


Article

A 3DCNN-LSTM Multi-Class Temporal Segmentation for Hand Gesture Recognition

Letizia Gionfrida ^{1,2,*} , Wan M. R. Rusli ¹, Angela E. Kedgley ¹ and Anil A. Bharath ¹¹ Department of Bioengineering, Imperial College London, London SW7 2AZ, UK² John A. Paulson School of Engineering and Applied Sciences, Harvard University, Boston, MA 02134, USA

* Correspondence: gionfrida@seas.harvard.edu

Abstract: This paper introduces a multi-class hand gesture recognition model developed to identify a set of hand gesture sequences from two-dimensional RGB video recordings, using both the appearance and spatiotemporal parameters of consecutive frames. The classifier utilizes a convolutional-based network combined with a long-short-term memory unit. To leverage the need for a large-scale dataset, the model deploys training on a public dataset, adopting a technique known as transfer learning to fine-tune the architecture on the hand gestures of relevance. Validation curves performed over a batch size of 64 indicate an accuracy of 93.95% (± 0.37) with a mean Jaccard index of 0.812 (± 0.105) for 22 participants. The fine-tuned architecture illustrates the possibility of refining a model with a small set of data (113,410 fully labelled image frames) to cover previously unknown hand gestures. The main contribution of this work includes a custom hand gesture recognition network driven by monocular RGB video sequences that outperform previous temporal segmentation models, embracing a small-sized architecture that facilitates wide adoption.

Keywords: hand gesture classification; transfer learning; three-dimensional convolutional; LSTM network



Citation: Gionfrida, L.; Rusli, W.M.R.; Kedgley, A.E.; Bharath, A.A. A 3DCNN-LSTM Multi-Class Temporal Segmentation for Hand Gesture Recognition. *Electronics* **2022**, *11*, 2427. <https://doi.org/10.3390/electronics11152427>

Received: 21 June 2022

Accepted: 31 July 2022

Published: 4 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Hand gestures are a critically important form of non-verbal communication. The interpretation of hand gestures using wearable sensors [1,2], or cameras [3,4] aims to transform the movement of the hand into meaningful instructions; this interaction is also known as hand gesture recognition. The field of hand gesture recognition has seen significant improvements over the past few years [5] and, most recently, bundled with the latest advancements in computer vision, has encouraged the development of new technologies to support rehabilitation [6,7], robot control, and home automation [8]. Amongst other techniques, deep learning and computer vision methods have aimed to reach a complex understanding of the dynamic behaviours of hand motion, with the advantage of being more sensitive to learning rapid time-varying features.

Computer vision techniques rely on convolutional neural networks (CNNs) to extract two-dimensional (appearance-based) and three-dimensional (motion-based) array features. CNNs are generally used in image recognition to process pixel data. They take raw pixel data as input, train the designed architecture, and automatically extract features. These models have been divided into static (two-dimensional) and dynamic (three-dimensional) based on the model's output features. Several investigations [9–11] have implemented two-dimensional static appearance-based hand gesture recognition models (also known as two-dimensional CNN models), intending to develop a computationally inexpensive classifier to extract stable shapes of the human hand. However, these models do not consider the spatio-temporal parameters that occur from sequential frames of a video recording, and appearance alone cannot accurately identify the gesture signature [12]. Therefore, new approaches, known as three-dimensional dynamic hand gesture recognition, have emerged to fill this gap.

Three-dimensional dynamic hand gesture recognition models also rely on CNNs, act similarly to conventional two-dimensional CNNs, and have spatial-temporal filters. Since their introduction in 2015 [13], these models have been primarily embraced for hand gesture recognition [13–15], presenting excellent characteristics in recognizing hand actions from both appearance and spatio-temporal features. However, they require more parameters than two-dimensional CNNs, meaning vast datasets are needed, and making them more challenging to train [16]. Furthermore, these approaches have additional drawbacks that include cost, the logistical challenges of dealing with complex and lengthy datasets, and the requisite quality of captured images needed for appropriate training. To overcome these drawbacks, previous research has leveraged a technique known as transfer learning [17].

Transfer learning is a methodology where architecture is implemented and trained on a generic activity and is then adopted for a specific different but linked activity (Figure 1). This technique is often employed to tackle the issue of a deficiency of training data [18,19]. The usual objective of transfer learning techniques is to learn visual features from the initial assignment [19]. This technique can train and acquire a forthcoming linked task from fewer data samples. Transfer learning is adopted when a novel, minor dataset is smaller than the dataset used to train the pre-trained architecture.

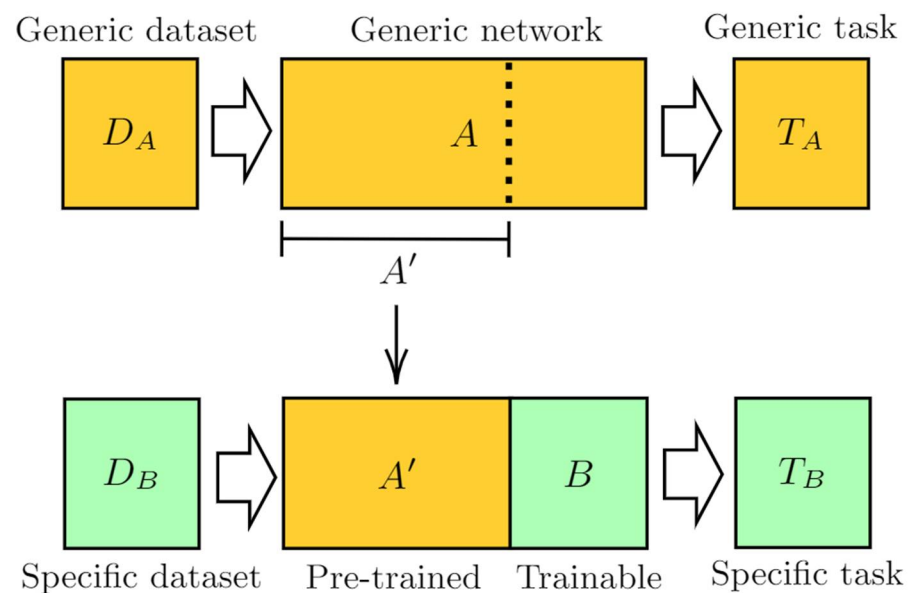


Figure 1. Schematic of the approach to transfer learning, whereby architecture implemented and trained on one activity is adapted for a different but linked activity.

Another hurdle in dynamic gesture recognition for three-dimensional CNNs is recognizing specific actions when dealing with continuous video streams [20]. Identifying human activities within video sequences is difficult because of the vast irregularity of hand actions on a time scale, unclear frame quantity, distribution, and limits of gesture signatures [21–23]. Furthermore, hand motions are often intricate and articulated and, when performed in an uncontrolled environment, can lead to occlusion that can limit the tracking. However, the ability to track and segment hand gestures in the real world can answer the need for application of these models to more realistic and generalizable tasks.

Manual segmentation of continuous video recordings is considered the most adopted technique when training hand gesture recognition [24]. However, the process is lengthy, and often a large proportion of frames is left unlabeled, causing indexing issues in the training of novel classification methods. The ability to automatically detect action in video recordings has an essential function for different applications that require end-to-end process automation. However, while much work has been produced on increasing the accuracy of hand gesture recognition models and enhancing the strength of these approaches [3,5,25], just a few attempts have been presented for temporal segmentation [26,27].

Attempts at temporal segmentation have focused on motion trajectory [28] and skeletal tracking [29] from depth cameras. However, these systems were sensitive to image backgrounds and lighting conditions. A different approach, presented by Camgoz et al., suggested windowing the continuous video stream for segmentation [30]. However, the length of the sliding volume was fixed, often cutting part of the critical features of the gestures. Moreover, appearance and hand motion information complement a temporal segmentation classifier [27]. However, Camgoz et al. also used only time-series data detected from hand motion, with no appearance information [30]. Kuehne et al. [31] proposed an end-to-end generative framework for video segmentation, using hidden Markov modelling for video segmentation and recognition of human activities. This has the drawback of an intensive processing time, reducing the ease of applying the approach in real-time. Ni et al. [32] presented an approach based on recurrent neural networks (RNNs) to perform sliding window detection and temporally segmenting continuous actions. The issue with this methodology is linked to the identification of peripheral boundaries only, with no global overview of the temporal events.

To overcome these disadvantages, recent approaches have suggested making a distinction between gestural frames, when the action is taking place, and translation frames, by merging both shape and spatiotemporal parameters. Such an approach has been presented by Wang [27]. Wang presented a segmentation method that contained both action and appearance-based information and used both RGB and depth capture modalities driven by dual architecture for hand gesture classification and segmentation. This approach requires dual-modality acquisition, which does not leverage the ubiquity of standard monocular RGB cameras. Similarly, most recently, Sahoo et al. [33] presented an end-to-end fine-tuning method using a pre-trained CNN for a hand gesture recognition model; however, their model was also driven by dual-modality and multiple architectures.

Increasingly, enormous datasets of human movement are publicly available, as researchers seek to pool resources and work more openly. The 20BN Jester is a state-of-the-art dataset and the largest of the human hand gestures collected from monocular RGB cameras. It contains a total of 148,092 videos corresponding to 5,331,312 frames [15]. Each video is, on average, three seconds, and the dataset contains a total of 27 classes.

This paper aims to present a novel pipeline based on the training of a CNN using a small set of data for the development of a narrow architecture that can run efficiently during continuous video recordings of hand gestures to effectively recognize different gesture interactions. The key contributions of this paper include:

- (a) The implementation and testing of a novel pipeline that leverages a three-dimensional CNN model combined with a long short-term memory (LSTM) unit to reliably classify and temporally segment continuous video recordings. This novel pipeline enables improved accuracy compared with previously presented methodologies.
- (b) The introduction of a model trained on a larger scale dataset and then fine-tuned on a small-scale dataset, that enables generalizability to different types of gestures, participants and hand shapes.
- (c) The introduction of a small-scale architecture that lays the foundations for a real-time model capable of executing tasks reliably in real-world scenarios. This paves the way to a broader and optimised application that can be used to automatically detect tasks in different domains.

To deliver these contributions, we proceed as follows. In Section 2, the experimental set-up, data collection, and pre-and post-processing steps implemented for the action recognition detector are explained. Section 3 discusses the experimental results and Section 4 summarizes the main implications of these findings and addresses future directions. Section 5 concludes the proposed work. The key novelty of the presented methodology (with evaluation) includes a temporal segmentation classifier driven by monocular video sequences that outperform previous investigations in terms of accuracy and enables fine-tuning on a small-scale dataset trained on a single, low-complexity, architecture.

2. Materials and Methods

2.1. Experimental Set-Up

Twenty-two volunteers (twelve female, ten male) participated in this experiment. All the participants were healthy, presenting with no hand pathology, no loss in mobility, and no experience of upper limb joint surgery or fracture in the six months preceding the data collection. All participants were informed, both verbally and in writing, of their right to withdraw from the study at any time. Written informed consent was obtained from each participant. The study was conducted according to the guidelines of the Declaration of Helsinki and approved by the Imperial College Research Ethics Committee (ICREC) of Imperial College London. Video data were captured using an Oqus RGB camera (Qualisys AB, Göteborg, Sweden) at a 30 Hz frame rate. The entire pipeline adopted in the study is illustrated in Figure 2.

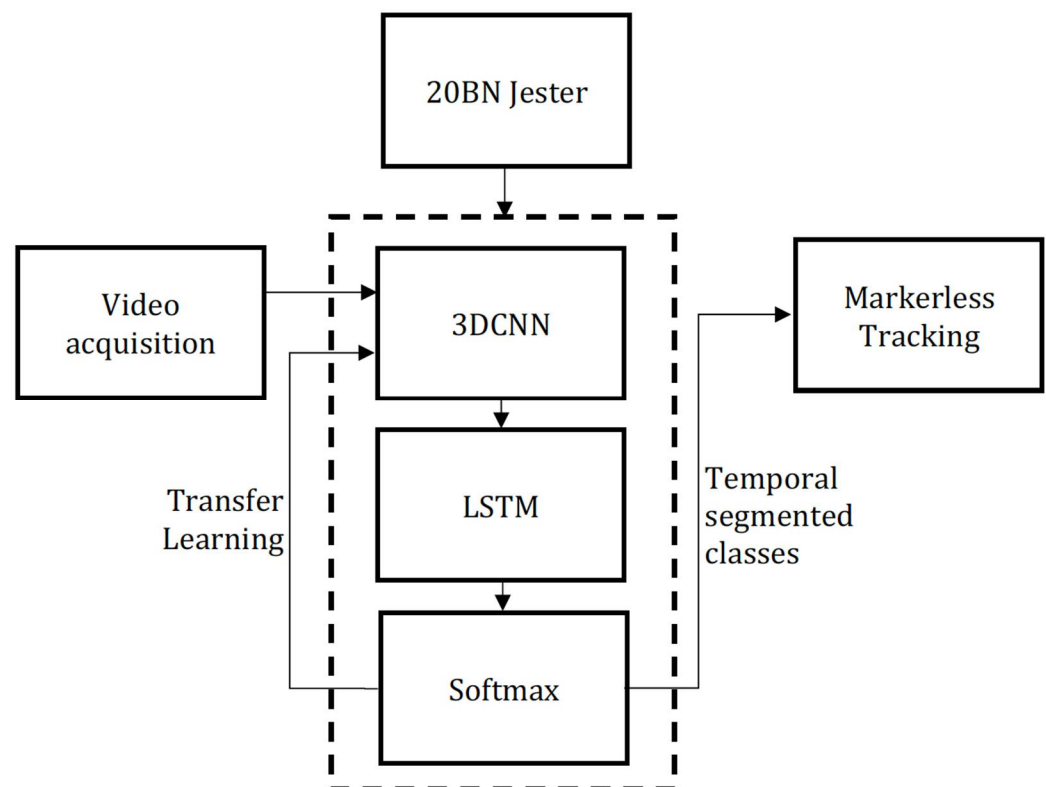


Figure 2. Flowchart of the experimental setup for the hand gesture recognition investigation. The pipeline uses transfer learning, pre-training the architecture on the 20BN Jester dataset [15], a three-dimensional convolutional neural network (3DCNN), a long short-term memory (LSTM) unit and the output function (Softmax).

2.2. Data Collection

Ground truth composition is an essential matter in CNN-based design. Given the absence of an available hand gestures dataset suitable for clinical hand applications, a novel recorded hand gesture dataset was introduced. While we acknowledge that many gestures can be performed by one person, to generate the hand gesture dataset, we included more participants to increase the population diversity (e.g., hand shapes, skin color) and generalizability of the proposed methodology. To enable comparison with other proposed hand gesture models, the accuracy was initially tested for 12 participants. To increase the performance of the model, 10 more participants were subsequently added for a total of 22 participants.

Participants were asked to record one video sequence during online video meetings. A timed PowerPoint (Microsoft, Redmond, WA, USA) show was used to make the video acquisition consistent, to support participants in the activities to be performed during the

recordings, and to inform participants regarding the way to position themselves relative to the device for the recordings.

To perform the hand gestures, participants were asked to use a standard device camera to capture the required hand exercises using any laptop, smartphone, or desktop computer. A standard camera was defined as a camera developed from 2012 onwards that was able to capture video recordings at a rate of thirty frames per second. To assess if the data were captured from an acceptable browser and operating system, participants were asked to check the specifications of their recording system.

The hand activities performed by participants included abduction and adduction, metacarpophalangeal joint flexion, and thumb opposition. Each was performed four times with both the left and right hands. During these exercises, participants were asked to hold static poses for five seconds. Four classes of gestures were defined based on the trials (Figure 3).



Figure 3. Illustration showing hand gestures classified during each trial: no gesture, abduction and adduction (Abd and add), metacarpophalangeal (MCP) flexion and thumb opposition.

The hand gesture sequences were captured from continuous video recordings of 250 s. The continuous video sequences were then manually segmented and labelled. Examples representing the data collected from twelve representative participants are illustrated in Figure 4.



Figure 4. Examples of anonymized frames of the videos from twelve representative participants. The images show the variance in the subjects' appearance and background scenes.

In addition to the captured data, the 20BN Jester dataset acquired by Materzynska et al. [15] was used. The classes of interest in this study, “no gesture”, “abduction and adduction”, “MCP flexion”, and “thumb opposition”, were not present in the Jester dataset. Therefore, out of the 27 classes of the 20BN Jester dataset, five hand activities were considered. These hand tasks of the 20BN Jester were count to five, swiping down and left, thumb up, and thumb down. These activities were selected to include different image frames of isolated digits and the palm with all the digits for both the left and right hands.

2.3. Pre-Processing

The captured frames were normalized to ensure that each input to the three-dimensional CNN had the same distribution, and each class had the same number of frames. This was particularly important as, although the timing of the participants' actions was marked by the PowerPoint presentation, individuals could execute hand gestures at different speeds. Ideally, a three-dimensional CNN input should always be balanced, making the model converge faster. If the input frames were not normalized, the weights could have had different calibrations across features, making the cost function converge ineffectively.

The frame length was set to be equal for all the acquisitions for which the hand gestures were at the centre of the video [9,34]. Following the structure of the 20BN Jester dataset, normalization was applied to impose a fixed length, set to be 32 frames. If the number of frames was higher or lower, a down-sampling or a padding function was applied, respectively, to generate fixed-length videos. Given the S_n sequence of RGB frames, the L_S length of the sequence, and the L_F fixed length, the padding and down-sampling techniques were defined as:

$$S_n = \begin{cases} padding(S_n), & L_S < L_F \\ (S_n), & L_S = L_F \\ downsampling(S_n), & L_S > L_F \end{cases} \quad (1)$$

Following normalization, the images were resampled to be 64×64 pixels to expedite classification. The labels were assigned manually, and the videos were manually trimmed for input into the segmentation classifier. Finally, for training and validation, the datasets were split into training, validation, and testing sets, with a 70:20:10 ratio.

Of the data from the video collected, a total of 2812 short video sequences of healthy volunteers performing three different hand activities were used for testing and validation, including 1968 ($\approx 70\%$ of the dataset) were used for training and 845 ($\approx 30\%$ of the dataset) were used for validation and testing. Each short video sequence contained 32 frames, for 89,984 frames in total. A total of 5155 short video sequences were collected, of which 3609 ($\approx 70\%$ of the dataset) were used for training and 1546 ($\approx 30\%$ of the dataset) were used for validation and testing. Each short video sequence contained 32 frames for a total of 113,410 frames for training and 6784 for validation.

2.4. Model Design, Training and Evaluation

After the data pre-processing, the architecture was implemented based on an existing model originally introduced by Tran et al. [35], known as C3D. Specifically, a modified version of the C3D network, similar to the multimodal RGB-D-based network by Hakim et al. [12], was considered. Furthermore, to make sure that the three-dimensional CNN model was able to learn longer sequences, another unit, able to acquire long-term temporal features, was combined with the three-dimensional CNN, an LSTM unit. The final architecture (Figure 5) consisted of a three-dimensional CNN layer with three convolutional layers, a Rectified Linear Unit (ReLU) as activation function in the hidden layers used to avoid vanishing gradient, one LSTM layer, a flatten layer, a fully connected dense layer and an activation function, also known as the Softmax layer.

The multi-dimensional input tensors were flattened into a single dimension. A flattened layer is often employed in the presence of multi-dimensional output. This layer aims to produce a linear output that can be conveyed onto a dense layer. A dense layer (also called fully connected) joined every input neuron to every output neuron in the preceding layer. Finally, the Softmax function produced a vector that denoted the list of probability classes of possible results. Based on the output from the Softmax, the frames were then segmented into those where the activities occurred and those where there was no gesture. The class "no gesture" was provided in case no activity was performed, but also for frames without a hand, when participants placed the hand down following a performed activity.

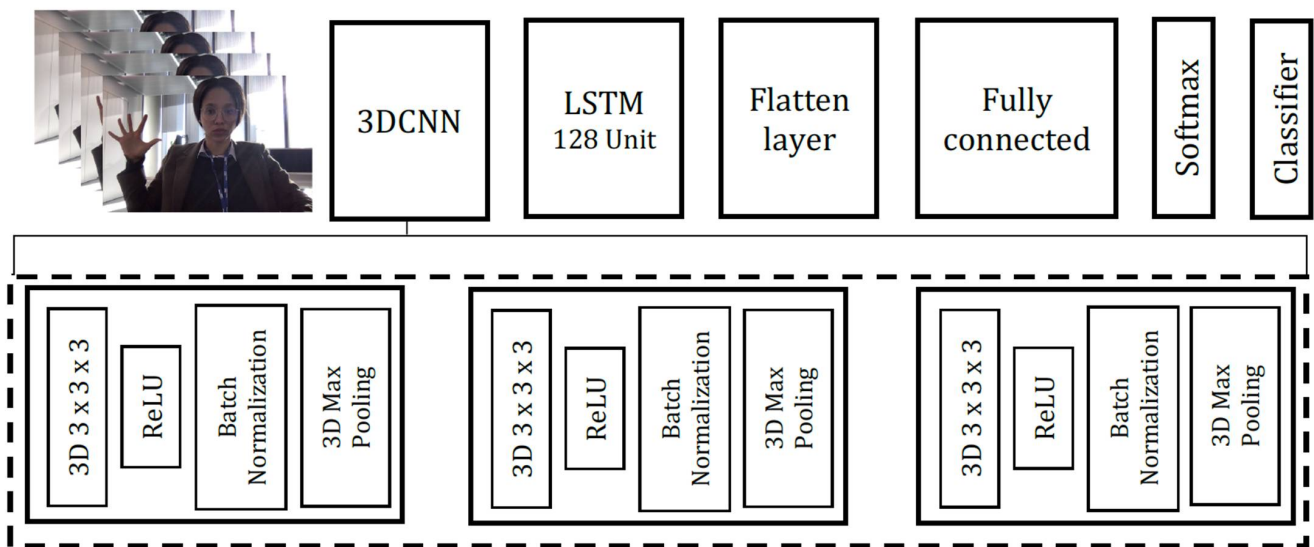


Figure 5. Three-dimensional convolutional neural network (3DCNN) with long short-term memory (LSTM) for dynamic hand gesture recognition. The video sequence is fed into the 3DCNN to operate 1D and 3D convolutions for time and space dimensions. The 3DCNN hidden layers (dashed box) with Rectified Linear Units (ReLU) as activation function have limitations to learning long-term information and therefore, the vector goes into an LSTM. The tensor in output is then flattened into a single dimension, passed into a fully connected layer, and finally, the activation function (Softmax) predicts the classes.

The baseline model was pre-trained on the selected five classes of the 20BN Jester dataset. Starting from the pre-trained architecture, a technique known as transfer-learning [18] was then used to fine-tune the model to the activities performed in this study. The technique took the parameters from the previously trained model, froze the last layers to avoid the weights in the last (frozen) layers being updated, and then new trainable layers were added, together with new data to fine-tune the model.

A total of four tests were performed. During the first two tests, transfer learning was used with three convolutional layers. Then, to increase performance, an additional convolutional layer and an increased sample size were considered. The first two tests were evaluated over mini-batches of 13 epochs, following the segmentation classifier proposed by Wang [27]. The last two tests were evaluated over a batch size of 64 epochs, a training batch size also presented in Wang's investigation [27]. A 12 gigabytes (GB) NVIDIA Tesla K80 graphics processing unit provided by Google Colaboratory was used for training the 20BN Jester dataset for the baseline model, TensorFlow [36] was used to deploy the model, and the training took approximately nine and a half hours. For the first and the second tests, the training times were, respectively, one and a half hours and two and a half hours, whereas for the last two tests, they were two and four hours.

The metric used to evaluate the performances of the model was the Jaccard index or intersection over union value [37,38]. The index is often used for segmentation classifiers and was computed to analogize a set of predicted labels with a set of the corresponding true labels. Letting A and B be the set of frames predicted and ground truth manually labelled, respectively, the index is defined as:

$$JACCARD = \frac{|A \cap B|}{|A \cup B|} \quad (2)$$

The Jaccard index varies from zero to one, the larger the index, the higher the accuracy of the temporal segmentation classifier. The mean Jaccard index recorded was used as a similarity to define performances of the proposed model with comparative studies. Training and validation accuracies were tested for 13 and 64 epochs for a small sub-portion of 12

participants and for 22 participants to evaluate how variations in population sizes can improve training and validation performances.

3. Results

Training and validation accuracies for 13 and 64 epochs for 12 and 22 participants show limited levels of accuracy (below 70%) for 13 epochs and an increased level of accuracy (93.95%) reached for 64 epochs (Figure 6). In the training and validation curves illustrated for 64 epochs, the training performed on 22 participants outperforms the training on 12 participants. Overfitting was observed during training after 50 epochs in both cases (12 and 22 participants), suggesting that additional training would not result in improved learning for the model.

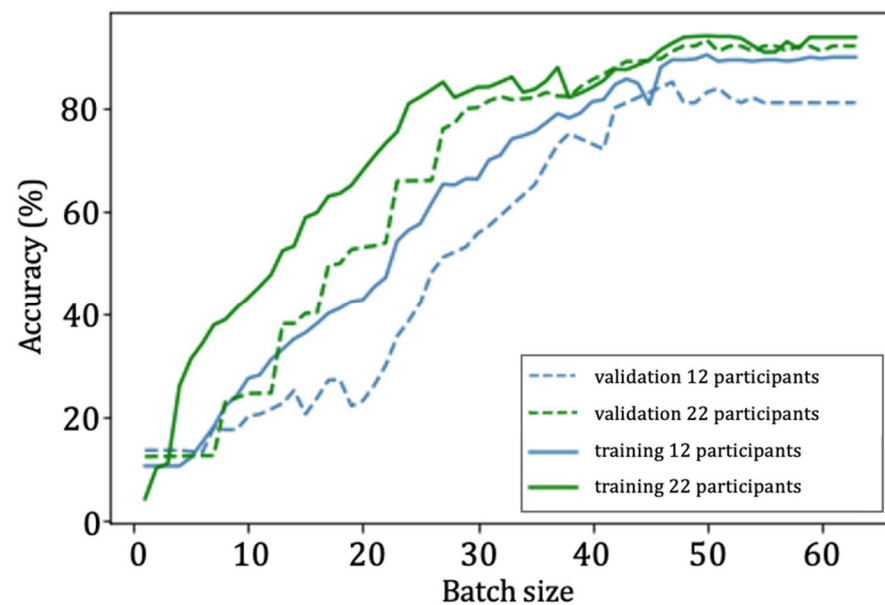


Figure 6. Results of the training (solid line) and validation (dotted line) for a training batch (batch size) of 64 epochs for 12 and 22 participants.

Representative output from the Softmax function (Figure 7) of the temporal segmentation for a continuous video recording for the three-dimensional CNN hand gesture classifier trained for 64 epochs and 22 participants illustrates the agreement with manual segmentation (ground truth).

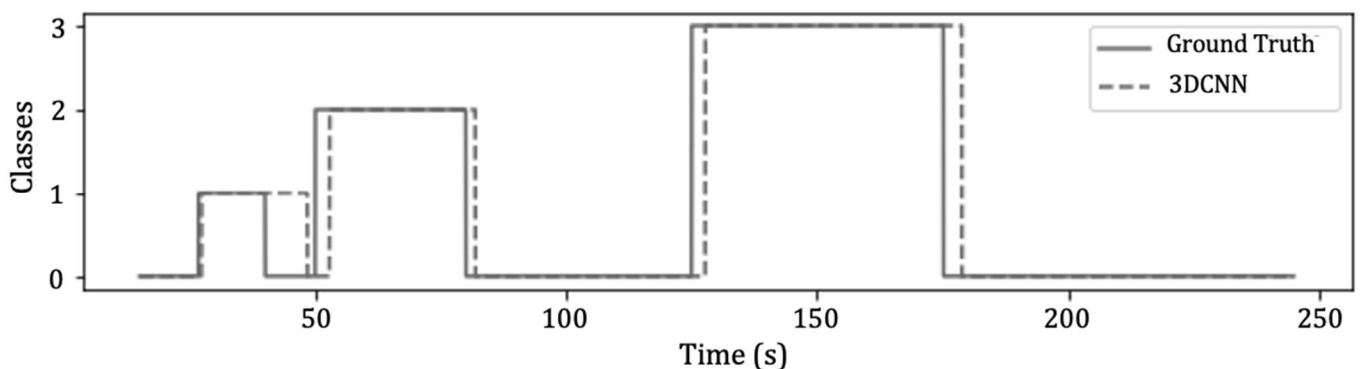


Figure 7. An example of the temporal segmentation and classification in output from Softmax function of the three-dimensional convolutional neural network for 64 epochs and 22 participants (dashed lines) compared against the ground truth manually segmented for Participant 1 for the labels “no gesture” (class = 0), “abduction and adduction” (class = 1), “metacarpophalangeal (MCP) flexion” (class = 2), and “thumb opposition” (class = 3).

The training runs, executed for batch-sized 64, had an initial mean Jaccard index that reached 0.794 (± 0.44), increasing to 0.812 (± 0.105) for the enlarged sample size of 22 participants (Table 1).

Table 1. Comparison of the three-dimensional convolutional neural network for 12 and 22 participants using the mean Jaccard index \bar{J}_s and the accuracy percentage (%).

Dataset	Number of Frames	Mean Jaccard Index \bar{J}_s	Accuracy (%)
12 participants	89,984	0.794	83%
22 participants	113,410	0.812	93.95%

The validation accuracy was 83% (± 0.05), increasing to an accuracy level of 93.95% (± 0.37) when additional participants were included. The “no gesture” label agreed with the manually segmented ground truth 96.47% of the time for all participants. The “abduction and adduction” class agreed with the ground truth 92.5% of the time for all participants. The “MCP flexion” label agreed with the manually obtained labels 95.7% of the time for all participants. Finally, the “thumb opposition” class agreed with the ground truth 90.93% of the time for all participants.

4. Discussion

This work illustrates a CNN that automatically classifies and segments videos containing specific hand exercises including no gesture, abduction and adduction, MCP flexion, and thumb opposition. The segmentation of continuous video recordings was based upon a classifier that identified when the label “no gesture” was present. The presented pipeline addressed the challenge of hand gesture recognition from long video sequences captured using a monocular RGB camera.

The implementation of the three-dimensional CNN was based on a model known as C3D, proposed by Tran et al. [35] and made of an high-resolution and a low-resolution sub-architecture, both trained individually. Even if the C3D model presented good performance, the cost of training two different models is high, so a modified version, which incorporated the two networks into one, was used in this work. This modified C3D, however, could only detect short temporal characteristics from short video sequences, whereas this work aimed to introduce a network that detects short-term temporal features from long video sequences. Therefore, the final CNN was combined with an LSTM unit, capable of learning the long-term dependencies in long video sequences.

Previous studies that combined three-dimensional CNN with LSTM units for hand activity recognition used both RGB and depth modalities to extract the motion signature [12,27], whereas the three-dimensional architecture implemented in this work was only based on an RGB sequence, showing a similar level of accuracy (93.95%) can be reached also from a single acquisition modality. Furthermore, the proposed network outperformed the 82% accuracy presented by Hakim et al. [12]. The overfitting observed after 64 epochs was similar to that of other investigations that used dual modalities [25,26]. The use of transfer learning to reach an acceptable (above 80%) level of accuracy enables the possibility of scaling this approach to include different hand gesture activities, showing how the model can be trained effectively on a small dataset to create an effective small-size segmentation classifier.

The mean Jaccard index recorded and used as a comparative index also in a similar investigation, was benchmarked against similar approaches. In Wang’s [27] and Wang et al.’s [39] studies, the Jaccard index was lower compared with the one presented in this investigation. The mean Jaccard index presented in this study reached 0.794 for the same number of participants, outperforming the value presented in previous investigations (Table 2). However, Wang’s accuracy was based on the Montalbano Gesture Dataset [39], containing different hand activities from those implemented in this investigation. Therefore, further investigations would be needed to compare the performances of this network

using this metric. Furthermore, no inconsistency was shown across the segmented video recordings for action and participants, meaning that segmentation accuracy was not based on specific actions or specific participants.

Table 2. Comparison of the proposed method and other methods for the mean Jaccard index \bar{J}_s in ascending order of accuracy based on 12 participants.

Methods	Mean Jaccard Index \bar{J}_s
Wang et al. [39]	0.2403
Chai et al. [29]	0.2655
Wang et al. [40]	0.5214
Wang [27]	0.6904
Our approach	0.794

To adopt and scale this application in real-world scenarios, if multiple classes are considered, future directions could include testing this approach for real-time application using a finite state machine system that can decrease the classes under inspection and increase the accuracy for real-time deployment. To further improve the model's performance for real-time applications, the input image size or the number of layers could be increased. On top of the 20BN Jester dataset, an additional dataset could be used to enhance the model's performance. The Jester dataset was developed by actors and did not provide numerous occlusion cases. Regardless, in realistic circumstances, occlusion exists. A foreseen limitation of the results reported here includes the absence of edge cases for the recordings captured in unconstrained scenarios. Ambiguous appearance results may lead to tracking errors. Capturing methods solely relying on two-dimensional appearance information could struggle in scenarios where images are blurry, out of the plane or rotated, distant or small. Visual tracking methods may be incorporated to consider types of interference (e.g., blurry hand gestures if the participants or the camera moves suddenly during the acquisition) with the goal of disambiguating the recognition target. Rescuing identifiable appearance cues of image interference for a real-time hand recognition model, for instance, with an image blur classification and blur removal, would be an attractive research direction.

Even given the limitations of the monocular technique, when incorporated into a pipeline that is intended for further processing, the temporal segmentation results are still usable when viewed in the context of performing manual temporal segmentation. One intended use case would be in a patient assessment setting, where hand exercises could be monitored, particularly when they are intended for use as therapy; this potentially extends suggested approaches of home exercise monitoring [41].

Furthermore, while the supervised-based transfer learning produced expected outcomes, the approach presented in this work could be transported to unsupervised learning and could support the automated labelling and segmentation of long video recordings, increasing the models' generalizability. Furthermore, hybrid deep learning models, such as the work from Nasser et al. [42], that combine recurrent networks to also model the temporal dependencies in high-dimensional sequences, which is an interesting area to explore further.

Adapting current gesture recognition techniques to specific mobility exercises would have benefits that go beyond this single application. A real-time device that requires minimal manual processing could process and identify multiple gestures as soon as an image frame is received. This approach could be deployed in online hand gesture recognition studies for advanced assistance systems, surveillance, aided robotics, and clinical applications. For instance, the pipeline illustrated here could be integrated into remote monitoring clinical solutions, presenting the training of a model that uses a smaller dataset implemented on a small architecture that can run efficiently to solve the classification problem for hand temporal segmentation. This would pave the way to a broader application of hand tracking models, incorporating other hand activity categories, and obtaining a more

generalizable approach, that would include different hand exercise programs and different hand conditions.

5. Conclusions

This work offers an approach for hand gesture segmentation from large-scale video sequences. The video sequences were first segmented into single hand gesture sequences by classifying the frames into different gestures. For one each of the segmented hand gesture series, the suggested technique utilized spatiotemporal information based on a three-dimensional convolutional neural network combined with a long short-term memory unit. To enhance the accuracy of the model, the training was performed on a large-scale hand dataset and fine-tuned for the relevant hand gestures. The introduced pipeline illustrated a model trained on a small-scale set of RGB image frames that presents increased accuracy (93.95%) compared with the previously presented techniques. Furthermore, the pipeline is performed on a small-sized architecture that enables real-time deployment and easier integration of further hand gesture classes using monocular cameras, leveraging ubiquitous technologies (e.g., in smartphones/laptops) and encouraging scalability for future investigations. Future investigations could investigate the performances of the model in real-time scenarios using small board and power-efficient devices. Moreover, visual tracking techniques could be explored to assess diverse types of interferences during real-time applications. Finally, the adoption of these models to support remote clinical monitoring could be evaluated further in future studies.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: L.G., A.A.B. and A.E.K.; data collection: L.G. and W.M.R.R.; data analysis and interpretation of results: L.G.; draft manuscript preparation: L.G., A.A.B. and A.E.K. All authors have read and agreed to the published version of the manuscript.

Funding: The dataset analysed during the current study was acquired through funding from the Wellcome Trust as part of the Wellcome Trust Institutional Translational Partnership Award 208858/Z/17/Z-Imperial Msk Accelerator at Imperial College London.

Institutional Review Board Statement: The study was conducted in accordance with the Declaration of Helsinki, and approved by the Institutional Review Board (or Ethics Committee) of Imperial College London (ICREC reference code 18IC4673) approved on 26 June 2018.

Informed Consent Statement: Written informed consent has been obtained from the participants to publish this paper.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors acknowledge Google Colaboratory from Google Research for providing the deep learning hardware (12 gigabytes NVIDIA Tesla K80 graphics processing unit).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Li, Y.; Di, H.; Xin, Y.; Jiang, X. Optical fiber data glove for hand posture capture. *Optik* **2021**, *233*, 166603. [[CrossRef](#)]
2. Dipietro, L.; Sabatini, A.M.; Dario, P. Evaluation of an instrumented glove for hand-movement acquisition. *J. Rehabil. Res. Dev.* **2003**, *40*, 179–189. [[CrossRef](#)] [[PubMed](#)]
3. Pinto, R.F.; Borges, C.D.; Almeida, A.; Paula, I.C. Static hand gesture recognition based on convolutional neural networks. *J. Electr. Comput. Eng.* **2019**, *2019*, 4167890. [[CrossRef](#)]
4. Wu, W.; Shi, M.; Wu, T.; Zhao, D.; Zhang, S.; Li, J. Real-time Hand Gesture Recognition Based on Deep Learning in Complex Environments. In Proceedings of the 2019 Chinese Control And Decision Conference (CCDC), Nanchang, China, 3–5 June 2019; pp. 5950–5955. [[CrossRef](#)]
5. Sonkusare, J.S.; Chopade, N.B.; Sor, R.; Tade, S.L. A Review on Hand Gesture Recognition System. In Proceedings of the 2015 International Conference on Computing Communication Control and Automation, Pune, India, 26–27 February 2015; pp. 790–794. [[CrossRef](#)]
6. Primya, T.; Kanagaraj, G.; Muthulakshmi, K.; Chitra, J.; Gowthami, A. Gesture recognition smart glove for speech impaired people. *Mater. Today Proc.* **2021**. [[CrossRef](#)]

7. Halim, Z.; Abbas, G. A Kinect-Based Sign Language Hand Gesture Recognition System for Hearing- and Speech-Impaired: A Pilot Study of Pakistani Sign Language. *Assist. Technol.* **2015**, *27*, 34–43. [[CrossRef](#)]
8. Metsis, V.; Jangyodsuk, P.; Athitsos, V.; Iversen, M.; Makedon, F. Computer aided rehabilitation for patients with rheumatoid arthritis. In Proceedings of the 2013 international conference on computing, networking and communications (ICNC), San Diego, CA, USA, 28–31 January 2013; pp. 97–102.
9. Adithya, V.; Rajesh, R. A deep convolutional neural network approach for static hand gesture recognition. *Procedia Comput. Sci.* **2020**, *171*, 2353–2361.
10. Flores, C.J.L.; Cutipa, A.G.; Enciso, R.L. Application of convolutional neural networks for static hand gestures recognition under different invariant features. In Proceedings of the 2017 IEEE XXIV International Conference on Electronics, Electrical Engineering and Computing (INTERCON), Cusco, Peru, 15–18 August 2017; pp. 1–4.
11. Oyedotun, O.K.; Khashman, A. Deep learning in vision-based static hand gesture recognition. *Neural Comput. Appl.* **2017**, *28*, 3941–3951. [[CrossRef](#)]
12. Hakim, N.L.; Shih, T.K.; Arachchi, S.P.K.; Aditya, W.; Chen, Y.-C.; Lin, C.-Y. Dynamic hand gesture recognition using 3DCNN and LSTM with FSM context-aware model. *Sensors* **2019**, *19*, 5429. [[CrossRef](#)]
13. Molchanov, P.; Gupta, S.; Kim, K.; Kautz, J. Hand gesture recognition with 3D convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Boston, MA, USA, 7–12 June 2015; pp. 1–7.
14. Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; Fei-Fei, L. Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1725–1732.
15. Materzynska, J.; Berger, G.; Bax, I.; Memisevic, R. The Jester Dataset: A Large-Scale Video Dataset of Human Gestures. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Korea, 28 October 2019; pp. 2874–2882. [[CrossRef](#)]
16. Carreira, J.; Zisserman, A. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 26 July 2017; pp. 4724–4733. [[CrossRef](#)]
17. Tan, C.; Sun, F.; Kong, T.; Zhang, W.; Yang, C.; Liu, C. A survey on deep transfer learning. In *International Conference on Artificial Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 270–279.
18. Tammina, S. Transfer learning using vgg-16 with deep convolutional neural network for classifying images. *Int. J. Sci. Res. Publ.* **2019**, *9*, 143–150. [[CrossRef](#)]
19. Shin, H.C.; Roth, H.R.; Gao, M.; Lu, L.; Xu, Z.; Nogues, I.; Yao, J.; Summers, R.M. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *Trans. Med. Imaging* **2016**, *35*, 1285–1298. [[CrossRef](#)]
20. Jiang, F.; Zhang, S.; Wu, S.; Gao, Y.; Zhao, D. Multi-layered gesture recognition with Kinect. *J. Mach. Learn. Res.* **2015**, *16*, 227–254.
21. Rodríguez-Moreno, I.; Martínez-Otzeta, J.M.; Sierra, B.; Rodríguez, I.; Jauregi, E. Video activity recognition: State-of-the-art. *Sensors* **2019**, *19*, 3160. [[CrossRef](#)] [[PubMed](#)]
22. Vrigkas, M.; Nikou, C.; Kakadiaris, I.A. A review of human activity recognition methods. *Front. Robot. AI* **2015**, *2*, 28. [[CrossRef](#)]
23. Mahmoud, R.; Belgacem, S.; Omri, M.N. Deep signature-based isolated and large scale continuous gesture recognition approach. *J. King Saud Univ.-Comput. Inf. Sci.* **2020**, *34*, 1793–1807. [[CrossRef](#)]
24. Panwar, M.; Mehra, P.S. Hand gesture recognition for human computer interaction. In Proceedings of the 2011 International Conference on Image Information Processing, Shimla, India, 3–5 November 2011; pp. 1–7. [[CrossRef](#)]
25. Al-Hammadi, M.; Muhammad, G.; Abdul, W.; Alsulaiman, M.; Hossain, M.S. Hand Gesture Recognition Using 3D-CNN Model. *Consum. Electron. Mag.* **2020**, *9*, 95–101. [[CrossRef](#)]
26. Zhu, G.; Zhang, L.; Shen, P.; Song, J.; Shah, S.A.A.; Bennamoun, M. Continuous gesture segmentation and recognition using 3DCNN and convolutional LSTM. *Trans. Multimed.* **2018**, *21*, 1011–1021. [[CrossRef](#)]
27. Wang, H. Two Stage Continuous Gesture Recognition Based on Deep Learning. *Electronics* **2021**, *10*, 534. [[CrossRef](#)]
28. Peng, X.; Wang, L.; Cai, Z.; Qiao, Y. Action and gesture temporal spotting with super vector representation. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 518–527.
29. Chai, X.; Liu, Z.; Yin, F.; Liu, Z.; Chen, X. Two streams recurrent neural networks for large-scale continuous gesture recognition. In Proceedings of the 2016 23rd International Conference on Pattern Recognition (ICPR), Cancún, Mexico, 4–8 December 2016; pp. 31–36.
30. Camgoz, N.C.; Hadfield, S.; Koller, O.; Bowden, R. Using Convolutional 3D Neural Networks for User-independent continuous gesture recognition. In Proceedings of the 2016 23rd International Conference on Pattern Recognition (ICPR), Cancún, Mexico, 4–8 December 2016; pp. 49–54. [[CrossRef](#)]
31. Kuehne, H.; Gall, J.; Serre, T. An end-to-end generative framework for video segmentation and recognition. In Proceedings of the 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Placid, NY, USA, 10 March 2016; pp. 1–8.
32. Ni, B.; Yang, X.; Gao, S. Progressively parsing interactional objects for fine grained action detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1020–1028.
33. Sahoo, P.; Prakash, A.J.; Pławiak, P.; Samantray, S. Real-Time Hand Gesture Recognition Using Fine-Tuned Convolutional Neural Network. *Sensors* **2022**, *22*, 706. [[CrossRef](#)]

34. Shanthakumar, V.A.; Peng, C.; Hansberger, J.; Cao, L.; Meacham, S.; Blakely, V. Design and evaluation of a hand gesture recognition approach for real-time interactions. *Multimed Tools Appl.* **2020**, *79*, 17707–17730. [[CrossRef](#)]
35. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning spatiotemporal features with 3d convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 4489–4497.
36. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Zheng, X.; et al. Tensorflow: A system for large-scale machine learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), Savannah, GA, USA, 2–4 November 2016; pp. 265–283.
37. Taha, A.A.; Hanbury, A. Metrics for evaluating 3D medical image segmentation: Analysis, selection, and tool. *BMC Med. Imaging* **2015**, *15*, 29. [[CrossRef](#)]
38. Escalera, S.; Athitsos, V.; Guyon, I. Challenges in multi-modal gesture recognition. *Gesture Recognit.* **2017**, 1–60.
39. Wang, P.; Li, W.; Liu, S.; Zhang, Y.; Gao, Z.; Ogunbona, P. Large-scale continuous gesture recognition using convolutional neural networks. In Proceedings of the 2016 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 4–8 December 2016; pp. 13–18.
40. Wang, H.; Wang, P.; Song, Z.; Li, W. Large-scale multimodal gesture segmentation and recognition based on convolutional neural networks. In Proceedings of the IEEE International Conference on Computer Vision Workshops 2017, Venice, Italy, 22–29 October 2017; pp. 3138–3146.
41. Veiga, C.; Pedras, S.; Oliveira, R.; Paredes, H.; Silva, I. A Systematic Review on Smartphone Use for Activity Monitoring During Exercise Therapy in Intermittent Claudication. *J. Vasc. Surg.* **2022**. [[CrossRef](#)] [[PubMed](#)]
42. Nasser, A.R.; Hasan, A.M.; Humaidi, A.J.; Alkhayyat, A.; Alzubaidi, L.; Fadhel, M.A.; Santamaria, J.; Duan, Y. IoT and cloud computing in health-care: A new wearable device and cloud-based deep learning algorithm for monitoring of diabetes. *Electronics* **2021**, *10*, 2719. [[CrossRef](#)]