American Sign Language Fingerspelling Interpreter

Team 36: Sophia Marquez, Wei Shao, Guanqiao Wang, Austin Maung

GitHub Repository: https://github.com/YoungKameSennin/Team36-SignLanguageInterpreter

1 Introduction

While languages around the world share their similarities and differences, sign language has a very unique feature being that it involves hand gestures rather than speaking out loud. This provides the opportunity to hold natural conversations even with people that are hearing impaired. However when it comes to its translation it requires the other party to hold the knowledge of understanding what the meaning behind each gesture is which is not that some would consider common knowledge. Unlike other languages that have the potential of sharing common ancestors and thus sometimes having similar words, sign language is more of its own separate making.

However, with the help of technology and machine learning there has been research into the subject of interpreting sign language. One focus from this research has been translating sign language between different sign language systems. Due to there being many languages, sign languages were developed in conjunction with the prevailing language in whatever area of the world it was needed. This led to differences amongst the sign languages developed and so even if they use hand gestures each gesture can mean something different depending on the system.

Based on our research of past works, we decided to build a machine learning model which can interpret fingerspelling. Fingerspelling is a subset of sign language where individual characters are represented with fingers and are used together to create words and sentences. Our current working dataset focuses only on the fingerspelling of American Sign Language (ASL) rather than multiple different systems to not confuse the model with differing meanings. Our model works by using inputted images to predict the written English equivalent of the sign shown in the image. The model used is a Convolutional Neural Network (CNN) which is one of the more effective models that can be used to classify images. Once the model was trained with the dataset, measurements such as accuracy and precision were used to determine that the model was a viable solution to the problem we are trying to solve.

2 Literature Review

In "Sign language identification and recognition: A comparative study", an examination is made between different machine learning and deep learning models and their ability to first detect and translate sign language into written words. These words are Sign Language Recognition (SLR), as described in the paper as well as in the chapter, "Sign Language Recognition " of the book, "Visual Analysis of Humans". The paper goes further in detecting and translating between two different sign language systems, calling it Sign Language Identification (SLID). This is important because most research has been done for the former but much more is needed for the ladder in order to help people within different sign language communities to understand each other.

Both resources explain how sign language is made up of multiple parts, facial expressions, body language, and lastly the actual signing in order to convey meaning and allow people to communicate and understand each other. The chapter goes further, describing how instead of simple symbols like letters and numbers, the signing is used to show phonemes and combined together to formulate words, part of speech, and sentences. Our model focuses on simply translating each symbol into written English, those being letters and numbers. This is because the scope of the project would become too

difficult if we moved on from simply detecting and differentiating different hand signs to translating full phrases and sentences.

In the paper, a number of datasets were trained on in order to expose the model on different characteristics of sign language that can be detected on. For example there were datasets focussed on training the model on skin and body detection, gesture recognition, and sign identification. In contrast, our model is only trained on one dataset, instead of training the model and extracting the important features of each dataset onto our model, we just hoped that the model was strong enough to detect the different signs.

Lastly, the model from the paper was trained using different technologies such as XBOX Kinect with specialized gloves, an example of the data gloves with accelerometers described within the chapter, that could be mapped in 3D space computationally. This allowed their model to circumvent certain issues like edge detection and color differentiation by instead focusing their models to train on vectors that formed the hands with their signs. In contrast, our model had to handle all the images of the dataset, discerning different hand signs from others, without all the optimizations along the way.

3 Dataset Description

The dataset used to train and test our model is made up of only images. Images are split amongst 36 different folders each pertaining to a letter (A - Z) or digit (0 - 9). The dataset contains 2,520 total images, each the same size of 400 by 400 pixels as well as all are jpeg images. The images are of a singular hand doing the equivalent ASL fingerspelling. Some images are taken all the way to the wrist while others are only to the end of the hand. Each folder contains a total of 70 images of the same character taken from slightly different angles and have slightly different lighting. The images have all been removed from the background which they were originally taken in and put in front of a completely blackground before they are run through the model.

No images were excluded from the dataset as outliers. However there were certain folders of images which can be considered to be very varied compared to the rest of the dataset. One such example is "j" due to the fingerspelling of this letter including movement of the hand which can not be captured by images. Certain images in this folder are not uniform due to capturing the spelling of the letter mid movement. In the dataset there was another problem of similar fingerspelling of characters across multiple folders in the dataset. ASL is a language that is very dependent on the context of the conversation. Due to this and the limitations of being a language, using only two hands can lead to having similar signs meaning different things or words. In our dataset the fingerspelling for 'f' and '9' are both similar in that the thumb and index finger are touching while the rest of the fingers are pointed upwards. There is a difference of 'f' having the rest of the fingers touching while in '9' the fingers are spread apart. Though this is a difference in the dataset, a person doing the fingerspelling in a conversation might not do this difference which could lead to incorrect results from the model.

4 Proposed Solution

The project involves the classification of American Sign Language (ASL) gestures, which is a multi-classification problem with image data. To tackle this problem, we propose a methodology that utilizes Convolutional Neural Networks (CNNs) for their effectiveness in image-related tasks. Additionally, we consider incorporating hyperparameter tuning and regularization techniques to enhance model performance.

4.1 Model Selection

CNNs have proven to be successful for image-related problems, including sign language gesture recognition. Therefore, we choose CNN as our primary model for this project. CNNs are well-suited for capturing spatial features in images, allowing them to learn discriminative patterns from ASL gesture images. The model's architecture, which is a sequence of layers to optimize classification. The foundation comprises three Conv2D layers with increasing filters, it detects complicated patterns. Then we used MaxPooling2D layers to reduce spatial dimensions, also controlling computation and overfitting. The Flatten layer transforms the pooled feature map matrix into a single column, which will be fed

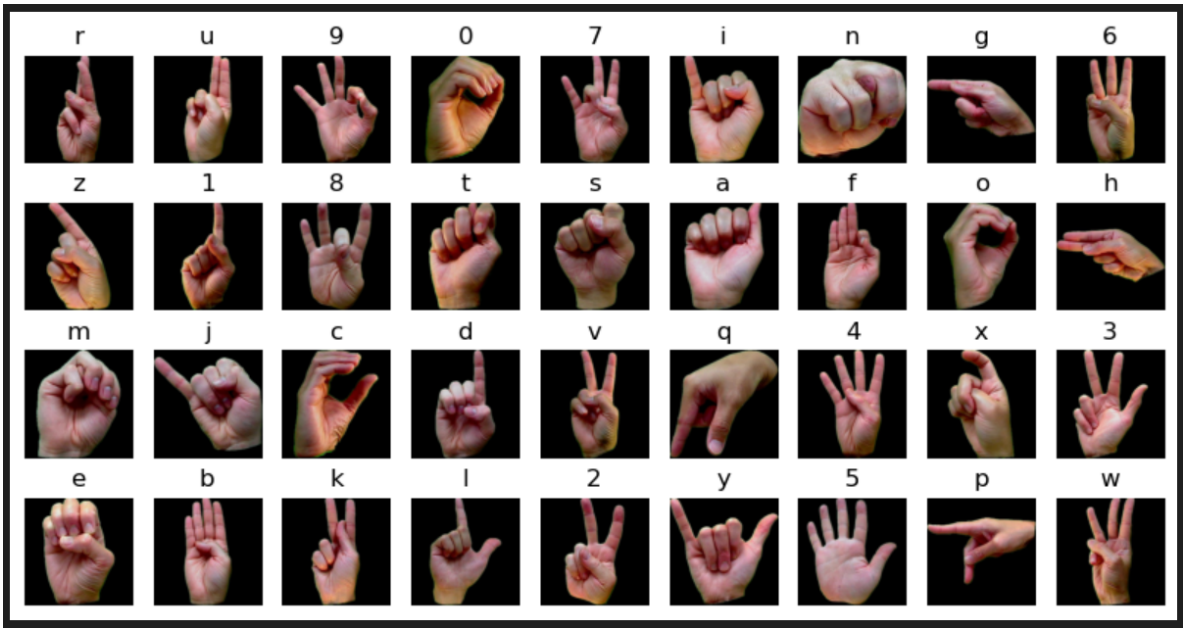

Figure 1: Demonstration of dataset.

into the neural network. We also added a fully-connected Dense layer with 256 neurons, where high-level feature learning occurs. Finally, we added an output Dense layer, utilizing a 'softmax' activation function to output a probability distribution over the 36 classes.

4.2 Preprocessing

As the ASL dataset consists of images, we need to preprocess the input data to ensure compatibility with the chosen models. The preprocessing steps may involve resizing the images to a consistent size, normalizing the pixel values, and splitting the dataset into training and validation sets.

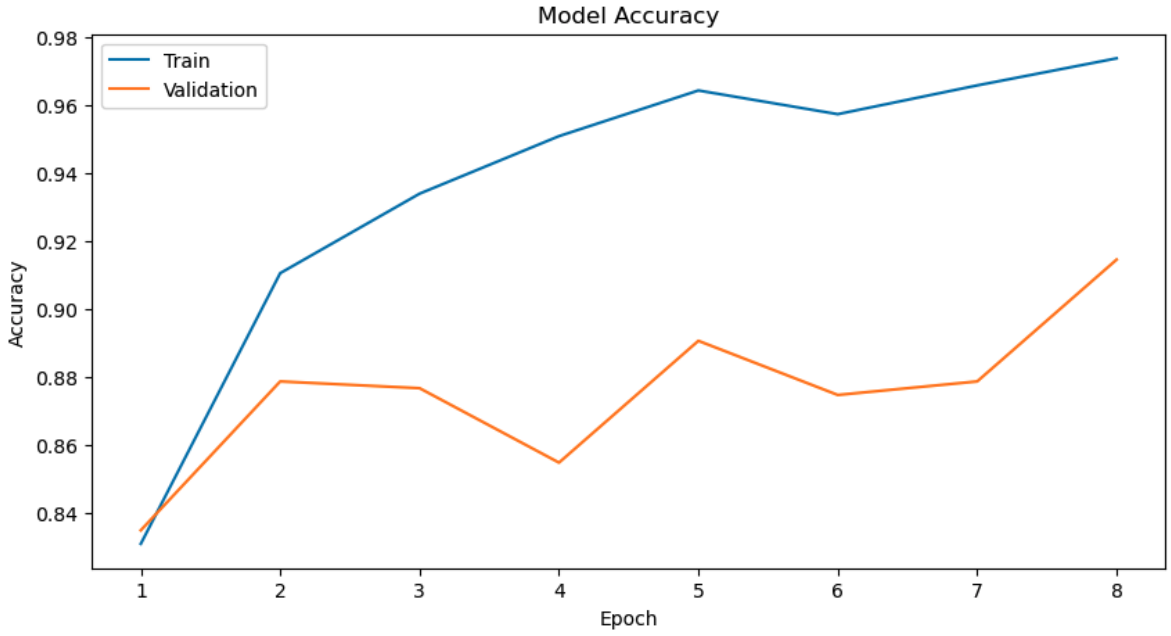

Figure 2: Accuracy vs. Epochs

4.3 Hyper Parameter Tuning

To optimize the performance of our models, we plan to use hyperparameter tuning. This involves systematically trying for the best combination of hyperparameters, like learning rate, batch size, number of layers, and filter sizes, using techniques like grid search and random search. The model is compiled using the 'adam' optimizer and 'categorical_crossentropy' as the loss function. By tuning these parameters, we are aiming to improve the model's accuracy and capabilities.

4.4 Regularization Techniques

Overfitting is a common challenge in deep learning models. To address this issue, we will apply regularization techniques such as dropout and L2 regularization. Dropout randomly drops out a fraction of the neurons during training, preventing the network from relying too heavily on specific features. L2 regularization adds a penalty term to the loss function, discouraging the model from assigning excessive importance to any particular feature. Additionally, the model has an early stopping mechanism during training, which halts the learning process when validation performance ceases to improve. It prevents the model from memorizing noise in the training data.

4.5 Model Evaluation

After the models are trained, we will evaluate their performance using appropriate evaluation metrics such as accuracy, precision, recall, and F1-score. We will generate a classification report, providing metrics for each individual class, to gain deeper insights into the model's performance for each class. This report will give us a comprehensive understanding of the model's strengths and weaknesses in recognizing ASL gestures.

5 Experimental Results

	precision	recall	f1-score	support
0	0.94	0.90	0.92	70
1	0.96	1.00	0.98	70
2	0.83	0.99	0.90	70
3	0.99	1.00	0.99	70
4	0.93	1.00	0.97	70
5	1.00	0.96	0.98	70
6	0.88	0.93	0.90	70
7	1.00	0.97	0.99	70
8	1.00	0.97	0.99	70
9	0.97	1.00	0.99	70
a	0.93	0.99	0.96	70
b	1.00	1.00	1.00	70
c	1.00	1.00	1.00	70
d	0.97	1.00	0.99	70
e	1.00	1.00	1.00	70
f	1.00	1.00	1.00	70
g	0.96	1.00	0.98	70
h	1.00	0.97	0.99	70
i	1.00	1.00	1.00	70
j	1.00	1.00	1.00	70
k	0.99	1.00	0.99	70
...				
accuracy			0.97	2515
macro avg	0.97	0.97	0.97	2515
weighted avg	0.97	0.97	0.97	2515

Figure 3: Testing Results of CNN

The proposed methodology was implemented to train a Convolutional Neural Network (CNN) model for the classification of American Sign Language (ASL) gestures. The training process concluded

after 8 epochs, and the model's performance was evaluated using a classification report.

The classification report provides insights into the precision, recall, and F1-score for each class, along with the support (number of samples) for each class. The overall accuracy achieved by the model was 97%, indicating a high level of accuracy in predicting ASL gestures.

The precision scores ranged from 83% to 100% for different classes, with an average precision of 97%. This metric measures the ability of the model to correctly classify instances for a specific class. Classes such as '2', '6', and 'g' achieved slightly lower precision scores, indicating some challenges in accurately classifying these ASL gestures. However, the majority of the classes achieved precision scores above 90%, demonstrating the model's effectiveness in recognizing ASL gestures.

The recall scores ranged from 90% to 100% for different classes, with an average recall of 97%. Recall represents the ability of the model to correctly identify instances of a particular class. Similar to precision, some classes, such as '2', '6', and 'g', achieved slightly lower recall scores. However, most classes achieved recall scores above 95%, indicating the model's capability to identify ASL gestures accurately.

The F1-scores ranged from 90% to 100% for different classes, with an average F1-score of 97%. The F1-score is a harmonic mean of precision and recall, providing an overall measure of a model's performance. The high F1-scores achieved by the majority of the classes indicate the model's ability to achieve a balance between precision and recall for accurate classification.

In conclusion, the experimental results demonstrate the successful application of the proposed methodology, resulting in a highly accurate CNN model for ASL gesture classification. The achieved performance metrics validate the potential of the model in practical ASL-related applications.

6 Conclusion and Discussion

In our project, we successfully developed an American Sign Language (ASL) interpreter using a machine learning approach based on Convolutional Neural Networks (CNNs). The program we created is designed to interpret images of ASL into corresponding text outputs, demonstrating the capabilities of CNNs in image recognition.

The model's architecture, consisting of three convolutional layers, max-pooling layers, a flattening layer, and two densely connected layers, played a significant role in achieving an impressive accuracy of 95%. This indicates that our model can be a very powerful tool for interpreting ASL static images and potentially bridging communications. Additionally, our model's training and validation process showed its generalization capabilities. Using the 'adam' optimizer, 'categorical_crossentropy' as the loss function, and 'accuracy' as the performance metric, along with early stopping and L2 regularization, ensured the model avoided overfitting while maintaining high predictive performance. This was validated by the split of our image dataset into 80/20 training and validation subsets.

Despite the model's strong performance, there's always room for improvement. In future iterations of this project, we might explore augmenting the dataset with a wider range of hand gestures or angles. One significant avenue for improvement could be expanding the model's ability to interpret ASL from video data, not just static images. Video interpretation would enhance the model's ability in real-world scenarios. This could potentially lead to an even more comprehensive ASL interpretation.

Another aspect to consider is the impact of the image's background on the model's accuracy. In our current project, we found that images with purely black backgrounds lead to better results. However, in real-world, the background of images might be varied and complex. Developing a solution to maintain high accuracy on different backgrounds could involve employing techniques such as background subtraction. This improvement would significantly increase the model's effectiveness in real-life environments.

This project has far-reaching implications to the communities. Our ASL interpreter can serve as a simple communication tool bridging the deaf community and those who do not understand sign language. With further development, this technology could be integrated into various platforms and devices, from educational tools to real-time interpretation services during live events or broadcasts. The potential for expanding accessibility and inclusion is immense.

References

Dataset

https://www.kaggle.com/datasets/ayuraj/asl-dataset

Literature Review

Sultan, Ahmed, Makram, Walied, Kayed, Mohammed and Ali, Abdelmaged Amin. "Sign language identification and recognition: A comparative study" Open Computer Science, vol. 12, no. 1, 2022, pp. 191-210. https://doi.org/10.1515/comp-2022-0240

Cooper, H., Holt, B., Bowden, R. (2011). Sign Language Recognition. In: Moeslund, T., Hilton, A., Krüger, V., Sigal, L. (eds) Visual Analysis of Humans. Springer, London. https://doi.org/10.1007/978-0-85729-997-0_27