Interpretation of American Sign Language Videos using Deep Learning based on a Computer Vision Approach.

Alejandro Lorenzo
alr@connect.ust.hk

Tara Relan
trelan@connect.ust.hk

Sandeepti Singh
ssinghaf@connect.ust.hk

The Hong Kong University of Science and Technology

Abstract

Sign languages are a language system that allows those hard of hearing to communicate. The extraction of complex head and hand movements along with their constantly changing shapes for recognition of sign language is considered a difficult problem in computer vision. This paper proposes the recognition of American Sign Language (ASL) video gestures using Convolutional Neural Networks (CNN) to extract high level features and Long Short-Term Memory (LSTM) to generate captions as well as a deeper network such as MobileNet v2. The database is trained from scratch using the 12,000 videos available from Kaggle's World Level American Sign Language data-set. Each sign occupies 40 frames in a video. Training was performed with 60% of the data-set, validating with 20% and testing with the remaining 20%. The final selected architecture has only 9 layers including dropout layers, which have increased the training accuracy to 100%, validation accuracy to 80%, and testing accuracy to 94%.

1. Introduction

Sign languages are a system of communication that use the visual-manual modality to convey meaning, instead of just spoken words. Wherever communities of deaf or hard-of-hearing (HOH) people exist, sign languages have developed as useful means of communication and form the core of local deaf cultures [11]. Although signing is used primarily by the deaf and hard-of-hearing, it is also used by hearing individuals, such as those unable to physically speak, those who have trouble with oral language due to a disability or condition, and those with deaf family members [8].

The problem we investigated in this project is sign language interpretation. A major issue with this form of communication is the lack of knowledge of this language, which creates a barrier between the hard of hearing and the hearing who do not understand sign language. Computer vision gesture recognition can help in the creation of a real-time sign language interpretation system that can resolve this communication barrier, as it can be used to create a form of real-time captioning for virtual conferences [13].

Even though sign language recognition using machines started years ago, due to the challenge of the problem, there is not an automatic transcription system. One problem is scalability as the number of gestures and variations in one's movement, facial expressions, and contextual meaning, is not trivial. Sign language has several symbols to represent different concepts. These symbols have their respective variations that change their meaning based on the context and what the person wishes to communicate.

To solve this classification task, we used algorithms used in sequential data, such as CNN and LSTM-type of networks. CNN is a popular neural network in image processing due to its success in recognising local features. These features are often used as an input to further models such as LSTMs. This way of interaction avoids the use of sensors and other peripherals, allowing a more natural way of communicating with a computer. The path followed led to a robust model which could correctly classify the videos into one label. Models discussed in previous literature reviews point out that the biggest parameters are hand space, location, palm orientation, body movement, and facial grammar. We also aimed to verify our model's performance and discuss possible changes that could be proposed to solve the problems faced. Given the broad scope of this task, this project limited the scope of the study to ASL words. Adding more gestures and features to the system and modifying the environment will generalise the problem, resulting in a more robust system.

1.1. Related Works

Gesture recognition is a competitive area of research. It often includes different approaches to recognise gestures such as the use of sensory hardware. Several devices are available for specific tasks or applications. One popular form of interaction is the area of natural interactions between humans and machines. By using sensors and capturing the diverse interactions of the user's body, it is therefore possible for machines to recognise given commands and perform the required tasks.

Research done in Sign Language Interpretation is mostly undertaken using a glove-based system. The performance of different glove-based and vision-based sign language detection was compared by Sruthi C. J. and Al. Lijiya [5] In this, sensors are attached to each finger. Based on their readings, the corresponding character is displayed. The main problem with this system is that it must be re-calibrated for each user, so their fingertips are captured [2]. Most gloves are made in similar sizes, which creates a problem for most users. Furthermore, due to frequent use, the glove may break, and wires can restrict the freedom of movements [2]. Overall, this glove-based system is not an effective solution.

This project will instead focus on video classification of different sign language characters. We have implemented a predictive model technology to automatically classify sign language symbols that can be used to create a form of real-time captioning. This is especially relevant as the COVID-19 global pandemic has led to more online conferences taking place over Zoom or Google Meets. Implementing a captioning system would then create a pathway where those who can hear and those who are deaf, or HOH can communicate.

Our current model has been inspired by 4 previous works. In 2017 there was a paper on human pose estimation via ResNet50 by X. Xiao and W.G. Wan [14]. In 2018, 2 papers were released, one on ASL recognition and computer vision (by K. Bantupalli and Y. Xie) [3], and the other on deep convolutional neural networks for sign language recognition (by G.A. Rao and K. Syamala) [10]. Finally, we based our model off a 2019 paper on deep learning and transfer learning approaches for image classification (by S.T. Krishna and H.K. Kalluri) [7].

As researchers prefer to use CNN for feature extraction from gesture data-sets, after investigating existing designs and studying the available alternatives, we were able to implement an algorithm that is based on a combination of image pre-processing and deep learning to achieve our objectives. Real-time prediction accuracies can be improved by trying transfer learning and CNN fine-tuning methods.

Our project differed from the previous literature in the sense that the papers we took inspiration from used CNN and ResNet50 to classify their images/videos. Our model goes further and uses bidirectional LSTM to help create captions which could then be used for live captioning.

2. Data

Kaggle has many data-sets, but we have decided to use the World Level American Sign Language (WLASL) as it has 12,000 videos of different people making signs (not just numbers) including men, women, and people of colour [4]. This is especially important since by training the model with a diverse group of people, there would be less bias, and therefore could be used effectively by different groups of people.

The Kaggle WLASL data-set consisted of 12,000 videos with 1,991 labels. The videos are in .mp4 RGB format with differing resolutions. After extracting 40 frames in each video, there were approximately 240 images or 6 videos per label.

After extracting the frames from each video, we then created the training variables. The images were resized to 77x48 to reduce processing power and increase processing speed. They were then converted to grey-scale and finally normalised. From there we converted the image into a tensor and extracted the training variables from that.

3. Methods

The model of the neural networks used in this paper can be described as a multi-layer perceptron. A block diagram of the system is shown in Fig 1. The description of each block is given below.

- Video database: the WLASL database contains 12,000 videos of different sign language gestures. These videos are taken from users of different genders, ages, and races to reduce bias. The resolution of each video may be varying.
- Video pre-processing: It is difficult to train videos as a whole and it may lead to poor performance. Therefore, we extracted 40 frames from each video, processed the images as mentioned in section 2 and used those images as input into the model.
- CNN training: Deep learning is used in this project. Hyperparameters are set accordingly before training the data-sets using any CNN architecture. The hy-

perparameters are batch size, number of epochs, and learning rate.
- LSTM training: the video frames are then fed into this model to generate labels. Hyperparameters are set accordingly as with the CNN model, and include sequence input size, batch size, dropout rate.
- Testing videos: 20% of the entire data-set of videos were used to test the architecture. To display the confusion matrix however, we used 16 videos to demonstrate.
- Display output: the predicted sign is displayed in text format alongside its actual sign on the video frame.

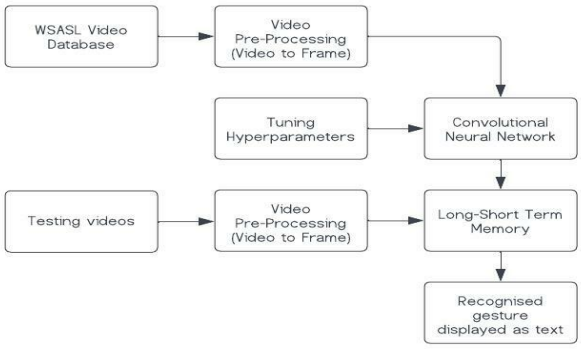

Figure 1. Proposed workflow

Python is an incredibly popular programming language and Python 3 was particularly useful in this project. This is because there are a wide range of libraries available to analyse image data. This project mainly used the following packages:

- NumPy: to hold the group of images to test the model.
- Keras and TensorFlow: to train the data-set.
- OpenCV: to capture the videos and process the videos as images.

As mentioned in section 1.1, we took inspiration from 4 previous works that used CNN and ResNet50. After reading these papers, we wanted to use CNN to extract the frames and ResNet50 to classify the videos. However, after conducting more research, we decided to use LSTM instead of ResNet50.

This is because LSTM is a Recurrent Neural Network (RNN) wherein you have 4 gates which help to remember or forget previous context. LSTM is especially useful as sign language is a dynamic language and to classify sign language videos, all sequential moves are crucial for classification. Furthermore, LSTM is often used for real-time recognition whereas ResNet is more often used for non real-time recognition. When one wants to include live captioning, LSTM would be relatively more useful.

Finally, we used transfer learning by implementing the pre-trained MobileNet v2 model. Transfer learning models often achieve the optimal performance quicker than traditional machine learning models as the pre-trained model leverages already known knowledge (for example features and weights).

4. Experiments

The video classification task has evolved according to technological advances in terms of parallel computing and hardware. Trends in video classification has been evolving throughout the years, finally reaching Convolutional 3D Networks to capture useful temporal features. Nevertheless, due to the scope of this course and limitations in our hardware, we are going to work with one of the first proposed approaches to video classification: The CNN and LSTM method. This can offer great results with our even "a priori" simpler CNN work frame as we can consult in [**?**]. The workflow therefore will be similar to what is stated in section 3. We have a video data set of World-Level American Sign Language made from different source videos which are very diverse in terms of video characteristics (as explained in section 3).

The first step was to pre-process the data as utilising the raw images to train the data in their current state might lead to very low performance rates. Due to our memory restrictions the data is pre-processed by extracting the individual frames from the video and resizing the images to $77 \times 48 \times 3$ shape. Resizing the frames will increase the computational efficiency of computations in the experimental phase. The images are also padded beforehand because as each video has a different resolution, but after padding we set it to be 1.5.

For the purpose of feature extraction and efficiency, we did not train a whole CNN. Instead, we made use of transfer learning from a previously trained model. The model selected is MobileNetV2.

This selection was made after deciding that small parameter models could work as well as other larger models as ResNet50. Since this model has a Conv2D layer at the end, we remove it and set an AveragePooling Layer. By doing this we save time and make our workflow more efficient. The image set weights are obtained from a previous training on "ImageNet" set.

Our input would be of size [N, T, W, H, C] where:

- N: *batch size*.
- T: *temporal extension* of the video in frames.
- W: *width* of the video frame.
- H: *height* of the video frame.
- C: *channels* of the video frame.

The features were calculated by feeding the MobileNetV2 model using the above input and generated a vector of 1280 features for each of the T = 40 frames. Hence, we end up having a variable of size [N, 40, 1280] which was fed into the Sequential Classifier.

The obtained feature vector was then fed into a Bidirectional LSTM. We averaged the predictions from each node in the forward and backward LSTM to increase the robustness of its predictions. Then Fully Connected Layers were then added to fit the Vocabulary Size.

4.1. Results

Accuracy and loss are two integral concepts that we have utilised to evaluate the performance of the model. Formally defined, accuracy is the sum of true positives and true negatives, which represents the number of correctly predicted data points, divided by the sum of the true positives, true negatives, false positives, and false negatives, which is the total number of data points [1]. In other words, the accuracy of the model aids in assessing how often an algorithm classifies a data point correctly [12].

The loss function is a mathematical function of the parameters of a machine learning algorithm used to ascertain how well an algorithm is modelling a data set. That is, loss is a number indicating how bad the model's prediction was on a single example. If the model's prediction is perfect, the loss is zero; otherwise, the loss is greater [12].

Another tool used to evaluate the model is the confusion matrix compares the predicted label to the true label and arranges this analysis in a 2D array. A confusion matrix helps present the results in a manner that aids in the visualisation of key model performance aspects such as its accuracy [9].

Table (1) shows the training and validation accuracies and losses. Here, we can see the training accuracy is 100% and validation accuracy is around 80%. Figs 2 and 3 show the change of accuracy and loss with the number of epochs. Compared to existing literature where the training and validation accuracy were 91% and 86% respectively [6], our results perform as well.

Table 1. Summary of training and validation accuracies and losses

Metric	Accuracy (%)	Loss
Training	100	0.0147
Validation	80	0.0347

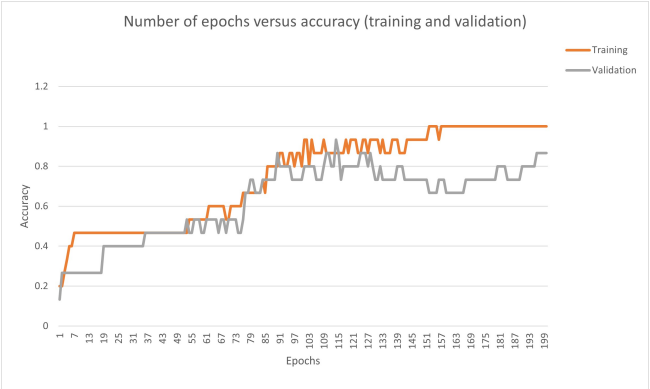

Figure 2. Training and validation accuracy against the number of epochs

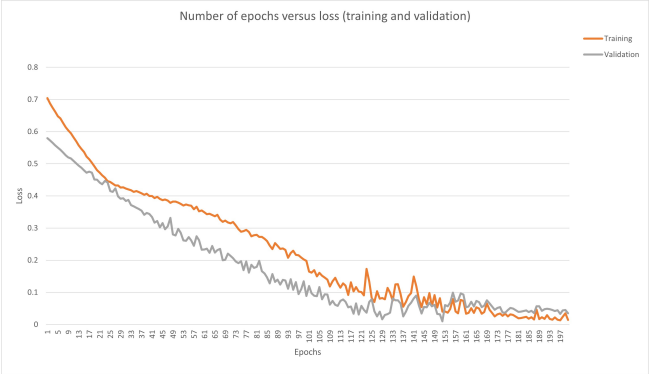

Figure 3. Training and validation loss against the number of epochs

We then ran the model over all videos in the testing data-set with 1098 labels. For the purposes of demonstration, for this report we created a confusion matrix consisting of 16 videos with 16 random labels that we put through the LSTM model. Fig (4) shows the obtained confusion matrix with the predicted and actual labels generated (for each frame). Generally, the testing accuracy over the entire testing data-set was quite good at 94% as seen in the matrix by how most predicted labels correspond with the actual labels.

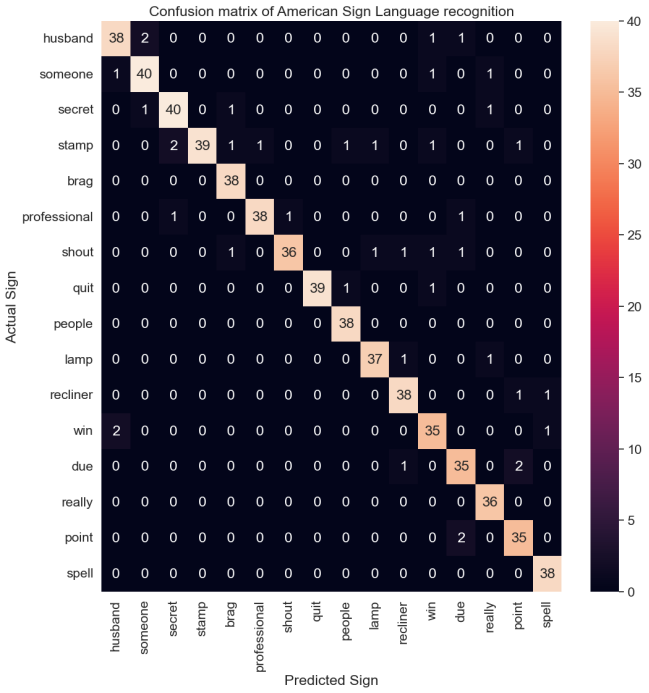

Figure 4. Confusion matrix of the testing videos

Figure 5. Example frames with captions

5. Conclusion

The problem of sign language recognition using videos is still a challenge. Similarity of gestures, user's accents, context and signs with multiple meanings lead to ambiguity. These are some of the reasons as to why previous works used limited data-sets. However, the data-set we used fulfilled our requirements as it had 12,000 videos with different types of people and therefore different manners of signing out different words.

This project explored the feasibility of using computer vision techniques to classify ASL videos to provide real-time captioning in video conferences. We used three algorithms – the pre-trained MobileNet v2, CNN and LSTM. CNN with five layers was used to extract the high-level training features, and LSTM with four layers to provide the suitable labels for each video frame.

With 12,000 videos and therefore 480,000 frames, the CNN and LSTM algorithms yielded a testing, validation, and training accuracy of 100%, 80% and 93% respectively. Compared to literature values of 91% and 86% for training and validation accuracy respectively, our algorithm performed adequately.

The video pre-processing on the database helped reduce the computational and storage complexity. A good set of training parameters was found by manually checking the performance of the model. Such a model will be useful in video conferences, increasing recognition speed while keeping the size and complexity of the neural network model at an acceptable level.

Currently, results show a high accuracy on the captured data-set, but the algorithm has been designed on our own experiences with tests and tutorials done before. It may be that adding more labels will decrease the model's precision. To improve the current system's design, we would start with a group of signs and test different configurations of the CNN to observe which configurations are the best. The group of signs will increase with additional classes and by using different evaluation methods, the best configuration cab be found. From this, our goal would be to find a relationship between the input number of classes and design of the CNN and LSTM so the system can construct an effective model for each case.

From our frames and predicted labels, we can then start to piece together the frames and include the appropriate captions, for example Fig 5 shows a few test frames with the label "recline". The next step in this process would be to explore real-time video captioning to completely fulfil our original idea of video-captioning for live conferences. Furthermore, we would explore different algorithms to see how well they would perform, for example VGG-16, ResNet, or AlexNet.

We could also use Hidden Markov Models to correct errors caused by the CNN and LSTM models. Sign language also does not have a direct translation of periods or prepositions. Query suggestion systems may be used in this case, but they have to be trained further in sign language transcriptions. Finally, as mentioned earlier, facial expressions can change the meaning of signs. The use of another model, such as another CNN model, should be useful in detecting facial expressions which could possibly improve the current system.

References

[1] Machine learning glossary and terms: Accuracy and error rate. 4

[2] Sign language interpretation using deep learning. *Edubirdie*, 2022. 2

[3] Kshitij Bantupalli and Ying Xie. American sign language recognition using deep learning and computer vision. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 4896–4899. IEEE, 2018. 2

[4] Risang Baskoro. World level american sign language video. 2

[5] Lijiya A C J, Sruthi. Signet: A deep learning based indian sign language recognition system. *IEEE*, 2019. 2

[6] Shruti Chavan, Xinrui Yu, and Jafar Saniie. Convolutional neural network hand gesture recognition for american sign language. *Department of Electrical and Computer Engineering, Illinois Institute of Technology*, 2021. 4

[7] Sajja Tulasi Krishna and Hemantha Kumar Kalluri. Deep learning and transfer learning approaches for image classification. *International Journal of Recent Technology and Engineering (IJRTE)*, 7(5S4):427–432, 2019. 2

[8] Irit Meir, Wendy Sandler, Carol Padden, and Aronoff Mark. The oxford handbook of deaf studies, language, and education. *The Oxford Handbook of Deaf Studies, Language, and Education*, 2:267–280. 1

[9] Sarang Narkede. Understanding confusion matrix. 2018. 4

[10] G Anantha Rao, K Syamala, PVV Kishore, and ASCS Sastry. Deep convolutional neural networks for sign language recognition. In *2018 Conference on Signal Processing And Communication Engineering Systems (SPACES)*, pages 194–197. IEEE, 2018. 2

[11] Wendy Sandler and Diane Lillo-Martin. Sign language and linguistic universals. *Cambridge University Press*, 2006. 1

[12] Shankar. Understanding loss function in deep learning. 2022. 4

[13] Ruth Wario. Sign language gesture recognition through computer vision. *IEEE*, 2018. 1

[14] Xiao Xiao and Wanggen Wan. Human pose estimation via improved resnet-50. 2017. 2