transformer, could further boost runtime performance.

Chapter 4: Randomly connected neural networks for self-supervised monocular depth estimation

The main focus for improving monocular depth estimation must be on improving depth accuracy and generalisability. This can be achieved by combining several approaches. Firstly, processing inputs and generating outputs (by modifying the decoder architecture) at higher image resolutions (e.g., 720p or 1080p) can boost accuracy [163]. Secondly, addressing scale ambiguity in monocular depth estimation is crucial, it can be achieved by jointly estimating camera intrinsics and depth, leveraging the latent space of the U-Net architecture [68], which enhances invariance to various camera configurations. Thirdly, new methods utilising zero-shot learning and ground truth data from simulation can further bolster generalisability [172] for changing scenes by employing ground truth depth as a sharper, unambiguous learning signal. Integrating these approaches creates a comprehensive solution that can lead to more accurate and versatile depth estimation models.

Chapter 5: Improving Stereo Depth Estimation via Self-Supervised Contrastive Representation Learning

In Chapter 6, it was demonstrated that self-supervised learning serves as an effective pre-training method for acquiring generalisable baseline features, which can then be fine-tuned using supervised learning for enhanced performance. Therefore, a potential improvement for a fully self-supervised pipeline is to use self-supervised learning for online adaptation. Given that photometric re-projection error is self-supervised, it allows for test-time training to continuously improve results in real-world settings. Furthermore, as I explored in chapter 7 and 8, a major challenge when deploying models in real-world settings is data drift. This occurs when the distribution of real-world data significantly deviates from the training data, leading to model estimation failures. Photometric re-projection metric can be employed to both detect data drift and address it through test-time training, as suggested previously.

Chapter 6: Generalisable Stereo Depth Estimation via Masked Image Modelling

In this chapter I showed how combining self-supervised pre-training and downstream supervised learning can obtain sub-millimeter accuracy in stereo depth estimation. In this work, I enhanced MIM by training it with perceptual loss, which improved the model's reconstruction quality and allowed for a deeper understanding of the scene's semantic context. Vector quantization, introduced in [155], discretizes features based on context, enabling better separation between objects and the scene context in the latent space. This has also proven to improve reconstruction quality. Incorporating vector quantization could significantly improve both the MIM pre-training phase and the downstream stereo depth estimation phase. For downstream stereo depth estimation, only simulation data was used in the training process. A straightforward improvement would be to employ more challenging samples, such as domain randomization [213] and adversarial attacks [137]. This approach would make the model more robust to data drift between simulated and real-world scenes, ultimately enhancing its generalisability.

Chapter 7: Boosting Depth with Temporal-Geometric Consistency

The current method was developed using a monocular depth estimation model for easier experimentation. This foundational approach can be extended to stereo videos, incorporating dual multi-view constraints for both temporal consistency (time-based warping) and stereo consistency (warping between corresponding stereo frames). Consequently, the StereoMAE (chapter 6) model can replace the baseline depth model, and the TCENet encoder architecture can be updated for stereo inputs. This extension allows for enhanced consistency across a range of preferred depth estimation models. Furthermore, the scene flow estimation model FlowMLP, treats videos as volumetric inputs, but this requires substantial computational resources. A potential improvement involves training FlowMLP in video frame batches and combining scene-

flow representations into a single volume, enhancing scaling efficiency for long videos and improving scene-flow estimation through forward-backward consistency checks.

Chapter 8: Deploying Deep Learning Models on HoloLens

To enhance an MLOps pipeline, focus on three key aspects: (1) Continuous Integration and Deployment (CI/CD). Implementing a robust CI/CD process ensures that new models, updates, and improvements are seamlessly integrated, tested, and deployed. This minimizes downtime and allows for rapid iteration, enhancing the overall efficiency and performance of the pipeline. (2) Monitoring and Evaluation, which is essential for tracking the performance and accuracy of deployed models. This enables the early identification of issues, such as data drift or model degradation, and allows for prompt remediation to maintain optimal performance. (3) Scalability and flexibility allows MLOps pipeline to accommodate future growth and changing requirements. By leveraging modular components and implementing best practices for resource management, the pipeline can efficiently adapt to new models, data sources, and deployment environments, ensuring long-term sustainability and success.

References

[1] Abadi, M., Barham, P., Chen, J., et al. "Tensorflow: A system for large-scale machine learning". In: *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*. 2016, pp. 265–283.

[2] Abdulla, W. *Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow*. https://github.com/matterport/Mask_RCNN. 2017.

[3] Ackerman, S., Farchi, E., Raz, O., et al. "Detection of data drift and outliers affecting machine learning model performance over time". In: *ArXiv* abs/2012.09258 (2020).

[4] Adamson, C., Kanu, O. O., Mehta, A. I., et al. "Glioblastoma multiforme: a review of where we have been and where we are going". In: *Expert Opinion on Investigational Drugs* 18.8 (2009). PMID: 19555299, pp. 1061–1083. DOI: 10.1517/13543780903052764. eprint: https://doi.org/10.1517/13543780903052764.

[5] Aleotti, F., Tosi, F., Poggi, M., et al. "Generative Adversarial Networks for unsupervised monocular depth prediction". In: *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*. Sept. 2018.

[6] Allan, M., McLeod, A. J., Wang, C. C., et al. "Stereo Correspondence and Reconstruction of Endoscopic Data Challenge". In: *CoRR* abs/2101.01133 (2021). arXiv: 2101.01133.

[7] Allan, M., Shvets, A., Kurmann, T., et al. "2017 Robotic Instrument Segmentation Challenge". In: abs/1902.06426 (2019). arXiv: 1902.06426.

[8] Allan, M., Shvets, A., Kurmann, T., et al. "2018 Robotic Instrument Segmentation Challenge". In: abs/1902.06426 (2019). arXiv: 1902.06426.

[9] Arjovsky, M., Chintala, S., and Bottou, L. *Wasserstein GAN*. cite arxiv:1701.07875. 2017.

[10] Armanious, K., Mecky, Y., Gatidis, S., et al. "Adversarial Inpainting of Medical Image Modalities". In: *ICAASP* abs/1810.06621 (2018). arXiv: 1810.06621.

[11] Arnab, A., Dehghani, M., Heigold, G., et al. "ViViT: A Video Vision Transformer". In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 6816–6826. DOI: 10.1109/ICCV48922.2021.00676.

[12] Bao, H., Dong, L., Piao, S., et al. "BEiT: BERT Pre-Training of Image Transformers". In: *ICLR 2022*. Apr. 2022.

[13] Barnes, C., Shechtman, E., Goldman, D. B., et al. "The Generalized PatchMatch Correspondence Algorithm". In: *European Conference on Computer Vision*. Sept. 2010.

[14] Baylor, D., Breck, E., Cheng, H.-T., et al. "TFX: A TensorFlow-Based Production-Scale Machine Learning Platform". In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '17. Halifax, NS, Canada: ACM, 2017, pp. 1387–1395. ISBN: 978-1-4503-4887-4. DOI: 10.1145/3097983.3098021.

[15] Biewald, L. *Experiment Tracking with Weights and Biases*. Software available from wandb.com. 2020.

[16] Bloesch, M., Czarnowski, J., Clark, R., et al. "CodeSLAM - Learning a Compact, Optimisable Representation for Dense Visual SLAM". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), pp. 2560–2568.

[17] Brown, T., Mann, B., Ryder, N., et al. “Language Models are Few-Shot Learners”. In: ed. by Larochelle, H., Ranzato, M., Hadsell, R., et al. Vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901.

[18] Brown, T. B., Mann, B., Ryder, N., et al. “Language models are few-shot learners”. In: *arXiv preprint arXiv:2005.14165* (2020).

[19] Bucci, M., Maity, A., Janss, A., et al. “Near complete surgical resection predicts a favorable outcome in pediatric patients with nonbrainstem, malignant gliomas: Results from a single center in the magnetic resonance imaging era”. In: *Cancer* 101 (Sept. 2004), pp. 817–24. DOI: 10.1002/cncr.20422.

[20] Butler, D. J., Wulff, J., Stanley, G. B., et al. “A naturalistic open source movie for optical flow evaluation”. In: *European Conf. on Computer Vision (ECCV)*. Ed. by A. Fitzgibbon et al. (Eds.) Part IV, LNCS 7577. Springer-Verlag, Oct. 2012, pp. 611–625.

[21] Caelles, S., Montes, A., Maninis, K.-K., et al. “The 2018 DAVIS Challenge on Video Object Segmentation”. In: *arXiv:1803.00557* (2018).

[22] Casser, V., Pirk, S., Mahjourian, R., et al. “Depth Prediction without the Sensors: Leveraging Structure for Unsupervised Learning from Monocular Videos”. In: AAAI’19/IAAI’19/EAAI’19. Honolulu, Hawaii, USA: AAAI Press, 2019. ISBN: 978-1-57735-809-1. DOI: 10.1609/aaai.v33i01.33018001.

[23] Catto, J. W. F., Khetrapal, P., Ricciardi, F., et al. “Effect of Robot-Assisted Radical Cystectomy With Intracorporeal Urinary Diversion vs Open Radical Cystectomy on 90-Day Morbidity and Mortality Among Patients With Bladder Cancer: A Randomized Clinical Trial”. In: *JAMA* 327.21 (June 2022), pp. 2092–2103. ISSN: 0098-7484. DOI: 10.1001/jama.2022.7393. eprint: https://jamanetwork.com/journals/jama/articlepdf/2792543/jama_catto_2022_oi_220048_1654531391.99388.pdf.

[24] Chang, J.-R. and Chen, Y.-S. “Pyramid Stereo Matching Network”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 5410–5418.

[25] Chang, Y.-L., Liu, Z. Y., Lee, K.-Y., et al. “Learnable Gated Temporal Shift Module for Deep Video Inpainting””. In: *BMVC* (2019).

[26] Chen, D., Liao, J., Yuan, L., et al. “Coherent Online Video Style Transfer”. In: *2017 IEEE International Conference on Computer Vision (ICCV)* (2017), pp. 1114–1123.

[27] Chen, M., Radford, A., Child, R., et al. “Generative Pretraining From Pixels”. In: *ICML*. Ed. by III, H. D. and Singh, A. Vol. 119. Proceedings of Machine Learning Research. PMLR, 13–18 Jul 2020, pp. 1691–1703.

[28] Chen, T., Kornblith, S., Norouzi, M., et al. “A Simple Framework for Contrastive Learning of Visual Representations”. In: *Proceedings of the 37th International Conference on Machine Learning*. Vol. 119. Proceedings of Machine Learning Research. 13–18 Jul 2020, pp. 1597–1607.

[29] Chen, W.-M. and Jhang, S.-H. “Improving Graph Cuts algorithm to transform sequence of stereo image to depth map”. In: *Journal of Systems and Software* 86.1 (2013), pp. 198–210. ISSN: 0164-1212. DOI: https://doi.org/10.1016/j.jss.2012.07.044.

[30] Chen, Y., Schmid, C., and Sminchisescu, C. “Self-Supervised Learning With Geometric Constraints in Monocular Video: Connecting Flow, Depth, and Camera”. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019), pp. 7062–7071.

[31] Chen, Y., Schmid, C., and Sminchisescu, C. “Self-Supervised Learning With Geometric Constraints in Monocular Video: Connecting Flow, Depth, and Camera”. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 7062–7071. DOI: 10.1109/ICCV.2019.00716.

[32] Cheng, X., Zhong, Y., Harandi, M., et al. "Hierarchical Neural Architecture Search for Deep Stereo Matching". In: *Advances in Neural Information Processing Systems* 33 (2020).
[33] Chung, J., Gulcehre, C., Cho, K., et al. *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*. cite arxiv:1412.3555Comment: Presented in NIPS 2014 Deep Learning and Representation Learning Workshop. 2014.
[34] Cleary, K. and Peters, T. M. "Image-guided interventions: technology review and clinical applications". In: *Annual Review of Biomedical Engineering* 12 (Aug. 2010), pp. 119–142. DOI: 10.1146/annurev-bioeng-070909-105249.
[35] Clevert, D., Unterthiner, T., and Hochreiter, S. "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)". In: *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*. Ed. by Bengio, Y. and LeCun, Y. 2016.
[36] Collamati, F., Olmos, R., Albanese, A., et al. "Current use and potential role of radioguided surgery in brain tumours". In: *Clinical and Translational Imaging* 10 (June 2022). DOI: 10.1007/s40336-022-00503-x.
[37] Cordts, M., Omran, M., Ramos, S., et al. "The Cityscapes Dataset for Semantic Urban Scene Understanding". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016.
[38] Cui, B., Islam, M., Bai, L., et al. "Surgical-DINO: Adapter Learning of Foundation Models for Depth Estimation in Endoscopic Surgery". In: *International Journal of Computer Assisted Radiology and Surgery* (2024).
[39] "Deep learning". English (US). In: *Nature Cell Biology* 521.7553 (May 2015). Funding Information: Acknowledgements The authors would like to thank the Natural Sciences and Engineering Research Council of Canada, the Canadian Institute For Advanced Research (CIFAR), the National Science Foundation and Office of Naval Research for support. Y.L. and Y.B. are CIFAR fellows. Publisher Copyright: © 2015 Macmillan Publishers Limited. All rights reserved., pp. 436–444. ISSN: 1465-7392. DOI: 10.1038/nature14539.
[40] Devlin, J., Chang, M., Lee, K., et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. Ed. by Burstein, J., Doran, C., and Solorio, T. Association for Computational Linguistics, 2019, pp. 4171–4186. DOI: 10.18653/v1/n19-1423.
[41] Dhariwal, P. and Nichol, A. "Diffusion Models Beat GANs on Image Synthesis". In: *Advances in Neural Information Processing Systems*. Ed. by Ranzato, M., Beygelzimer, A., Dauphin, Y., et al. Vol. 34. Curran Associates, Inc., 2021, pp. 8780–8794.
[42] Dhariwal, P. and Nichol, A. Q. "Diffusion Models Beat GANs on Image Synthesis". In: *Advances in Neural Information Processing Systems*. Ed. by Beygelzimer, A., Dauphin, Y., Liang, P., et al. 2021.
[43] Dinh, L., Sohl-Dickstein, J., and Bengio, S. "Density estimation using Real NVP". In: OpenReview.net, 2017.
[44] Doersch, C., Gupta, A., and Efros, A. A. "Unsupervised Visual Representation Learning by Context Prediction". In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 1422–1430. DOI: 10.1109/ICCV.2015.167.
[45] Dosovitskiy, A., Springenberg, J. T., Riedmiller, M., et al. "Discriminative Unsupervised Feature Learning with Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by Ghahramani, Z., Welling, M., Cortes, C., et al. Vol. 27. Curran Associates, Inc., 2014.

[46] Eigen, D., Puhrsch, C., and Fergus, R. "Depth Map Prediction from a Single Image using a Multi-Scale Deep Network". In: *Advances in Neural Information Processing Systems 27*. Ed. by Ghahramani, Z., Welling, M., Cortes, C., et al. Curran Associates, Inc., 2014, pp. 2366–2374.

[47] Elgammal, A. M., Liu, B., Elhoseiny, M., et al. "CAN: Creative Adversarial Networks, Generating "Art" by Learning About Styles and Deviating from Style Norms". In: *Proceedings of the Eighth International Conference on Computational Creativity, ICCC 2017, Atlanta, Georgia, USA, June 19-23, 2017*. Ed. by Goel, A. K., Jordanous, A., and Pease, A. Association for Computational Creativity (ACC), 2017, pp. 96–103.

[48] Esser, P., Chiu, J., Atighehchian, P., et al. *Structure and Content-Guided Video Synthesis with Diffusion Models*. 2023. arXiv: 2302.03011 [cs.CV].

[49] Fedus, W., Zoph, B., and Shazeer, N. *Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity*. 2022. arXiv: 2101.03961 [cs.LG].

[50] Feng, Z., Chi, S., Yin, J., et al. "A Variational Approach to Medical Image Inpainting Based on Mumford-Shah Model". In: July 2007, pp. 1–5. ISBN: 1-4244-0885-7. DOI: 10.1109/ICSSSM.2007.4280177.

[51] Fischer, P., Dosovitskiy, A., Ilg, E., et al. "FlowNet: Learning Optical Flow with Convolutional Networks". In: *Computer Vision and Pattern Recognition (CVPR)* abs/1504.06852 (2015). arXiv: 1504.06852.

[52] Forsyth, D. A. and Ponce, J. *Computer Vision - A Modern Approach, Second Edition*. Pitman, 2012, pp. 1–791. ISBN: 978-0-273-76414-4.

[53] Fu, H., Gong, M., Wang, C., et al. "Deep Ordinal Regression Network for Monocular Depth Estimation". In: vol. 2018. June 2018, pp. 2002–2011. DOI: 10.1109/CVPR.2018.00214.

[54] Gao, C., Saraf, A., Huang, J.-B., et al. "Flow-edge Guided Video Completion". In: *Proc. European Conference on Computer Vision (ECCV)*. 2020.

[55] Gao, W. and Tedrake, R. "SurfelWarp: Efficient Non-Volumetric Single View Dynamic Reconstruction". In: *Robotics: Science and System (RSS)*. 2018.

[56] Garg, R., G, V. K. B., and Reid, I. D. "Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue". In: *European Conference on Computer Vision (ECCV)* abs/1603.04992 (2016). arXiv: 1603.04992.

[57] Garg, R., Kumar, B. V., Carneiro, G., et al. "Unsupervised CNN for single view depth estimation: Geometry to the rescue". In: *European Conference on Computer Vision*. Springer. 2016, pp. 740–756.

[58] Garg, R., Kumar, B. V., Carneiro, G., et al. "Unsupervised CNN for single view depth estimation: Geometry to the rescue". In: *European Conference on Computer Vision (ECCV)*. Springer. 2016, pp. 740–756.

[59] Geiger, A., Lenz, P., Stiller, C., et al. "Vision meets Robotics: The KITTI Dataset". In: *International Journal of Robotics Research (IJRR)* (2013).

[60] Geiger, A., Roser, M., and Urtasun, R. "Efficient Large-Scale Stereo Matching". In: *ACCV*. 2010.

[61] Giannarou, S., Visentini-Scarzanella, M., and Yang, G.-Z. "Probabilistic Tracking of Affine-Invariant Anisotropic Regions". In: *IEEE Trans. Pattern Anal. Machine Intell* (2012).

[62] Gidaris, S., Singh, P., and Komodakis, N. "Unsupervised Representation Learning by Predicting Image Rotations". In: *ArXiv* abs/1803.07728 (2018).

[63] Godard, C., Mac Aodha, O., and Brostow, G. J. "Unsupervised Monocular Depth Estimation with Left-Right Consistency". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[64] Godard, C., Mac Aodha, O., Firman, M., et al. "Digging into Self-Supervised Monocular Depth Prediction". In: (Oct. 2019).
[65] Goh, E. Z. and Ali, T. "Robotic surgery: an evolution in practice". In: *Journal of Surgical Protocols and Research Methodologies* 2022.1 (Feb. 2022). snac003. ISSN: 2752-616X. DOI: 10.1093/jsprm/snac003. eprint: https://academic.oup.com/jsprm/article-pdf/2022/1/snac003/48653216/snac003.pdf.
[66] Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.
[67] Goodfellow, I., Pouget-Abadie, J., Mirza, M., et al. "Generative Adversarial Nets". In: ed. by Ghahramani, Z., Welling, M., Cortes, C., et al. Vol. 27. Curran Associates, Inc., 2014.
[68] Gordon, A., Li, H., Jonschkowski, R., et al. "Depth From Videos in the Wild: Unsupervised Monocular Depth Learning From Unknown Cameras". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, Nov. 2019, pp. 8976–8985. DOI: 10.1109/ICCV.2019.00907.
[69] Guizilini, V., Ambrus, R., Pillai, S., et al. "3D Packing for Self-Supervised Monocular Depth Estimation". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.
[70] Guo, X., Yang, K., Yang, W., et al. "Group-wise Correlation Stereo Network". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 3273–3282.
[71] Gutmann, M. and Hyvärinen, A. "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Teh, Y. W. and Titterington, M. Vol. 9. Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR, 13–15 May 2010, pp. 297–304.
[72] Haouchine, N., Dequidt, J., Peterlik, I., et al. "Image-guided Simulation of Heterogeneous Tissue Deformation For Augmented Reality during Hepatic Surgery". In: *ISMAR - IEEE International Symposium on Mixed and Augmented Reality 2013*. Adelaide, Australia, Oct. 2013.
[73] He, K., Chen, X., Xie, S., et al. "Masked Autoencoders Are Scalable Vision Learners". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022, pp. 16000–16009.
[74] He, K., Fan, H., Wu, Y., et al. "Momentum Contrast for Unsupervised Visual Representation Learning". In: *arXiv preprint arXiv:1911.05722* (2019).
[75] He, K., Zhang, X., Ren, S., et al. "Deep Residual Learning for Image Recognition". In: *Computer Vision and Pattern Recognition (CVPR)* abs/1512.03385 (2015). arXiv: 1512.03385.
[76] Higgins, I., Matthey, L., Pal, A., et al. "beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework". In: 2017.
[77] Hinton, G. E., Srivastava, N., Krizhevsky, A., et al. "Improving neural networks by preventing co-adaptation of feature detectors". In: *CoRR* abs/1207.0580 (2012). cite arxiv:1207.0580.
[78] Hirschmüller, H. and Scharstein, D. "Evaluation of Cost Functions for Stereo Matching". In: *2007 IEEE Conference on Computer Vision and Pattern Recognition* (2007), pp. 1–8.
[79] Ho, J., Salimans, T., Gritsenko, A., et al. "Video diffusion models". In: *arXiv:2204.03458* (2022).
[80] Hochreiter, S. and Schmidhuber, J. "Long Short-Term Memory". In: *Neural Computation* 9.8 (1997), pp. 1735–1780.
[81] Hoover, J. M., Chang, S. M., and Parney, I. F. "Clinical Trials in Brain Tumor Surgery". In: *Neuroimaging Clinics of North America* 20.3 (2010). State of the Art Brain Tumor Diagnostics, Imaging, and Therapeutics, pp. 409–424. ISSN: 1052-5149. DOI: https://doi.org/10.1016/j.nic.2010.04.006.

[82] Horé, A. and Ziou, D. "Image Quality Metrics: PSNR vs. SSIM". In: *ICPR*. 2010, pp. 2366–2369. DOI: 10.1109/ICPR.2010.579.
[83] Hu, J., Shen, L., and Sun, G. "Squeeze-and-Excitation Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018.
[84] Huang, G., Liu, Z., Maaten, L. van der, et al. "Densely Connected Convolutional Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017.
[85] Huang, P.-H., Matzen, K., Kopf, J., et al. "DeepMVS: Learning Multi-view Stereopsis". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 2821–2830. DOI: 10.1109/CVPR.2018.00298.
[86] Huang, J.-B., Kang, S. B., Ahuja, N., et al. "Temporally Coherent Completion of Dynamic Video". In: *ACM Trans. Graph.* 35.6 (Nov. 2016), 196:1–196:11. ISSN: 0730-0301. DOI: 10.1145/2980179.2982398.
[87] Huang, Z., Shi, X., Zhang, C., et al. "FlowFormer: A Transformer Architecture for Optical Flow". In: *ECCV* (2022).
[88] Iizuka, S., Simo-Serra, E., and Ishikawa, H. "Globally and Locally Consistent Image Completion". In: *ACM Transactions on Graphics (Proc. of SIGGRAPH 2017)* 36.4 (2017).
[89] Im, S., Jeon, H.-G., Lin, S., et al. "DPSNet: End-to-end deep plane sweep stereo". In: *International Conference on Learning Representations (ICLR)*. May 2019.
[90] Ioffe, S. and Szegedy, C. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Bach, F. and Blei, D. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, July 2015, pp. 448–456.
[91] J Marcus, H., P Cundy, T., Hughes-Hallett, A., et al. "Endoscopic and Keyhole Endoscope-assisted Neurosurgical Approaches: A Qualitative Survey on Technical Challenges and Technological Solutions". In: *British journal of neurosurgery* 28 (Feb. 2014). DOI: 10.3109/02688697.2014.887654.
[92] Jaderberg, M., Simonyan, K., Zisserman, A., et al. "Spatial Transformer Networks". In: *Advances in Neural Information Processing Systems 28*. Ed. by Cortes, C., Lawrence, N. D., Lee, D. D., et al. Curran Associates, Inc., 2015, pp. 2017–2025.
[93] Jaegle, A., Borgeaud, S., Alayrac, J.-B., et al. "Perceiver IO: A General Architecture for Structured Inputs & Outputs". In: *International Conference on Learning Representations*. 2022.
[94] Johnson, J., Alahi, A., and Fei-Fei, L. "Perceptual losses for real-time style transfer and super-resolution". In: *European Conference on Computer Vision*. 2016.
[95] Jolicoeur-Martineau, A. "The relativistic discriminator: a key element missing from standard GAN". In: *arXiv preprint arXiv:1807.00734* (2018).
[96] Jolicoeur-Martineau, A. "The relativistic discriminator: a key element missing from standard GAN". In: *International Conference on Learning Representations*. 2019.
[97] Jonschkowski, R., Stone, A., Barron, J., et al. "What Matters in Unsupervised Optical Flow". In: *ECCV* (2020).
[98] Kan, L. K., Drummond, K., Hunn, M., et al. "Potential biomarkers and challenges in glioma diagnosis, therapy and prognosis". In: *BMJ Neurology Open* 2 (2020), e000069. DOI: 10.1136/bmjno-2020-000069.
[99] Karabeber, H., Huang, R., Iacono, P., et al. "Guiding Brain Tumor Resection Using Surface-Enhanced Raman Scattering Nanoparticles and a Hand-Held Raman Scanner". In: *ACS Nano* 8.10 (2014). PMID: 25093240, pp. 9755–9766. DOI: 10.1021/nn503948b. eprint: https://doi.org/10.1021/nn503948b.
[100] Karnewar, A. and Wang, O. "MSG-GAN: Multi-Scale Gradients for Generative Adversarial Networks". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition,*

CVPR 2020, Seattle, WA, USA, June 13-19, 2020. IEEE, 2020, pp. 7796–7805. DOI: 10.1109/CVPR42600.2020.00782.

[101] Karnewar, A., Wang, O., and Iyengar, R. S. "MSG-GAN: Multi-Scale Gradient GAN for Stable Image Synthesis". In: 2020.

[102] Karras, T., Aila, T., Laine, S., et al. "Progressive Growing of GANs for Improved Quality, Stability, and Variation". In: *International Conference on Learning Representations*. 2018.

[103] Karras, T., Aila, T., Laine, S., et al. "Progressive Growing of GANs for Improved Quality, Stability, and Variation". In: 2018.

[104] Karras, T., Aittala, M., Hellsten, J., et al. "Training Generative Adversarial Networks with Limited Data". In: 2020.

[105] Karras, T., Laine, S., and Aila, T. "A Style-Based Generator Architecture for Generative Adversarial Networks". In: 2019, pp. 4396–4405. DOI: 10.1109/CVPR.2019.00453.

[106] Karras, T., Laine, S., Aittala, M., et al. "Analyzing and Improving the Image Quality of StyleGAN". In: 2020, pp. 8107–8116. DOI: 10.1109/CVPR42600.2020.00813.

[107] Khosla, P., Teterwak, P., Wang, C., et al. "Supervised Contrastive Learning". In: *Advances in Neural Information Processing Systems*. Ed. by Larochelle, H., Ranzato, M., Hadsell, R., et al. Vol. 33. Curran Associates, Inc., 2020, pp. 18661–18673.

[108] Kim, D., Woo, S., Lee, J., et al. "Deep Video Inpainting". In: *CoRR* abs/1905.01639 (2019). arXiv: 1905.01639.

[109] Kingma, D. P. and Ba, J. "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Bengio, Y. and LeCun, Y. 2015.

[110] Kingma, D. P. and Welling, M. "Auto-Encoding Variational Bayes". In: 2014. arXiv: http://arxiv.org/abs/1312.6114v10 [stat.ML].

[111] Kingma, D. P. and Dhariwal, P. "Glow: Generative Flow with Invertible 1x1 Convolutions". In: ed. by Bengio, S., Wallach, H., Larochelle, H., et al. Vol. 31. Curran Associates, Inc., 2018.

[112] Kirillov, A., Mintun, E., Ravi, N., et al. "Segment Anything". In: *arXiv:2304.02643* (2023).

[113] Koh, D., Jang, W., Park, J., et al. "Efficacy and Safety of Robotic Procedures Performed Using the da Vinci Robotic Surgical System at a Single Institute in Korea: Experience with 10000 Cases". In: *Yonsei Medical Journal* 59.8 (Oct. 2018), pp. 975–981. DOI: 10.3349/ymj.2018.59.8.975.

[114] Kolesnikov, A., Dosovitskiy, A., Weissenborn, D., et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: 2021.

[115] Kopf, J., Rong, X., and Huang, J.-B. "Robust Consistent Video Depth Estimation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021.

[116] Kreuzberger, D., Kühl, N., and Hirschl, S. "Machine Learning Operations (MLOps): Overview, Definition, and Architecture". In: *IEEE Access* 11 (2022), pp. 31866–31879.

[117] Kuprieiev, R., skshetry, () P. R., et al. *DVC: Data Version Control - Git for Data & Models*. Version 2.55.0. Apr. 2023. DOI: 10.5281/zenodo.7848331.

[118] Kusupati, U., Cheng, S., Chen, R., et al. "Normal Assisted Stereo Depth Estimation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[119] Kuznietsov, Y., Proesmans, M., and Van Gool, L. "CoMoDA: Continuous Monocular Depth Adaptation Using Past Experiences". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Jan. 2021, pp. 2907–2917.

[120] Lai, W.-S., Huang, J.-B., Wang, O., et al. "Learning Blind Video Temporal Consistency". In: *European Conference on Computer Vision*. 2018.

[121] Lamarca, J. and Montiel, J. "Camera tracking for SLAM in deformable maps". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018.

[122] Lamarca, J., Parashar, S., Bartoli, A., et al. "DefSLAM: Tracking and Mapping of Deforming Scenes from Monocular Sequences". In: *arXiv preprint arXiv:1908.08918* (2019).
[123] Lee, S., Oh, S. W., Won, D., et al. "Copy-and-Paste Networks for Deep Video Inpainting". In: *International Conference on Computer Vision (ICCV)* (2019).
[124] Li, S., Luo, Y., Zhu, Y., et al. "Enforcing Temporal Consistency in Video Depth Estimation". In: *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*. 2021, pp. 1145–1154. DOI: 10.1109/ICCVW54120.2021.00134.
[125] Li, Z., Liu, X., Drenkow, N., et al. "Revisiting Stereo Depth Estimation From a Sequence-to-Sequence Perspective With Transformers". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2021, pp. 6197–6206.
[126] Li, Z., Niklaus, S., Snavely, N., et al. "Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021.
[127] Liang, Z., Feng, Y., Guo, Y., et al. "Learning for Disparity Estimation Through Feature Constancy". In: Dec. 2017. DOI: 10.1109/CVPR.2018.00297.
[128] Lim, B., Arık, S. Ö., Loeff, N., et al. "Temporal Fusion Transformers for interpretable multi-horizon time series forecasting". In: *International Journal of Forecasting* 37.4 (2021), pp. 1748–1764. DOI: 10.1016/j.ijforecast.2021.
[129] Lipson, L., Teed, Z., and Deng, J. "RAFT-Stereo: Multilevel Recurrent Field Transforms for Stereo Matching". In: *arXiv preprint arXiv:2109.07547* (2021).
[130] Liu, C., Gu, J., Kim, K., et al. "Neural RGB®D Sensing: Depth and Uncertainty From a Video Camera". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), pp. 10978–10987.
[131] Liu, F., Shen, C., Lin, G., et al. "Learning Depth from Single Monocular Images Using Deep Convolutional Neural Fields". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 38.10 (Oct. 2016), pp. 2024–2039. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2015.2505283.
[132] Liu, G., Reda, F. A., Shih, K. J., et al. "Image Inpainting for Irregular Holes Using Partial Convolutions". In: *ECCV* abs/1804.07723 (2018). arXiv: 1804.07723.
[133] Liu, Y., Ott, M., Goyal, N., et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. cite arxiv:1907.11692. 2019.
[134] Lu, J., Jayakumari, A., Richter, F., et al. "SuPer Deep: A Surgical Perception Framework for Robotic Tissue Manipulation using Deep Learning for Feature Extraction". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)* (2020), pp. 4783–4789.
[135] Lu, J., Jayakumari, A., Richter, F., et al. "SuPer Deep: A Surgical Perception Framework for Robotic Tissue Manipulation using Deep Learning for Feature Extraction". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. Xi'an, China: IEEE Press, 2021, pp. 4783–4789. DOI: 10.1109/ICRA48506.2021.9561249.
[136] Luo, X., Huang, J., Szeliski, R., et al. "Consistent Video Depth Estimation". In: 39.4 (2020).
[137] Madry, A., Makelov, A., Schmidt, L., et al. "Towards Deep Learning Models Resistant to Adversarial Attacks". In: *International Conference on Learning Representations*. 2018.
[138] Mantiuk, R., Kim, K. J., Rempel, A. G., et al. "HDR-VDP-2: A Calibrated Visual Metric for Visibility and Quality Predictions in All Luminance Conditions". In: *ACM Trans. Graph.* 30.4 (July 2011). ISSN: 0730-0301. DOI: 10.1145/2010324.1964935.
[139] Mao, X., Li, Q., Xie, H., et al. "Multi-class Generative Adversarial Networks with the L2 Loss Function". In: *Computer Vision and Pattern Recognition (CVPR)* abs/1611.04076 (2016). arXiv: 1611.04076.
[140] Mao, X., Li, Q., Xie, H., et al. "Least Squares Generative Adversarial Networks". In: *2017 IEEE International Conference on Computer Vision (ICCV)* (2017), pp. 2813–2821.

[141] Mayer, N., Ilg, E., Häusser, P., et al. "A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation". In: *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. arXiv:1512.02134. 2016.

[142] Mayer, N., Ilg, E., Häusser, P., et al. "A Large@articleYe2017SelfSupervisedSL, title=Self-Supervised Siamese Learning on Stereo Image Pairs for Depth Estimation in Robotic Surgery, author=Menglong Ye and Edward Johns and A. Handa and L. Zhang and P. Pratt and G. Yang, journal=ArXiv, year=2017, volume=abs/1705.08260 Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 4040–4048.

[143] Mayer, N., Ilg, E., Häusser, P., et al. "A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pp. 4040–4048. DOI: 10.1109/CVPR.2016.438.

[144] McCraith, R., Neumann, L., Zisserman, A., et al. "Monocular Depth Estimation with Self-supervised Instance Adaptation". In: *ArXiv* abs/2004.05821 (2020).

[145] McMahon, D., Gleeson, J., O'Reilly, S., et al. "Management of newly diagnosed glioblastoma multiforme: current state of the art and emerging therapeutic approaches". In: *Medical Oncology* 39 (June 2022), p. 129. DOI: 10.1007/s12032-022-01708-w.

[146] Menze, M., Heipke, C., and Geiger, A. "Object Scene Flow". In: *ISPRS Journal of Photogrammetry and Remote Sensing (JPRS)* (2018).

[147] Miangoleh, S. M. H., Dille, S., Mai, L., et al. "Boosting Monocular Depth Estimation Models to High-Resolution via Content-Adaptive Multi-Resolution Merging". In: 2021.

[148] Mildenhall, B., Srinivasan, P. P., Tancik, M., et al. "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis". In: *ECCV*. 2020.

[149] Mnih, V., Kavukcuoglu, K., Silver, D., et al. "Human-level control through deep reinforcement learning". In: *Nature* 518.7540 (Feb. 2015), pp. 529–533. ISSN: 00280836.

[150] Nair, V. and Hinton, G. E. "Rectified linear units improve restricted boltzmann machines". In: *ICML 2010*. 2010, pp. 807–814.

[151] Nguyen, H. and La, H. "Review of Deep Reinforcement Learning for Robot Manipulation". In: *2019 Third IEEE International Conference on Robotic Computing (IRC)*. 2019, pp. 590–595. DOI: 10.1109/IRC.2019.00120.

[152] Nichol, A. Q., Dhariwal, P., Ramesh, A., et al. "GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models". In: ed. by Chaudhuri, K., Jegelka, S., Song, L., et al. Vol. 162. Proceedings of Machine Learning Research. PMLR, 17–23 Jul 2022, pp. 16784–16804.

[153] Noroozi, M. and Favaro, P. "Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles". In: *European Conference on Computer Vision*. 2016.

[154] Oh, J., Sahgal, A., Sanghera, P., et al. "Glioblastoma: Patterns of Recurrence and Efficacy of Salvage Treatments". In: *Canadian Journal of Neurological Sciences* 38.4 (2011), pp. 621–625. DOI: 10.1017/S0317167100012166.

[155] Oord, A. van den, Vinyals, O., and, k. kavukcuoglu koray. "Neural Discrete Representation Learning". In: *Advances in Neural Information Processing Systems*. Vol. 30. 2017.

[156] Oord, A. van den, Li, Y., and Vinyals, O. "Representation Learning with Contrastive Predictive Coding". In: *ArXiv* abs/1807.03748 (2018).

[157] Pandey, A., Sonawane, M., and Mamtani, S. *Deployment of ML Models using Kubeflow on Different Cloud Providers*. 2022. arXiv: 2206.13655 [cs.LG].

[158] Papadakis, N. and Caselles, V. "Multi-label Depth Estimation for Graph Cuts Stereo Problems". In: *Journal of Mathematical Imaging and Vision* 38 (2010), pp. 70–82. DOI: 10.1007/s10851-010-0212-8.

[159] Papernot, N., McDaniel, P. D., Jha, S., et al. "The Limitations of Deep Learning in Adversarial Settings." In: *CoRR* abs/1511.07528 (2015).

[160] Patil, V., Gansbeke, W. V., Dai, D., et al. "Don't Forget The Past: Recurrent Depth Estimation from Monocular Video". In: *CoRR* abs/2001.02613 (2020). arXiv: 2001.02613.

[161] Patil, V., Gansbeke, W. V., Dai, D., et al. "Don't Forget The Past: Recurrent Depth Estimation from Monocular Video". In: *IEEE Robotics and Automation Letters* 5 (2020), pp. 6813–6820.

[162] Payne, C. J. and Yang, G.-Z. "Hand-held medical robots". In: *Annals of Biomedical Engineering* 42.8 (Aug. 2014). Epub 2014 Jun 14, pp. 1594–1605. DOI: 10.1007/s10439-014-1042-4.

[163] Pillai, S., Ambrus, R., and Gaidon, A. "SuperDepth: Self-Supervised, Super-Resolved Monocular Depth Estimation". In: *2019 International Conference on Robotics and Automation (ICRA)* (2019), pp. 9250–9256.

[164] Pilzer, A., Xu, D., Puscas, M., et al. "Unsupervised Adversarial Depth Estimation Using Cycled Generative Networks". In: *2018 International Conference on 3D Vision (3DV)*. 2018, pp. 587–595. DOI: 10.1109/3DV.2018.00073.

[165] Pilzer, A., Lathuilière, S., Xu, D., et al. "Progressive Fusion for Unsupervised Binocular Depth Estimation using Cycled Networks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019).

[166] Pilzer, A., Xu, D., Puscas, M., et al. "Unsupervised Adversarial Depth Estimation using Cycled Generative Networks". In: *2018 International Conference on 3D Vision (3DV)*. IEEE. 2018, pp. 587–595.

[167] Poggi, M., Tosi, F., and Mattoccia, S. "Learning Monocular Depth Estimation with Unsupervised Trinocular Assumptions". In: *2018 International Conference on 3D Vision (3DV)* (2018), pp. 324–333.

[168] Poggi, M., Tosi, F., and Mattoccia, S. "Learning monocular depth estimation with unsupervised trinocular assumptions". In: *6th International Conference on 3D Vision (3DV)*. 2018.

[169] Pont-Tuset, J., Perazzi, F., Caelles, S., et al. "The 2017 DAVIS Challenge on Video Object Segmentation". In: *arXiv:1704.00675* (2017).

[170] Radford, A., Metz, L., and Chintala, S. "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks". In: *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*. Ed. by Bengio, Y. and LeCun, Y. 2016.

[171] Ranftl, R., Bochkovskiy, A., and Koltun, V. "Vision Transformers for Dense Prediction". In: *ICCV* (2021).

[172] Ranftl, R., Lasinger, K., Hafner, D., et al. "Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2020).

[173] Ravindra Padalkar, G., Dinkar Patil, S., Mallikarjun Hegadi, M., et al. "Drug Discovery using Generative Adversarial Network with Reinforcement Learning". In: *2021 International Conference on Computer Communication and Informatics (ICCCI)*. 2021, pp. 1–3. DOI: 10.1109/ICCCI50826.2021.9402449.

[174] Reddy, K., Gharde, P., Tayade, H., et al. "Advancements in Robotic Surgery: A Comprehensive Overview of Current Utilizations and Upcoming Frontiers". In: *Cureus* 15.12 (Dec. 2023), e50415. DOI: 10.7759/cureus.50415.

[175] Redmon, J., Divvala, S., Girshick, R., et al. *You Only Look Once: Unified, Real-Time Object Detection*. cite arxiv:1506.02640. 2015.

[176] Rieke, N., Hancox, J., Li, W., et al. "The future of digital health with federated learning". English. In: *npj Digital Medicine* 3.1 (Dec. 2020). ISSN: 2398-6352. DOI: 10.1038/s41746-020-00323-1.

[177] Rivero-Moreno, Y., Echevarria, S., Vidal-Valderrama, C., et al. "Robotic Surgery: A Comprehensive Review of the Literature and Current Trends". In: *Cureus* 15.7 (July 2023), e42370. DOI: 10.7759/cureus.42370.

[178] Robinson, R. D. "Reliable Machine Learning for Medical Imaging Data through Automated Quality Control and Data Harmonization". In: (July 2020).

[179] Rombach, R., Blattmann, A., Lorenz, D., et al. *High-Resolution Image Synthesis with Latent Diffusion Models*. 2021. arXiv: 2112.10752 [cs.CV].

[180] Rombach, R., Blattmann, A., Lorenz, D., et al. "High-Resolution Image Synthesis with Latent Diffusion Models". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022.

[181] Rong, L., Li, N., and Zhang, Z. "Emerging therapies for glioblastoma: current state and future directions". In: *Journal of Experimental & Clinical Cancer Research* 41 (Apr. 2022), p. 142. DOI: 10.1186/s13046-022-02349-7.

[182] Ronneberger, O., P.Fischer, and Brox, T. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Vol. 9351. LNCS. (available on arXiv:1505.04597 [cs.CV]). Springer, 2015, pp. 234–241.

[183] Saharia, C., Chan, W., Saxena, S., et al. "Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding". In: *Advances in Neural Information Processing Systems*. Ed. by Oh, A. H., Agarwal, A., Belgrave, D., et al. 2022.

[184] Salimans, T., Goodfellow, I., Zaremba, W., et al. "Improved Techniques for Training GANs". In: ed. by Lee, D., Sugiyama, M., Luxburg, U., et al. Vol. 29. Curran Associates, Inc., 2016.

[185] Sastry, R. A., Bi, W. L., Pieper, S. D., et al. "Applications of Ultrasound in the Resection of Brain Tumors". In: *Journal of Neuroimaging* 27 (2017).

[186] Sattarzadeh, S., Sudhakar, M., Plataniotis, K. N., et al. "Integrated Grad-Cam: Sensitivity-Aware Visual Explanation of Deep Convolutional Networks Via Integrated Gradient-Based Scoring". In: *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2021), pp. 1775–1779.

[187] Sauer, A., Karras, T., Laine, S., et al. "StyleGAN-T: Unlocking the Power of GANs for Fast Large-Scale Text-to-Image Synthesis". In: vol. abs/2301.09515. 2023.

[188] Scherer, D., Müller, A. C., and Behnke, S. "Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition." In: *ICANN (3)*. Ed. by Diamantaras, K. I., Duch, W., and Iliadis, L. S. Vol. 6354. Lecture Notes in Computer Science. Springer, 2010, pp. 92–101. ISBN: 978-3-642-15824-7.

[189] Schöps, T., Schönberger, J. L., Galliani, S., et al. "A Multi-View Stereo Benchmark with High-Resolution Images and Multi-Camera Videos". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[190] Sculley, D., Holt, G., Golovin, D., et al. "Hidden Technical Debt in Machine Learning Systems." In: *NIPS*. Ed. by Cortes, C., Lawrence, N. D., Lee, D. D., et al. 2015, pp. 2503–2511.

[191] Senoo, T., Wakunami, K., Sasaki, H., et al. "Fast Depth Estimation Using Non-Iterative Local Optimization for Super Multi-View Images". In: *GlobalSIP 2015*. Presentation Slides. IEEE. Dec. 2015.

[192] Shen, Z., Zhang, M., Zhao, H., et al. "Efficient Attention: Attention with Linear Complexities". In: *WACV*. 2021.

[193] Shi, W., Caballero, J., Huszar, F., et al. "Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network." In: *CVPR*. IEEE Computer Society, 2016, pp. 1874–1883. ISBN: 978-1-4673-8851-1.

[194] Shi, X., Chen, Z., Wang, H., et al. "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting". In: *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'15. Montreal, Canada: MIT Press, 2015, pp. 802–810.

[195] Simonyan, K. and Zisserman, A. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *International Conference on Learning Representations*. 2015.

[196] Singer, U., Polyak, A., Hayes, T., et al. "Make-A-Video: Text-to-Video Generation without Text-Video Data". In: *The Eleventh International Conference on Learning Representations*. 2023.

[197] Sinha, R., Raje, S. R., and Rao, G. "Three-dimensional laparoscopy: Principles and practice". In: *Journal of Minimal Access Surgery* 13 (2016), pp. 165–169.

[198] Sitzmann, V., Martel, J., Bergman, A., et al. "Implicit Neural Representations with Periodic Activation Functions". In: *Advances in Neural Information Processing Systems*. Ed. by Larochelle, H., Ranzato, M., Hadsell, R., et al. Vol. 33. Curran Associates, Inc., 2020, pp. 7462–7473.

[199] Sogancioglu, E., Hu, S., Belli, D., et al. "Chest X-ray Inpainting with Deep Generative Models". In: *Med. Imaging with Deep Leanring* abs/1809.01471 (2018). arXiv: 1809.01471.

[200] Song, J., Wang, J., Zhao, L., et al. "MIS-SLAM: Real-Time Large-Scale Dense Deformable SLAM System in Minimal Invasive Surgery Based on Heterogeneous Computing". In: *IEEE Robotics and Automation Letters* 3 (2018), pp. 4068–4075.

[201] Stoyanov, D., Scarzanella, M. V., Pratt, P., et al. "Real-Time Stereo Reconstruction in Robotically Assisted Minimally Invasive Surgery". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2010*. Ed. by Jiang, T., Navab, N., Pluim, J. P. W., et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 275–282. ISBN: 978-3-642-15705-9.

[202] Stupp, R., Hegi, M. E., Bent, M. J. van den, et al. "Changing Paradigms—An Update on the Multidisciplinary Management of Malignant Glioma". In: *The Oncologist* 11.2 (Feb. 2006), pp. 165–180. ISSN: 1083-7159. DOI: 10.1634/theoncologist.11-2-165. eprint: https://academic.oup.com/oncolo/article-pdf/11/2/165/41850802/oncolo_11_2_165.pdf.

[203] Sun, D., Yang, X., Liu, M.-Y., et al. "PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume". In: 2018.

[204] Sutskever, I., Martens, J., Dahl, G., et al. "On the importance of initialization and momentum in deep learning". In: *Proceedings of the 30th International Conference on Machine Learning*. Ed. by Dasgupta, S. and McAllester, D. Vol. 28. Proceedings of Machine Learning Research 3. Atlanta, Georgia, USA: PMLR, 17–19 Jun 2013, pp. 1139–1147.

[205] Suvorov, R., Logacheva, E., Mashikhin, A., et al. "Resolution-robust Large Mask Inpainting with Fourier Convolutions". In: *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* (2021), pp. 3172–3182.

[206] Szegedy, C., Liu, W., Jia, Y., et al. "Going Deeper with Convolutions". In: *CVPR* abs/1409.4842 (2014). arXiv: 1409.4842.

[207] Szeliski, R. *Computer vision algorithms and applications*. 2011. URL: http://dx.doi.org/10.1007/978-1-84882-935-0.

[208] Tanaka, H., Kunin, D., Yamins, D. L., et al. "Pruning neural networks without any data by iteratively conserving synaptic flow". In: *Advances in Neural Information Processing Systems*. Ed. by Larochelle, H., Ranzato, M., Hadsell, R., et al. Vol. 33. Curran Associates, Inc., 2020, pp. 6377–6389.

[209] Tancik, M., Casser, V., Yan, X., et al. "Block-NeRF: Scalable Large Scene Neural View Synthesis". In: *arXiv* (2022).

[210] Tankovich, V., Häne, C., Fanello, S., et al. "HITNet: Hierarchical Iterative Tile Refinement Network for Real-time Stereo Matching". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), pp. 14357–14367.

[211] Teed, Z. and Deng, J. "DeepV2D: Video to Depth with Differentiable Structure from Motion". In: *International Conference on Learning Representations*. 2020.

[212] Teed, Z. and Deng, J. "RAFT: Recurrent All-Pairs Field Transforms for Optical Flow". In: *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II*. Glasgow, United Kingdom: Springer-Verlag, 2020, pp. 402–419. ISBN: 978-3-030-58535-8. DOI: 10.1007/978-3-030-58536-5_24.

[213] Tobin, J., Fong, R., Ray, A., et al. "Domain randomization for transferring deep neural networks from simulation to the real world". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2017), pp. 23–30.

[214] Tonioni, A., Tosi, F., Poggi, M., et al. "Real-time self-adaptive deep stereo". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

[215] Tremblay, J., To, T., and Birchfield, S. "Falling Things: A Synthetic Dataset for 3D Object Detection and Pose Estimation". In: *CVPR Workshop on Real World Challenges and New Benchmarks for Deep Learning in Robotic Vision*. June 2018.

[216] Tukra, S. and Giannarou, S. "Randomly connected neural networks for self-supervised monocular depth estimation". In: *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization* 10.4 (2022), pp. 390–399. DOI: 10.1080/21681163.2021.1997648. eprint: https://doi.org/10.1080/21681163.2021.1997648.

[217] Tukra, S. and Giannarou, S. "Stereo Depth Estimation via Self-supervised Contrastive Representation Learning". In: *Medical Image Computing and Computer Assisted Intervention–MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part VII*. Springer. 2022, pp. 604–614.

[218] Tukra, S., Lidströmer, N., Ashrafian, H., et al. "AI in Surgical Robotics". In: *Artificial Intelligence in Medicine*. Ed. by Lidströmer, N. and Ashrafian, H. Cham: Springer International Publishing, 2020, pp. 1–20. ISBN: 978-3-030-58080-3. DOI: 10.1007/978-3-030-58080-3_323-1.

[219] Tukra, S., Marcus, H. J., and Giannarou, S. "See-Through Vision With Unsupervised Scene Occlusion Reconstruction". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.7 (2022), pp. 3779–3790. DOI: 10.1109/TPAMI.2021.3058410.

[220] Vaswani, A., Shazeer, N., Parmar, N., et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Ed. by Guyon, I., Luxburg, U. V., Bengio, S., et al. Vol. 30. Curran Associates, Inc., 2017.

[221] Wang, H., Liu, T., Chen, J., et al. "Full-Perception Robotic Surgery Environment with Anti-Occlusion Global–Local Joint Positioning". In: *Sensors* 23.20 (2023). ISSN: 1424-8220. DOI: 10.3390/s23208637.

[222] Wang, J., Zhang, G., Wu, Z., et al. "Self-Supervised Joint Learning Framework of Depth Estimation via Implicit Cues". In: *ArXiv* abs/2006.09876 (2020).

[223] Wang, R., Pizer, S. M., and Frahm, J.-M. "Recurrent Neural Network for (Un-) supervised Learning of Monocular Video Visual Odometry and Depth". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 5555–5564.

[224] Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., et al. "High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs". In: 2018.

[225] Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., et al. "Video-to-Video Synthesis". In: *Advances in Neural Information Processing Systems*. Ed. by Bengio, S., Wallach, H., Larochelle, H., et al. Vol. 31. Curran Associates, Inc., 2018.

[226] Wang, W., Zhu, D., Wang, X., et al. "TartanAir: A Dataset to Push the Limits of Visual SLAM". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020.

[227] Wang, X., Yu, K., Wu, S., et al. "ESRGAN: Enhanced super-resolution generative adversarial networks". In: Sept. 2018.

[228] Wang, Y., Wang, P., Yang, Z., et al. "UnOS: Unified Unsupervised Optical-Flow and Stereo-Depth Estimation by Watching Videos". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

[229] Wang, Y., Yang, Y., Yang, Z., et al. "Occlusion Aware Unsupervised Learning of Optical Flow". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018.

[230] Wang, Z., Simoncelli, E., and Bovik, A. "Multiscale structural similarity for image quality assessment". In: *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers*. Vol. 2. 2003, 1398–1402 Vol.2. DOI: 10.1109/ACSSC.2003.1292216.

[231] Wang, Z., Bovik, A. C., Sheikh, H. R., et al. "Image Quality Assessment: From Error Visibility to Structural Similarity". In: *IEEE Trans. on Image Processing* 13.4 (Apr. 2004), pp. 600–612. ISSN: 1057-7149. DOI: 10.1109/TIP.2003.819861.

[232] Wang, Z., Bovik, A. C., Sheikh, H. R., et al. "Image Quality Assessment: From Error Visibility to Structural Similarity". In: *IEEE TRANSACTIONS ON IMAGE PROCESSING* 13.4 (2004), pp. 600–612.

[233] Wang, Z., Bovik, A. C., Sheikh, H. R., et al. "Image quality assessment: from error visibility to structural similarity." In: *IEEE Transactions on Image Processing* 13.4 (Dec. 21, 2005), pp. 600–612.

[234] Watson, J., Aodha, O. M., Prisacariu, V., et al. "The Temporal Opportunist: Self-Supervised Multi-Frame Monocular Depth". In: *Computer Vision and Pattern Recognition (CVPR)*. 2021.

[235] Watson, J., Firman, M., Brostow, G. J., et al. "Self-Supervised Monocular Depth Hints". In: *The International Conference on Computer Vision (ICCV)*. Oct. 2019.

[236] Watts, D. and Strogatz, S. "Collective dynamics of 'small-world' networks". In: *Nature* 393 (1998), pp. 440–442.

[237] Wei, X., Zhang, Y., Li, Z., et al. "DeepSFM: Structure from Motion via Deep Bundle Adjustment". In: *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I*. Glasgow, United Kingdom: Springer-Verlag, 2020, pp. 230–247. ISBN: 978-3-030-58451-1. DOI: 10.1007/978-3-030-58452-8_14.

[238] Wexler, Y., Shechtman, E., and Irani, M. "Space-Time Completion of Video". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.3 (Mar. 2007), pp. 463–476. DOI: 10.1109/TPAMI.2007.60.

[239] Xian, K., Shen, C., Cao, Z., et al. "Monocular Relative Depth Perception with Web Stereo Data Supervision". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 311–320. DOI: 10.1109/CVPR.2018.00040.

[240] Xie, D., Chen, W., Zhao, J., et al. "Surgical instruments hyalinization: Occlusion removal in minimally invasive endoscopic surgery". In: *Biomimetic Intelligence and Robotics* 3.3 (2023), p. 100105. ISSN: 2667-3797. DOI: https://doi.org/10.1016/j.birob.2023.100105.

[241] Xie, J., Girshick, R. B., and Farhadi, A. "Deep3D: Fully Automatic 2D-to-3D Video Conversion with Deep Convolutional Neural Networks". In: *European Conf. Computer Vision (ECCV)*. 2016.

[242] Xie, S., Kirillov, A., Girshick, R. B., et al. "Exploring Randomly Wired Neural Networks for Image Recognition". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019), pp. 1284–1293.

[243] Xie, Z., Zhang, Z., Cao, Y., et al. "SimMIM: A Simple Framework for Masked Image Modeling". In: 2022.
[244] Xu, N., Yang, L., Fan, Y., et al. "YouTube-VOS: Sequence-to-Sequence Video Object Segmentation". In: *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part V*. Munich, Germany: Springer-Verlag, 2018, pp. 603–619. ISBN: 978-3-030-01227-4. DOI: 10.1007/978-3-030-01228-1_36.
[245] Y. Cao, Z. W. and Shen, C. "Estimating Depth From Monocular Images as Classification Using Deep Fully Convolutional Residual Networks". In: 2018, pp. 3174–3182.
[246] Yamaguchi, K., McAllester, D. A., and Urtasun, R. "Efficient Joint Segmentation, Occlusion Labeling, Stereo and Flow Estimation". In: *ECCV*. 2014.
[247] Yao, Y., Luo, Z., Li, S., et al. "MVSNet: Depth Inference for Unstructured Multi-view Stereo". In: *ECCV*. 2018.
[248] Yao, Y., Luo, Z., Li, S., et al. "Recurrent MVSNet for High-resolution Multi-view Stereo Depth Inference". In: *Computer Vision and Pattern Recognition (CVPR)* (2019).
[249] Ye, M., Johns, E., Handa, A., et al. "Self-Supervised Siamese Learning on Stereo Image Pairs for Depth Estimation in Robotic Surgery". In: *ArXiv* abs/1705.08260 (2017).
[250] Yin, W., Zhang, J., Wang, O., et al. "Learning to Recover 3D Scene Shape from a Single Image". In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (CVPR)*. 2021.
[251] Yosinski, J., Clune, J., Bengio, Y., et al. "How transferable are features in deep neural networks?" In: *Advances in Neural Information Processing Systems*. Ed. by Ghahramani, Z., Welling, M., Cortes, C., et al. Vol. 27. Curran Associates, Inc., 2014.
[252] Yu, J., Lin, Z., Yang, J., et al. "Generative Image Inpainting with Contextual Attention". In: *CVPR* abs/1801.07892 (2018). arXiv: 1801.07892.
[253] Yu, J., Wang, Z., Vasudevan, V., et al. "CoCa: Contrastive Captioners are Image-Text Foundation Models". In: *Transactions on Machine Learning Research* (2022). ISSN: 2835-8856.
[254] Zaharia, M. A., Chen, A., Davidson, A., et al. "Accelerating the Machine Learning Lifecycle with MLflow". In: *IEEE Data Eng. Bull.* 41 (2018), pp. 39–45.
[255] Zhan, J., Cartucho, J., and Giannarou, S. "Autonomous Tissue Scanning under Free-Form Motion for Intraoperative Tissue Characterisation". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)* (2020), pp. 11147–11154.
[256] Zhang, F., Prisacariu, V., Yang, R., et al. "GA-Net: Guided Aggregation Net for End-to-end Stereo Matching". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 185–194.
[257] Zhang, H., Shen, C., Li, Y., et al. "Exploiting Temporal Consistency for Real-Time Video Depth Estimation". In: 2019.
[258] Zhang, L., Zhang, L., Mou, X., et al. "FSIM: A Feature Similarity Index for Image Quality Assessment". In: *IEEE Transactions on Image Processing* 20.8 (2011), pp. 2378–2386. DOI: 10.1109/TIP.2011.2109730.
[259] Zhang, R., Isola, P., Efros, A. A., et al. "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric". In: *CVPR*. 2018.
[260] Zhang, R., Isola, P., and Efros, A. A. "Colorful Image Colorization." In: *ECCV (3)*. Ed. by Leibe, B., Matas, J., Sebe, N., et al. Vol. 9907. Lecture Notes in Computer Science. Springer, 2016, pp. 649–666. ISBN: 978-3-319-46486-2.
[261] Zhang, Z., Cole, F., Tucker, R., et al. "Consistent depth of moving objects in video". In: *ACM Transactions on Graphics (TOG)* 40.4 (2021), pp. 1–12.
[262] Zhao, H., Gallo, O., Frosio, I., et al. "Loss Functions for Image Restoration With Neural Networks". In: *IEEE Transactions on Computational Imaging* 3.1 (2017), pp. 47–57. DOI: 10.1109/TCI.2016.2644865.

[263] Zhao, L., Wang, J., Pang, L., et al. "GANsDTA: Predicting Drug-Target Binding Affinity Using GANs". In: *Frontiers in Genetics* 10 (2020). ISSN: 1664-8021. DOI: 10.3389/fgene.2019.01243.

[264] Zhong, Y., Dai, Y., and Li, H. "Self-Supervised Learning for Stereo Matching with Self-Improving Ability". In: *ArXiv* abs/1709.00930 (2017).

[265] Zhou, T., Brown, M., Snavely, N., et al. "Unsupervised Learning of Depth and Ego-Motion from Video". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 6612–6619. DOI: 10.1109/CVPR.2017.700.

[266] Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N., et al. "Unet++: A Nested U-Net Architecture for Medical Image Segmentation". In: *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Springer, 2018, pp. 3–11.

Appendices

9.4 Permissions for Reuse of Materials

For papers currently submitted and pending acceptance (i.e. chapters 6 and 7) the same rules apply.

9.4.1 IEEE

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

- In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- In the case of illustrations or tabular material, we require that the copyright line [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

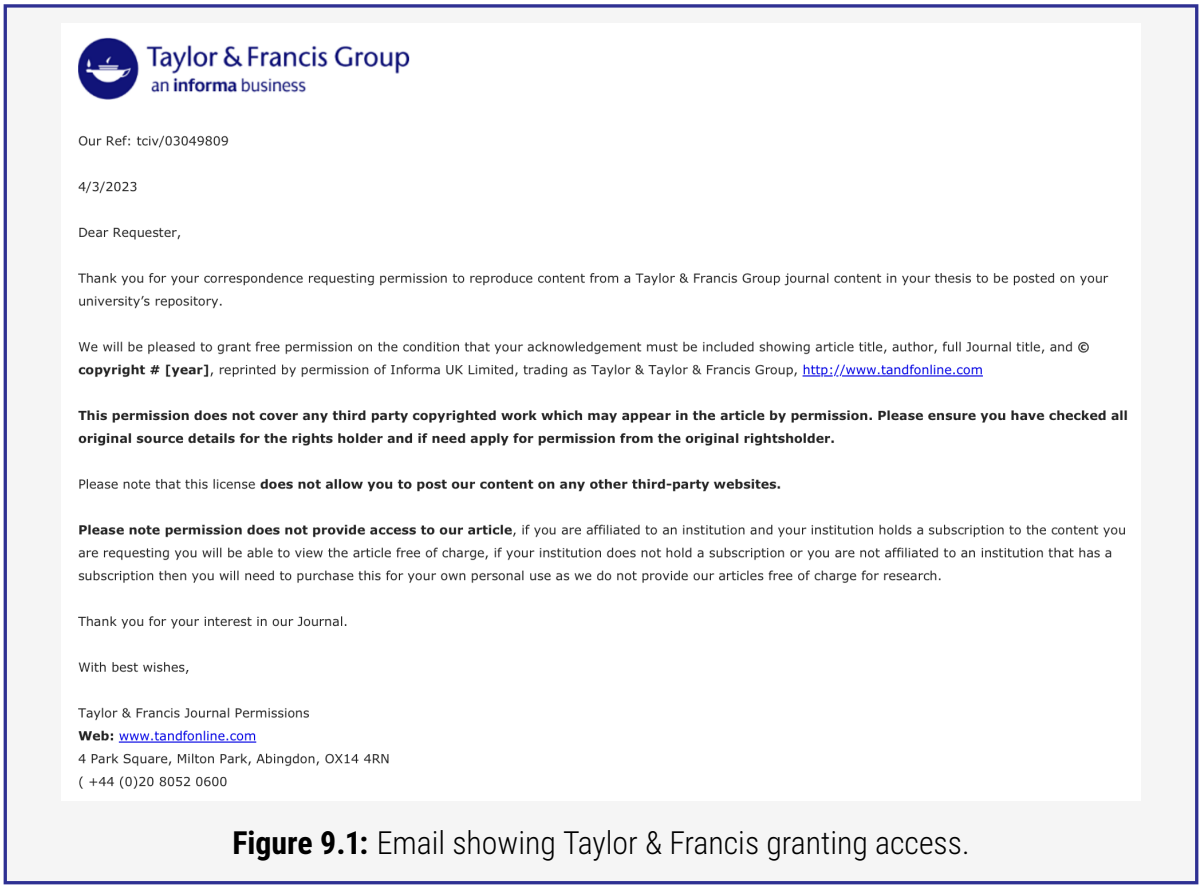
9.4.2 Taylor & Francis

License for reuse of the following paper (presented in chapter 4) was granted:

Samyakh Tukra & Stamatia Giannarou (2022) Randomly connected neural networks for self-supervised monocular depth estimation, Computer Methods in Biomechanics and Biomedical Engineering: Imaging Visualization, 10:4, 390-399

Fig 9.1 shows the license being granted, where the following rules apply:

"Reuse in a Dissertation/Thesis: A dissertation/thesis license authorizes an advanced degree candidate to reproduce portions of the content in his/her doctoral thesis or dissertation. For full article/chapter content, please contact publisher directly to obtain approval**"**



Taylor & Francis Group
an informa business

Our Ref: tciv/03049809

4/3/2023

Dear Requester,

Thank you for your correspondence requesting permission to reproduce content from a Taylor & Francis Group journal content in your thesis to be posted on your university's repository.

We will be pleased to grant free permission on the condition that your acknowledgement must be included showing article title, author, full Journal title, and © **copyright # [year]**, reprinted by permission of Informa UK Limited, trading as Taylor & Taylor & Francis Group, http://www.tandfonline.com

This permission does not cover any third party copyrighted work which may appear in the article by permission. Please ensure you have checked all original source details for the rights holder and if need apply for permission from the original rightsholder.

Please note that this license **does not allow you to post our content on any other third-party websites.**

Please note permission does not provide access to our article, if you are affiliated to an institution and your institution holds a subscription to the content you are requesting you will be able to view the article free of charge, if your institution does not hold a subscription or you are not affiliated to an institution that has a subscription then you will need to purchase this for your own personal use as we do not provide our articles free of charge for research.

Thank you for your interest in our Journal.

With best wishes,

Taylor & Francis Journal Permissions
Web: www.tandfonline.com
4 Park Square, Milton Park, Abingdon, OX14 4RN
(+44 (0)20 8052 0600

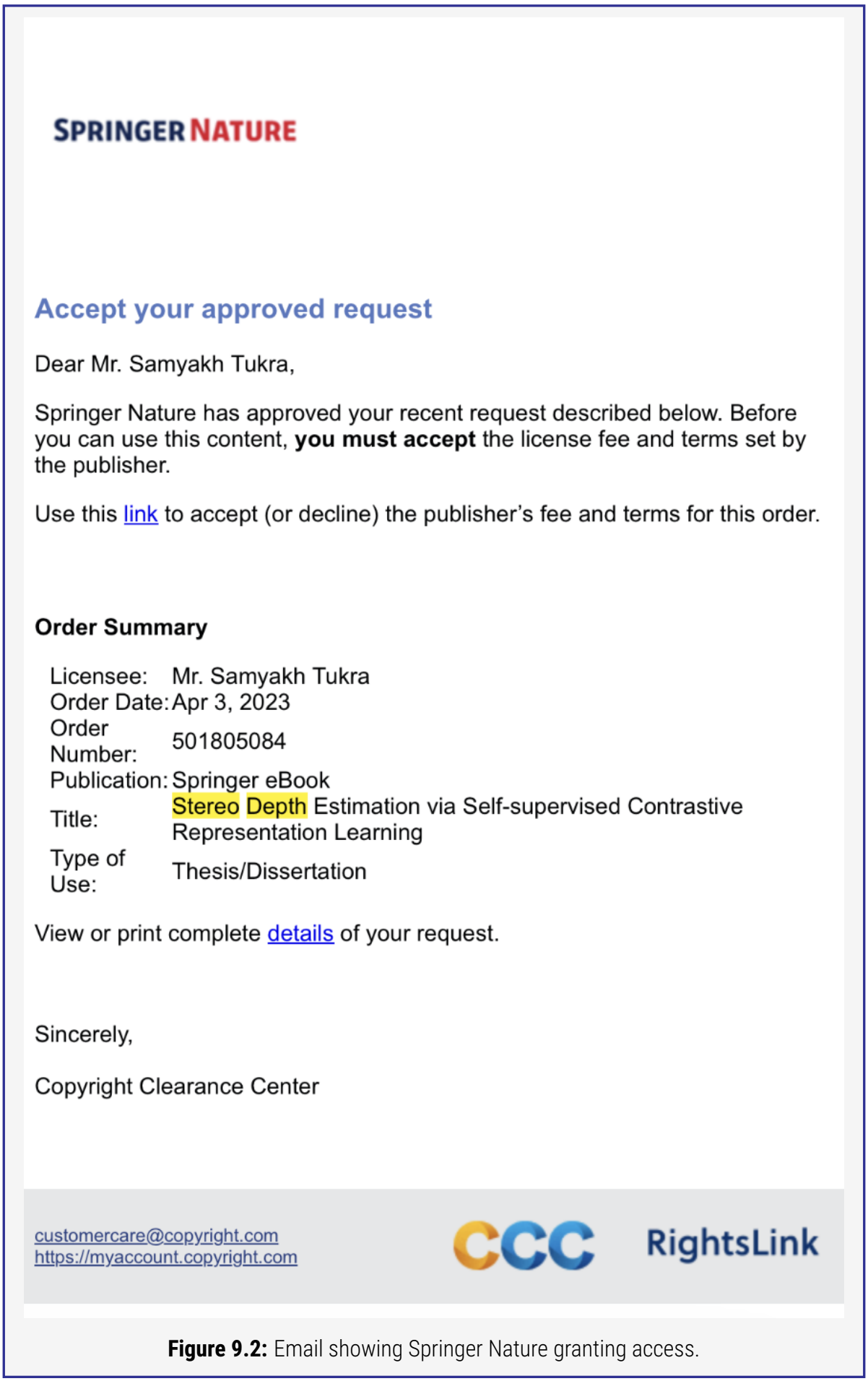
Figure 9.1: Email showing Taylor & Francis granting access.

9.4.3 Springer Nature

License for reuse of the following paper (presented in chapter 5) was granted:

Tukra, S., Giannarou, S. (2022). Stereo Depth Estimation via Self-supervised Contrastive Representation Learning. In: Wang, L., Dou, Q., Fletcher, P.T., Speidel, S., Li, S. (eds) Medical Image Computing and Computer Assisted Intervention – MICCAI 2022. MICCAI 2022. Lecture Notes in Computer Science, vol 13437. Springer, Cham.

Fig 9.2 shows the license granted, where the following rules apply: **"**The Licensor grants you a personal, non-exclusive, non-transferable, non-sublicensable, revocable, world-wide License to reproduce, distribute, communicate to the public, make available, broadcast, electronically transmit or create derivative works using the Licensed Material for the purpose(s) specified in your RightsLink Licence Details only. Licenses are granted for the specific use requested in the order and for no other use, subject to these Terms and Conditions. You acknowledge and agree that the rights granted to you under this License do not include the right to modify, edit, translate, include in collective works, or create derivative works of the Licensed Material in whole or in part unless expressly stated in your RightsLink Licence Details. You may use the Licensed Material only as permitted under this Agreement and will not reproduce, distribute, display, perform, or otherwise use or exploit any Licensed Material in any way, in whole or in part, except as expressly permitted by this License.**"**



SPRINGER NATURE

Accept your approved request

Dear Mr. Samyakh Tukra,

Springer Nature has approved your recent request described below. Before you can use this content, **you must accept** the license fee and terms set by the publisher.

Use this link to accept (or decline) the publisher's fee and terms for this order.

Order Summary

Licensee:	Mr. Samyakh Tukra
Order Date:	Apr 3, 2023
Order Number:	501805084
Publication:	Springer eBook
Title:	Stereo Depth Estimation via Self-supervised Contrastive Representation Learning
Type of Use:	Thesis/Dissertation

View or print complete details of your request.

Sincerely,

Copyright Clearance Center

customercare@copyright.com
https://myaccount.copyright.com

CCC RightsLink

Figure 9.2: Email showing Springer Nature granting access.