Supplementary information: Machine learning to support visual auditing of home-based lateral flow immunoassay self-test results for SARS-CoV-2 antibodies

Supplementary Information 4 – Use of classical algorithms to interpret LFIA reading

Figure S4-1: Colour spaces and projection signatures. The projection signatures are generated by averaging (mean) the images pixel values across the short axis, which produces a signal in the long axis. A: Red, green, and blue colour channels, and their projection signatures, $\mathbf{P_{(R|G|B)}(x_\theta)}$. B: Normalised red, green, and blue colour channels, and the signatures $\mathbf{P_{(nR|nG|nB)}(x_\theta)}$. C: Hue, saturation, and value (intensity) channels, and signatures $\mathbf{P_{(H|S|V)}(x_\theta)}$. D: The O opponency channel, and its projection signature, $\mathbf{P_O(x_\theta)}$.

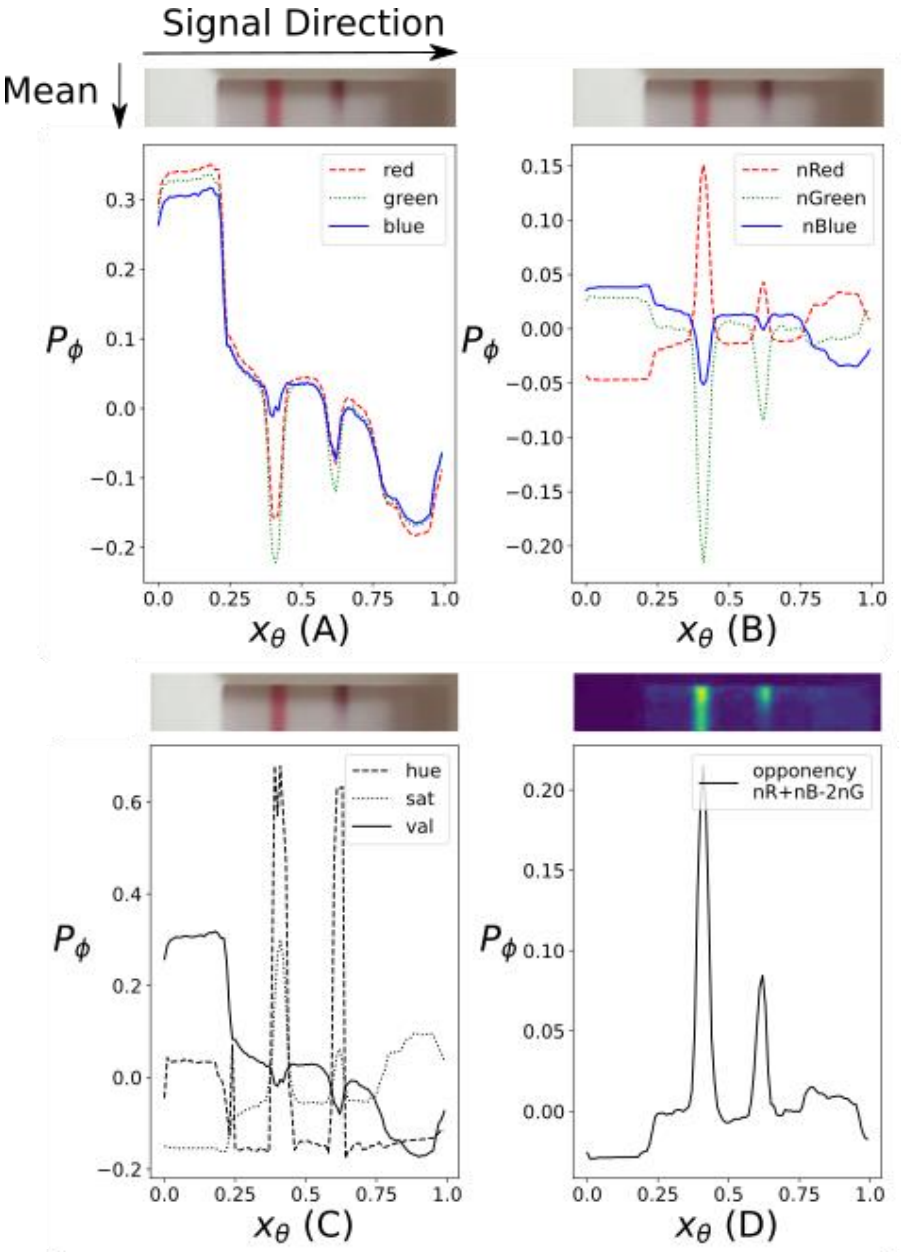


Figure S4-2: Normalised RGB projection signatures for an invalid and valid LFIA}. The figure shows examples of invalid (left) and valid (right) LFIA test result windows and their respective nRGB projection signatures. We developed a simple algorithm that takes these signatures as input to determine the validity of the LFIA. The signatures also form the basis of semi-quantitative analyses presented later in this report.

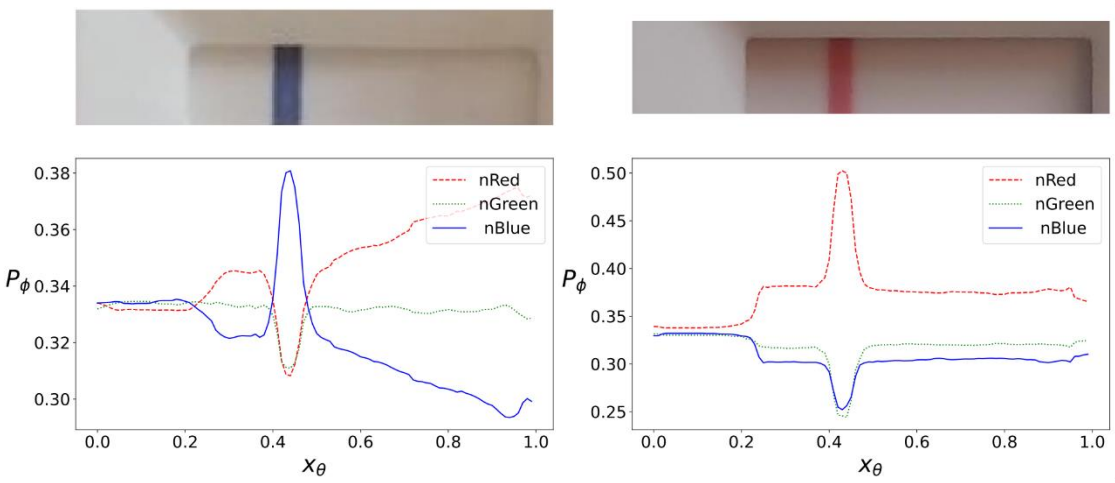


Figure S4-3: Opponency and edge-intensity projection signatures for seropositive and seronegative LFIAs}. The figure shows examples of seropositive (left) and seronegative (right) LFIA test result windows and their respective opponency and edge-intensity projection signatures. The algorithm labelled Heuristic-C in Tables S4-1 and S4-2 takes these signatures as input to determine the IgG result of the LFIA.

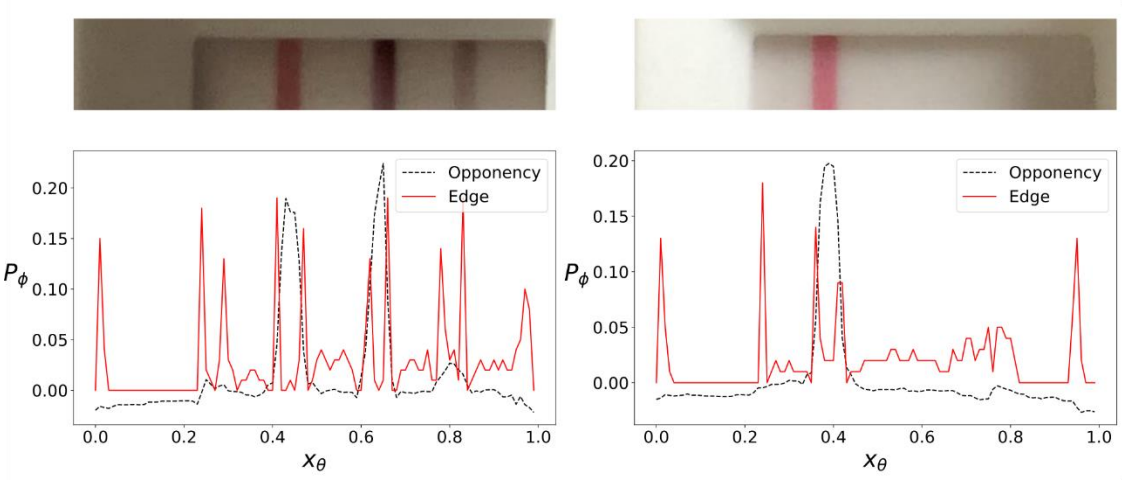


Table S4-1 Classification Experiment 1: A comparison of the performance of the projection-based peak detection algorithm described in this Supplementary material. 1D CNNs make use of the projection colour space we describe in Section E2, whilst "Heuristic C" represent the best parameter selection for peak detection. Bear in mind that all of these techniques for pre-processing of the data can be viewed as a combination of i) dimensionality reduction and ii) KL projection, effectively reflecting either geometric or statistical algorithms. Whilst these algorithms work very well for determining the status of the control line (see Section S4-B.3 and S4-B.4), the

same principle displays relatively poor sensitivity, and this is due to the presence of weak positives. In contrast, the 2D CNN model displays the best combination of specificity and sensitivity.

Model/Heuristic/Participants	Specificity	Sensitivity	Accuracy	Cohen's Kappa
2D CNN	0.994	0.971	0.983	0.966
1D CNN Model 1	0.995	0.831	0.917	0.832
1D CNN Model 2	0.968	0.853	0.913	0.824
1D CNN Model 3	0.999	0.879	0.941	0.882
1D CNN Model 4	0.988	0.833	0.914	0.827
1D CNN Model 5	0.990	0.900	0.946	0.892
Heuristic-C	0.994	0.757	0.881	0.759
Study Participants	0.961	1	0.980	0.959

Table S4-2 Classification Experiments when substantial weak positives are included This provides a comparison between carefully tuned peak-detection algorithm and CNN performance for data containing a significant proportion of weak positive samples. Thus, the notion of sensitivity and specificity, though probabilistically well defined, are only reliable when the dataset correctly reflect the background priors of weak cases amongst positive cases. This is a critical point in representing the sensitivity of detection, and is the key reason that the more sophisticated 2D CNN is required.

Model/Heuristic/Participants	Specificity	Sensitivity	Accuracy	Cohen's Kappa
2D CNN (CE1)	1.000	0.852	0.949	0.883
2D CNN (CE2, retrained)	0.987	0.901	0.958	0.905
1D CNN Model 1	0.969	0.825	0.920	0.701
1D CNN Model 2	0.965	0.810	0.912	0.667
1D CNN Model 3	0.968	0.844	0.926	0.733
1D CNN Model 4	0.941	0.854	0.911	0.678
1D CNN Model 5	0.960	0.852	0.923	0.726
Heuristic-C	0.976	0.487	0.815	0.525
Study Participants	0.974	0.679	0.873	0.699

S4-A Classical algorithms

The quasi-linear structure of the LFIA window does indeed suggest that the application of relatively simple operations e.g. linear projections – similar to principal components analysis – to solve the problem of detecting the location and status of control lines. Indeed, we use standard algorithms of a mature Python library (scipy-signal) to detect these peaks. However, we also make use of a colour transformation that uses non-orthogonal bases of colour space that are *very* similar to a Karhunen-Loeve transform on 3D colour space to increase the detectability of the control line using these algorithms. Whilst these are quite adequate for the *control* line (with a small number of exceptions), straightforward 1D statistical peak detection is inappropriate for the IgG status line due to two key factors:

i) The presence of weak positives, which reduce contrast to below the typical noise threshold using linear projections.

ii) The presence of blood leakage into the window which – though providing high degrees of specificity, does so with reduced sensitivity, bringing the

Indeed, we apply such a technique to peak detection for the **control line status**, and this works quite robustly.

In the following sections of this Appendix to our response to Reviewers' Comments, we present the evidence that – although a simpler approach to status detection seems feasible – it does not match the individual precision and sensitivity that is attained by the 2D CNN.

The reason for this is relatively intuitive (in hindsight!): although the ideal image sample contains blood only in the well, either leakage or "spatter" will sometimes lead to blood that falls into the readout window (this is a more common occurrence than one would anticipate). Damaged devices can also display significant departures from the expected appearance. The presence of such cases is not always discernible in 1D projection space from a positive IgG result: it is best captured from the segmented window of the original 2D image.

S4-A Read-out Algorithms (interpreting the result, post segmentation)

Having substantially reduced the size and complexity of the image data through the pre-processing steps, we are in a position to focus on the read-out window. Due to mapping the pixel data of the read-out window into a reference axis (x_θ, y_θ), we note that the image has a strongly linear structure. The alignment of this linear structure with the representation axis (i.e. row-column organisation of the image) means that we can apply low-complexity algorithms to perform several of the steps needed for interpreting read-out, without recourse to data-driven training.

The read-out window of the Fortress device provides a control line that switches colour from blue to red when the test has been correctly performed. Thus, a key stage that gates subsequent interpretation is the presence of the control line and the reading of its colour.

S4-A.1 Colour Spaces

The raw RGB values of the pixel intensities corresponding to the read-out window can shift with changes in lighting, the colour calibration of the device and the presence of shadows. In practice, colour-space transformations to the raw pixel values can be a convenient way of obtaining at least partial invariance to nuisance sources of colour shift. We found it convenient to use three different colour transformations, depending on the need to be colour-selective or approximately colour-invariant.

We use ten colour channels across four colour spaces. These are (R, G, B) (original pixel values), (nR, nG, nB) (normalised colour spaces), where $nR = R/(R + G + B)$ etc, (H, S, V) and a single-channel opponent colour space, $O = nR - 2nG + nB$. Like the (H, S, V) colour space, opponent colour spaces correlate well with perceptual notions of colour. The O channel, in particular, is recognisable as proportional to one channel of an Ohta (Tkalcic and Tasic 2003) colour space, which is an

approximation to one channel of the Karhunen-Loeve (KL) decomposition of colours for natural imagery. The Ohta space is known to be a good choice for colour image segmentation (Kartikeyan, Sarkar, and Majumder 1998) and visual descriptors (Payne and Singh 2005).

S4-A.2 Projection Signatures

Rather than using principal components analysis in colour space, we simply project (sum, or take an average) of the intensity data along the y_θ direction, yielding a function of x_θ. These projection signatures, $P_\Phi(x_\theta)$ are produced for each channel of each of three selected colour spaces. Specifically, for any colour-spatial field in the aligned coordinate system $f_\Phi(x_\theta, y_\theta)$, for $\Phi \in \{R, G, B, nR, nG, nB, H, S, V, O\}$ we approximate the 2D to 1D projection operator:

$$P_\Phi(x_\theta) = K \int f_\Phi(x_\theta, y_\theta) dy_\theta \qquad \text{Eq. S4} - 1$$

where K is a normalising constant that compensates for different zoom factors. In practice, the integral is simply approximated by taking an average over the rows (or columns) of the two-dimensional image array representing the normalised image window corresponding to the read-out.

Illustrations of the projection signatures are provided in Figure S4-1. Due to the white background of the test read-out window, the presence of red or blue lines is somewhat counterintuitive in the native (R, G, B) space, manifesting as dips in intensity on the primary colour channels. The normalised channels provide a slightly more obvious depiction, and we can observe the differences in appearance between the control lines for these cases (see also Figure S4-2).

S4-A.3 Detection of the Control Line

The first step to interpret the LFIA result is to locate the control line. The pipeline uses the opponency signature (Eq S4-1) and runs a peak detection heuristic algorithm (*scipy-signal* module, find_peaks) to find a peak corresponding to the control line. Because of the rescaling operation to a reference coordinate system (using segmentation and image rescaling), performing peak detection on this approach works to a satisfactory level.

This algorithm takes 5 parameters of height, width, relative height, distance, and prominence. The optimal settings for these five parameters was set by a grid search across three hyper-parameters $\alpha_h \in [0.9, 1.1]$, $\alpha_w \in [0.1, 0.5]$ and $\alpha_p \in [0.005, 0.016]$, then setting the *height* parameter of find_peaks() to $\mu \times \alpha_h$, *prominence* to $\mu \times \alpha_p$ the *width* parameter to $100 \times \alpha_w$; μ is the average amplitude of the projection signal. The parameter *relative_height* was fixed at 0.5, and *distance* was fixed at 100×0.1.

If this simple approach fails to locate the control line, then the LFIA result is deemed ‘unreadable’ and removed from the pipeline.

S4-A.4 Reading the control line status

A 2nd heuristic algorithm determines the validity by reviewing the normalised red (nR) and blue (nB) values at the control-peak location. For invalid tests, the nB value is higher than nR. The relationship is consistently reversed for valid tests, and Figure S4-2 illustrates this. Invalid LFIAs are removed

from the pipeline and reported as invalid test results, whilst the images deemed to be valid; move to the later stages of analysis.

In testing, this second heuristic algorithm on dataset 2, all examples are classified correctly for test *validity*. Though dataset 2 is small, and the results may not reflect the performance in the field, we can assume that the errors of Heuristic-B will be few. For future work, we can collect the data that participants have labelled invalid and create ground truth, a more trivial task than labelling the IgG status of the LFIA. Should this approach begin to fail significantly, a 1D or 2D CNN could be developed to classify validity.

S4-A.5 Seroprevalence Read-out: IgG Status

Initially, with the limited amount of labelled data, we developed an algorithm based on established peak-detection techniques to classify the IgG Status of samples. As the amount of data we labelled increased, more data-intensive methods replaced this approach, leading to the creation of first 1D and then 2D CNNs. The initial development and testing, Classification Experiment 1 (CE1), used Data set 3 The development set consists of samples from REACT-2 Study-4 and Study-5's R1. The test set came from Study-5's R2, removing any risk of data leakage; additionally, Study-5's R2 contained participant response data for comparison against both human experts and algorithm readings. For CE1, we did five-fold cross-validation on just the development set, ensuring that it was suitable for training; we then retrain on the whole development set and report the testing of the final model on the test set. Reported metrics are specificity, sensitivity, overall accuracy and Cohen's kappa.

Taking advantage, again, of the linear structure, we designed an initial data bootstrapping algorithm using the opponency signature and an edge intensity signature, which is only used for IgG status classification. The principles for detecting the control line are also applied; the heuristic looks for distinct peaks at identified locations. For a seropositive image, we require: a peak in opponency space, and to differentiate from possible blood leakage, two peaks in the edge-intensity space. The signatures are shown in Figure S4-3.

S4-B Using data-driven deep learning to finesse IgG status detection

The performance of the algorithms based on the colour space and geometric projection was excellent for strong (high visual contrast) IgG seroprevalence, but inadequate for weak IgG line. Weak IgG lines can be caused by a variety of factors, but the most likely cause is due to waning or weak immune response.

We found that the detection of weak lines using peak detection to be below that of expert readers, and in some cases, even of members of the public. Thus, we explored data driven machine learning with incrementing degrees of complexity: low parameter-count networks using 1D CNNs, then high-parameter count networks and transfer learning using 2D CNNs.

S4-B.1 1D CNN with projection signatures

A popular deep-learning method is the 2D CNN; however, as explained in Section S4-A, the LFIA read-out has a strong linear structure that facilitates collapsing the colour information into 1D

signatures. We, used ten signatures from different colour spaces, detailed in Section S4-A with examples in Figure S4-1, as inputs to our 1D CNNs. Five 1D network architectures were used, varying in the number of convolutional filters, the filter lengths and the number of layers. These variations enabled us to explore how architecture parameters might affect the performance. The 1D CNNs were implemented with PyTorch (Paszke et al. 2019), and training was completed over 100 epochs, using binary cross-entropy loss, and a learning rate of 0.0013. The development set was split into a 90:10 ratio for training: validation and batch sizes set to 2. Several random generator seeds were used, providing a mean and standard deviation (across seeds).

Alongside the networks, we developed custom data augmentation routines for the projection signatures. These included shifts in y_θ, scale changes and one-dimensional blur. These routines were selected as being appropriate augmentation routines for the one-dimensional signatures, reflecting the nature of the data transformations likely to be encountered.

S4-B.2 2D CNN with normalised read-out images

As more data was accumulated, we were able to leverage discrepancies between user-submitted, peak detection algorithms, then 1D-CNNs to the point where sufficient data was available to allow training the 2D CNN, implemented in Tensorflow, for read-out interpretation. This CNN utilised a MobileNetV2 (Sandler et al. 2018) architecture with pretrained weights learnt on ImageNet (Deng et al. 2009). The CNN was fine-tuned using the development set, over 100 epochs using a sparse categorical cross-entropy loss function, with a learning rate set to 0.001. Reported performance for this network is obtained from the test-set.

S4-B.3 Classification Experiment 1 Results

With regards to specificity, all methods perform better than the study participants. However, this is not the case for sensitivity where the participants are perfect, followed by the 2D CNN, 1D CNN models and finally, Heuristic-C. Looking at overall accuracy, the 2D CNN and participants perform similarly, both being better than the 1D CNNs and a heuristic approach based on colour-space projections and peak detection. For Cohen's Kappa with the expert, we can see that the 2D CNN has performed the best, with the value representing 'almost perfect agreement.' These results are promising as they initially imply that a 2D CNN, in general, could perform at a level on par with experts and hence auto-validation for LFIA could be possible. Although the 1D CNNs are not at the standard of the 2D CNN, the method still shows promise as it has high specificity while being nearly 27-50 times smaller than the 2D CNN, see Table S4-1.

S4-B.4 Classification Experiment 2 Results

Based on the apparent high-quality of these results, we implemented the methods into ALFA and deployed it to analyse the 'wild' data consisting of 500,000 images. A pattern emerged, many discrepancies were flagged where participants reported seropositive, but ALFA reported seronegative. Experts reviewed a sample of the discrepancies, discovering cases of 'weak IgG positives.' Detection of these cases is the objective of Classification Experiment 2 (CE2, shown in Table S4-2), which utilises samples from R5 of Study-5.

S4-C Cases of the Weak Positives

As REACT-2 Study-5 was ongoing throughout the pandemic, R5 was the first round where a small sample of participants had received their 1st vaccination dose. It was expected that the participants who received their first dose at least 21 days prior to completing the LFIA; should be seropositive; however, some reported seronegative tests. An expert review of these cases was completed, confirming the presence of weak IgG positive examples in addition to creating more training data. This new data set was used for CE2, where the networks were retrained and tested using the new data sets. Most training parameters remained the same from CE1 to CE2; however, the number of epochs increased from 100 to 150, and the training and validation batch sizes increase to 10 and 5, respectively.

Table S4-2 shows the results of CE2 and confirms the necessity of providing difficult cases in the training data. As the 2D CNN model trained in CE1 was the best performing method with a near-perfect agreement with an expert, we applied it to the test set of CE2. The results show that CE1's 2D CNN could not handle the weak positives; however, after retraining the model (CE2, retrained), we see improvement implying a cycle of iterative improvement is possible. This cycle would involve applying the best performing method to the 'wild' data, reviewing the discrepancies, identifying weak cases and retraining the network. Still, ground truth labelling is a labour-intensive task and requires a certain level of expertise; a solution could be to open-source the data set by scrubbing personal information and providing the labelling methods (e.g. Oxford's VGG Image Annotator).

Another vital message is that although the weak positives have caused the performance of the methods to decrease compared to expert reviews, we see that all methods, including Heuristic-C, outperform the study participants. This suggests that once implemented into ALFA, the algorithms can identify reporting mistakes.

References

M. Tkalcic and J. F. Tasic, "Colour spaces: perceptual, historical and applicational background," *The IEEE Region 8 EUROCON 2003. Computer as a Tool.*, 2003, pp. 304-308 vol.1, doi: 10.1109/EURCON.2003.1248032.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, *32*, 8026-8037.

Kartikeyan, B, A Sarkar, and KL Majumder. 1998. "A Segmentation Approach to Classification of Remote Sensing Imagery." *International Journal of Remote Sensing* 19 (9): 1695–1709.

Payne, Andrew, and Sameer Singh. 2005. "A Benchmark for Indoor/Outdoor Scene Classification." In *International Conference on Pattern Recognition and Image Analysis*, 711–18. Springer.

Sandler, M., A. Howard, M. Zhu, A. Zhmoginov, and L. Chen. 2018. "MobileNetV2: Inverted Residuals and Linear Bottlenecks." In 2018 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4510–20. https://doi.org/10.1109/CVPR.2018.00474.

Deng, J., W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. 2009. "ImageNet: A Large-Scale Hierarchical Image Database." In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–55. https://doi.org/10.1109/CVPR.2009.5206848.