Developing a Classification Model for American Sign Language

Pratik Narendra Borse
Computer Science
Illinois Institute of Technology
pborse1@hawk.iit.edu

Ashlesh Khajbage
Data Science
Illinois Institute of Technology
akhajbage@hawk.iit.edu

Nikhil Singh Thakur
Artificial Intelligence
Illinois Institute of Technology
nthakur4@hawk.iit.edu

PROBLEM STATEMENT

The objective of this research project is to design and develop a highly accurate real-time classification model capable of recognizing and interpreting American Sign Language (ASL) gestures from video data. ASL, a visual language used by the deaf and hard-of-hearing community in the United States, can be better understood by a machine learning model, thereby improving communication accessibility for this community.

However, recognizing ASL signs from video data poses significant challenges, including variations in lighting, camera angles, and hand movements, which can cause variations in the appearance of signs. Furthermore, the context in which ASL signs are used can significantly impact their meaning, emphasizing the importance of considering the surrounding signs and context when classifying individual signs. Additionally, the complexities of ASL, including the use of facial expressions and body language in conveying meaning, further complicates the task of ASL sign recognition.

To address these challenges, a large and diverse dataset representing a range of signers, signing styles, and environments must be utilized to train the model effectively. The model should leverage deep learning techniques such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to capture both spatial and temporal features in the video data.

The success of this research project will be evaluated based on the accuracy and speed of the proposed model in recognizing ASL signs in real-time. Future applications of this technology could potentially be integrated into various virtual assistants, communication tools, and educational resources, providing a more inclusive and accessible experience for the deaf and hard-of-hearing community.

Despite the significant progress in ASL recognition in recent years, there still exists a considerable gap in the accuracy and robustness of the existing methods. Existing models are often limited by their inability to capture the complex and dynamic nature of ASL signs and the context in which they are used. As a result, the current state-of-the-art models may struggle in real-world applications, particularly in noisy environments or when signers exhibit variations in their signing styles.

This research project aims to address these limitations by developing a more sophisticated and robust model that can generalize well to different signers, signing styles, and environments. The proposed model will be designed to capture not only the spatial and temporal features of the video data but also the context and meaning of the signs within a broader linguistic context.

The potential impact of this research project extends beyond the realm of assistive technology for the deaf and hard-of-hearing community. The ability to recognize and interpret ASL signs from video data has implications for a wide range of fields, including human-computer interaction, robotics, and surveillance. Additionally, the insights gained from this research could potentially inform the development of similar recognition models for other sign languages, expanding the reach of this technology to other communities worldwide.

KEYWORDS

Classification Model, Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Image Processing, Spatial Features, Temporal Features.

DATASET

The ASL Alphabet dataset contains over 87,000 labeled images of hand gestures representing each letter of the American Sign Language (ASL) alphabet. The dataset is split into two parts: a training set containing 64,800 images and a test set containing 26,400 images. The images were captured using a Leap Motion Controller, which is a small, USB-powered device that uses two cameras and three infrared LEDs to track hand and finger movements in 3D space. The images are in the RGB format and have a resolution of 200x200 pixels. The images were captured from a diverse set of signers, including both genders, a range of ages, and different skin colors, to ensure a wide variety of signing styles and hand shapes were captured. Each image was labeled with the corresponding letter of the ASL alphabet using a hand-labeled ground truth, making it possible to train and test machine learning models on this dataset. This dataset is suitable for a range of computer vision tasks, including image classification, object recognition, and hand gesture recognition. The dataset is well-suited for training and evaluating deep learning models, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), which are widely used for image classification and sequence modeling tasks.

The ASL Alphabet dataset has numerous potential applications in real-world scenarios, such as virtual assistants, sign language translation tools, and assistive technology for the deaf and hard-of-hearing community. The dataset can also be used to better understand the complexities of ASL and inform the development of more advanced recognition models for sign language. The dataset consists of 29 classes, one for each letter of the American Sign Language alphabet, represented in the images as hand gestures. The images in the dataset were captured in a variety of lighting conditions, and the backgrounds include both plain and textured surfaces. In addition to the training and testing sets, the dataset also includes a validation set containing 8,100 images. This can be used to tune the hyperparameters of machine learning models or to perform early stopping during training to prevent overfitting. The dataset is balanced, with each class containing approximately the same number of images. This ensures that machine learning models are trained on an equal number of examples from each class, which can help prevent the model from being biased towards one particular class.

The dataset was collected and labeled by a team of researchers from the National Institute of Standards and Technology (NIST) and the American Sign Language Linguistic Research Project (ASLLRP). The team used a standardized protocol for data collection and labeling, which ensures consistency across different signers and labeling styles.

Overall, the ASL Alphabet dataset is a valuable resource for researchers and practitioners working in the field of computer vision and machine learning. Its high quality, diversity, and balance make it well-suited for developing and evaluating algorithms for image classification, object recognition, and hand gesture recognition tasks, particularly those related to American Sign Language.

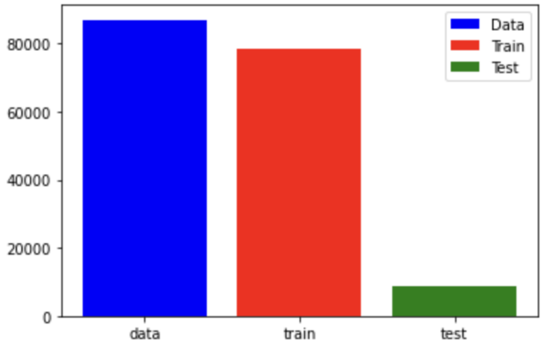


Fig.1 Distribution of Dataset into Training and Testing sets.

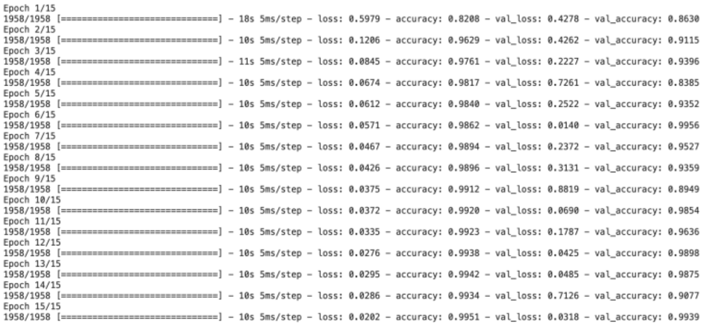
COMPLETED WORK

As it was mentioned in the Project plan section of our Project Proposal, we have completed a series of tasks so far that follow the timeline of the entire project as a whole.

1. Gather and review existing research on ASL recognition using machine learning techniques. - The existing research on ASL recognition using machine learning techniques covers a broad range of topics, including image and video processing, feature extraction, and classification algorithms. Some of the recent approaches in this area have employed deep learning techniques, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), to improve the accuracy of ASL recognition. Several datasets have also been developed for ASL recognition, including the ASL Alphabet dataset that contains images of hand gestures representing each letter of the ASL alphabet. Other datasets include video recordings of signers performing phrases or sentences in ASL, which can be used to study the temporal dynamics of sign language.

2. Collect a diverse dataset of ASL gestures and pre-process the data. – As mentioned previously, we have selected a dataset that is diverse in nature and has a considerable amount of training and testing images in order to build a classification model.

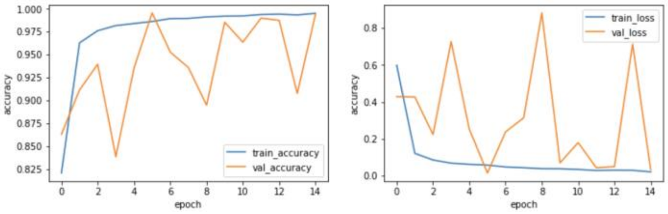
3. Executing a baseline model and computing the accuracy and validation loss for each iteration – During the development of a classification model for American Sign Language (ASL), it is important to analyze the training and validation accuracy and loss for each epoch of training. The training accuracy measures the percentage of correctly classified training examples in each epoch, while the validation accuracy measures the percentage of correctly classified examples in the validation set. By analyzing the training and validation accuracy for each epoch, it is possible to track the progress of the model over time and identify any overfitting or underfitting issues. Similarly, the training loss measures the error rate in the training set, while the validation loss measures the error rate in the validation set. By analyzing the training and validation loss for each epoch, it is possible to determine whether the model is converging or diverging and adjust the model's architecture or hyperparameters as needed.



```
Epoch 1/15
1958/1958 [==============================] - 18s 5ms/step - loss: 0.5979 - accuracy: 0.8208 - val_loss: 0.4278 - val_accuracy: 0.8630
Epoch 2/15
1958/1958 [==============================] - 10s 5ms/step - loss: 0.1206 - accuracy: 0.9629 - val_loss: 0.4262 - val_accuracy: 0.9115
Epoch 3/15
1958/1958 [==============================] - 11s 5ms/step - loss: 0.0845 - accuracy: 0.9761 - val_loss: 0.2227 - val_accuracy: 0.9396
Epoch 4/15
1958/1958 [==============================] - 10s 5ms/step - loss: 0.0674 - accuracy: 0.9817 - val_loss: 0.7261 - val_accuracy: 0.8385
Epoch 5/15
1958/1958 [==============================] - 10s 5ms/step - loss: 0.0612 - accuracy: 0.9840 - val_loss: 0.2522 - val_accuracy: 0.9352
Epoch 6/15
1958/1958 [==============================] - 10s 5ms/step - loss: 0.0571 - accuracy: 0.9862 - val_loss: 0.0140 - val_accuracy: 0.9956
Epoch 7/15
1958/1958 [==============================] - 10s 5ms/step - loss: 0.0467 - accuracy: 0.9894 - val_loss: 0.2372 - val_accuracy: 0.9527
Epoch 8/15
1958/1958 [==============================] - 10s 5ms/step - loss: 0.0426 - accuracy: 0.9896 - val_loss: 0.3131 - val_accuracy: 0.9359
Epoch 9/15
1958/1958 [==============================] - 10s 5ms/step - loss: 0.0375 - accuracy: 0.9912 - val_loss: 0.8819 - val_accuracy: 0.8949
Epoch 10/15
1958/1958 [==============================] - 10s 5ms/step - loss: 0.0372 - accuracy: 0.9920 - val_loss: 0.0690 - val_accuracy: 0.9854
Epoch 11/15
1958/1958 [==============================] - 10s 5ms/step - loss: 0.0335 - accuracy: 0.9923 - val_loss: 0.1787 - val_accuracy: 0.9636
Epoch 12/15
1958/1958 [==============================] - 10s 5ms/step - loss: 0.0276 - accuracy: 0.9938 - val_loss: 0.0425 - val_accuracy: 0.9898
Epoch 13/15
1958/1958 [==============================] - 10s 5ms/step - loss: 0.0295 - accuracy: 0.9942 - val_loss: 0.0485 - val_accuracy: 0.9875
Epoch 14/15
1958/1958 [==============================] - 10s 5ms/step - loss: 0.0286 - accuracy: 0.9934 - val_loss: 0.7126 - val_accuracy: 0.9077
Epoch 15/15
1958/1958 [==============================] - 10s 5ms/step - loss: 0.0202 - accuracy: 0.9951 - val_loss: 0.0318 - val_accuracy: 0.9939
```

4. Ideally, the model should achieve high accuracy and low loss on both the training and validation sets, indicating that it is able to generalize well to new examples. However, if the training accuracy and loss are much better than the validation accuracy and loss, it may

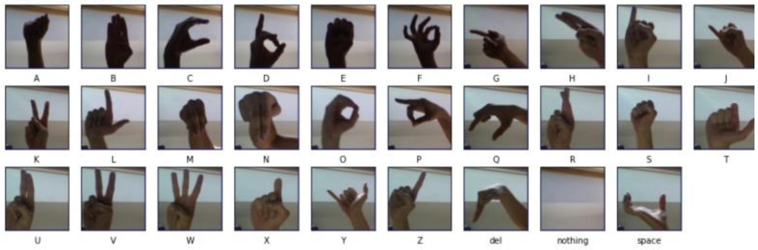
indicate overfitting, where the model is fitting too closely to the training data and not able to generalize well to new examples. In this case, regularization techniques, such as dropout or weight decay, may be used to prevent overfitting. Overall, analyzing the training and validation accuracy and loss for each epoch is a crucial step in the development of a classification model for ASL, as it allows for the identification of potential issues and optimization of the model's performance.



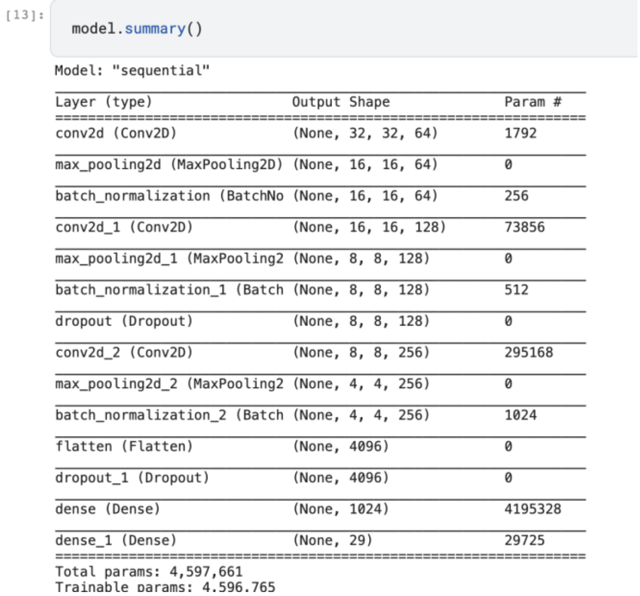
The above image shows the training accuracy, validation accuracy, training loss and validation loss of the classification model that we implemented.

5. In order to ensure that the model was working correctly, we used the below testing dataset.

```
<Figure size 432x288 with 0 Axes>
```



6. Below is the model summary that we obtained. –



```
[13]:  model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 32, 32, 64)        1792
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 16, 16, 64)        0
_________________________________________________________________
batch_normalization (BatchNo (None, 16, 16, 64)        256
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 16, 16, 128)       73856
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 8, 8, 128)         0
_________________________________________________________________
batch_normalization_1 (Batch (None, 8, 8, 128)         512
_________________________________________________________________
dropout (Dropout)            (None, 8, 8, 128)         0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 8, 8, 256)         295168
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 4, 4, 256)         0
_________________________________________________________________
batch_normalization_2 (Batch (None, 4, 4, 256)         1024
_________________________________________________________________
flatten (Flatten)            (None, 4096)              0
_________________________________________________________________
dropout_1 (Dropout)          (None, 4096)              0
_________________________________________________________________
dense (Dense)                (None, 1024)              4195328
_________________________________________________________________
dense_1 (Dense)              (None, 29)                29725
=================================================================
Total params: 4,597,661
Trainable params: 4.596.765
```

NEXT STEPS

Initially we have used a pre-determined testing dataset in order to measure the accuracy of the model. Although this approach gives us almost 99% accuracy, the input given to the model is not a real-time input. Therefore, one of the most important update to the project would be to accept real time video input and classify the hand gestures into the alphabets of the American Sign Language.

Apart from this important feature, we would also be looking to optimize the model's hyperparameters so that it returns an even better accuracy. Optimizing the hyperparameters of a machine learning model is an important step in improving its performance. Hyperparameters are parameters that are not learned from the data but are set before the training process, such as learning rate, batch size, and number of epochs. These hyperparameters can significantly affect the performance of a model, and finding the optimal combination can be a challenging task.

Another important step in the project is to try and explore other approaches to creating a classification model such as CNNs, RNNs and Graph Neural Networks. Once we are able to implement a model using each of these approaches, we can form an in-depth analysis of each approach and perform a comparative study of those networks.

Incorporating other modalities such as facial expressions and body movements into a classification model for ASL recognition can enhance the accuracy and robustness of the model. Facial expressions and body movements play a crucial role in conveying meaning in ASL and ignoring them can lead to misinterpretations of signs.

One approach for incorporating facial expressions and body movements into the model is to use a multimodal framework that combines information from different modalities, such as video and audio. For example, the model can be trained on video sequences that include both hand gestures and facial expressions, using techniques such as 3D convolutional neural networks (CNNs) to capture spatial and temporal features from the data.

Another approach is to use an attention mechanism that focuses on relevant regions of the input, such as the face and body, while ignoring irrelevant regions. This can be achieved by training the model with an auxiliary task, such as face recognition, that encourages the model to learn relevant features from the face.

Incorporating other modalities can also help in addressing the issue of variability in signing styles and contexts. For example, different signers may use different facial expressions or body movements to convey the same sign, and incorporating these modalities can help the model to generalize better across different signers and contexts.

Overall, incorporating other modalities such as facial expressions and body movements into the model can improve the accuracy and robustness of the ASL recognition model and provide a more comprehensive and inclusive representation of the language.

REFERENCES

[1] Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, Richard Bowden, 2020. *Sign Language Transformers: Joint End-to-end Sign Language Recognition and Translation*.
DOI: https://doi.org/10.48550/arXiv.2003.13830

[2] Jie Huang, Wengang Zhou, Houqiang Li and Weiping Li, "Sign Language Recognition using 3D convolutional neural networks," 2015 IEEE International Conference on Multimedia and Expo (ICME), Turin, 2015, pp. 1-6, doi: 10.1109/ICME.2015.7177428.

[3] S. Ikram and N. Dhanda, "American Sign Language Recognition using Convolutional Neural Network," 2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON), Kuala Lumpur, Malaysia, 2021, pp. 1-12, doi: 10.1109/GUCON50781.2021.9573782.

[4] de Amorim, Cleison Correia, David Macêdo, and Cleber Zanchettin. "Spatial-temporal graph convolutional networks for sign language recognition." Artificial Neural Networks and Machine Learning–ICANN 2019: Workshop and Special Sessions: 28th International Conference on Artificial Neural Networks, Munich, Germany, September 17–19, 2019, Proceedings 28. Springer International Publishing, 2019.

[5] Huang, J., Zhou, W., Zhang, Q., Li, H., & Li, W. (2018). Video-based Sign Language Recognition without Temporal Segmentation. AAAI Conference on Artificial Intelligence.