American Sign Language Character Recognition

TEAM SINGULARITY

Shourya Mupparapu, MS Computer Science (2023-25)

Divesh Pandey, Phd Finance (2023-28)

Sahith Reddy Beereddy, MS Computer Science (2023-25)

***Abstract*— The inability to speak constitutes a significant disability, compelling individuals to adopt alternative communication methods. Sign language emerges as a prevalent mode among these, enabling expression and interaction for those affected. This project addresses the communication challenges faced by individuals with speech disabilities by developing a machine learning-based sign language recognition system. Our vision-based application identifies finger-spelled letters A to I of American Sign Language (ASL), offering a foundational step towards facilitating seamless communication between the deaf community and those unversed in sign language. A collaborative dataset creation and a modified ResNet152 model form the core of our methodology. The resulting system achieved impressive test accuracy of 99.26, demonstrating high potential for real-world application and future expansion to comprehensive sign language interpretation.**

Keywords—Convolutional Neural Networks, Machine learning, Classification

I. INTRODUCTION

Communication takes various forms, with vocal speech being the most prevalent. However, for individuals who are deaf or have significant hearing impairments, vocal communication is not always viable. To bridge this gap, sign language was developed. It is a form of communication that relies on gestures, allowing those with hearing difficulties to express themselves effectively. Sign language is distinct from spoken language in that it uses a combination of facial expressions and body movements to convey messages.

Sign language is not universal. Like how different regions have their own spoken languages, sign languages vary globally. Examples include British Sign Language, Spanish Sign Language, and other countries have multiple sign languages in use. Sign language comprises several components, such as finger spelling, word-level sign vocabulary, and non-manual features. Finger spelling involves spelling out words letter by letter using hand gestures. This project specifically focuses on American Sign Language (ASL), particularly the letters A to I.

The methodology of this project includes collecting a comprehensive dataset of ASL images, pre-processing these images, extracting pertinent features, and choosing an appropriate machine learning model. This report will provide a detailed account of our approach and discuss the results we have achieved.

II. IMPLEMENTATION

The implementation of the machine learning model was a pivotal phase of the project, involving the deployment of the ResNet152 architecture to classify images of American Sign Language (ASL). The distinctive architecture of ResNet152, characterized by its residual blocks and global average pooling, was not chosen at random. The selection of the ResNet152 model was not a matter of chance but the result of careful and detailed hyperparameter tuning. We sifted through a multitude of settings, assessing each to determine the most suitable configuration for our classification challenge. Hyperparameters such as the learning rate, batch size, and number of epochs were systematically varied in a controlled manner, employing techniques such as grid search and random search to explore the hyperparameter space. This rigorous approach ensured that the model was finely tuned to the nuances of our dataset.

The algorithm at the heart of our model, the Convolutional Neural Network (CNN) with its ResNet152 variant, is renowned for its deep learning prowess. This deep learning algorithm operates by passing the input through a series of convolutional, ReLU, and pooling layers before reaching the fully connected layers that output the classification. Each convolutional layer acts as a feature detector, picking up on patterns and details within the images, which are then passed down through the network, getting refined and abstracted at each stage.

III. Experimentation

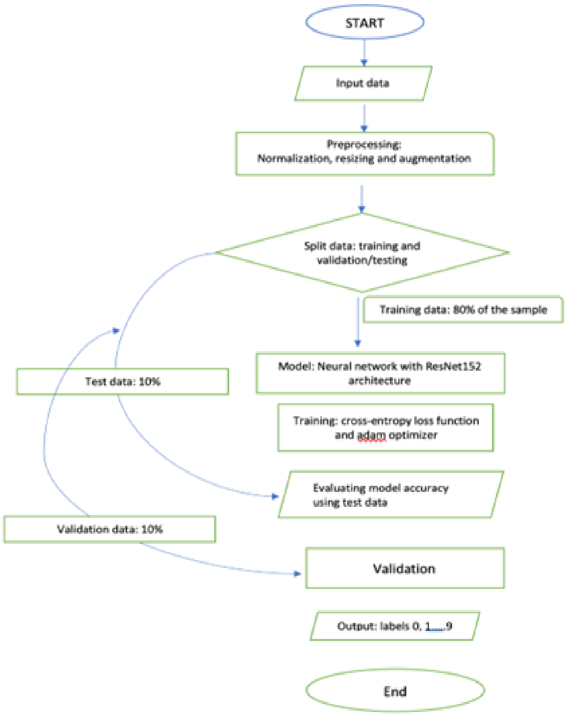

START
Input data
Preprocessing:
Normalization, resizing and augmentation
Split data: training and validation/testing
Training data: 80% of the sample
Test data: 10%
Model: Neural network with ResNet152 architecture
Training: cross-entropy loss function and adam optimizer
Evaluating model accuracy using test data
Validation data: 10%
Validation
Output: labels 0, 1.....9
End

Fig. 1. Model flowchart. (Neural network)

A. Data Collection

This project utilizes a dataset comprised of images formatted in .npy, categorically divided into training data and corresponding labels. It was collaboratively compiled by the students participating in the course, with each individual contributing a total of 90 images. This collection features images of American Sign Language (ASL), specifically representing letters A to I, captured in various background settings. The dataset's diversity in terms of hand shapes and backgrounds, crucial for a robust recognition system, is implicit in the data.

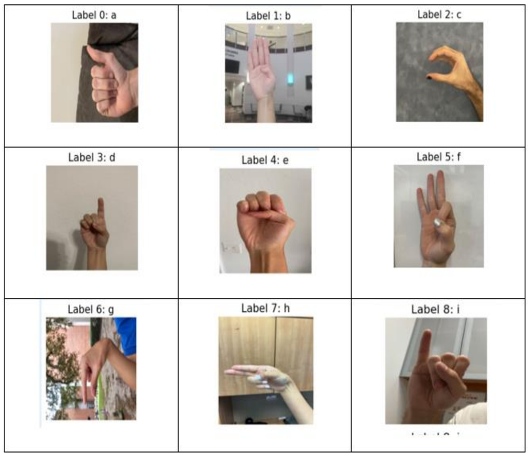


Fig. 2. Images in the Dataset

B. PreProcessing

In the preprocessing stage, a comprehensive pipeline is implemented to condition the images for optimal performance with the chosen model. Initially, all images are resized to the dimensions of 224x224 pixels. This specific size is chosen not only for computational efficiency but also to align with the architecture of the ResNet152 model, which requires input dimensions in multiples of 32.

To enhance the model's ability to generalize, data augmentation techniques are employed. These include a random horizontal flip and random rotation of the images, introducing necessary variability and robustness against overfitting. Additionally, standard preprocessing practices are also integrated into the pipeline. This encompasses image normalization, which standardizes the pixel values across the dataset, and manipulation of image brightness, further diversifying the training data and potentially improving the model's performance under varying lighting conditions.

C. Machine Learning Model

In this project, we use Convolutional Neural Networks (CNNs) with ResNet152 architecture to train the model and classify the images into their respective categories. CNN models have revolutionized the field of image classification, and the ResNet152 architecture is a powerful extension of traditional CNN models to address some of their limitations [1]. In the following section, we will delve into the significance of CNN and the ResNet152 architecture and how this combination enhances image classification tasks.

1) Model : Convolutional Neural Networks (CNNs)

Convolutional Neural Networks are a class of deep neural networks designed specifically for processing structured grid data, such as images. They consist of convolutional layers, pooling layers, and fully connected layers, and are highly effective in learning hierarchical representations of visual data.

CNNs have three key components:

a) Convolutional layers: These layers use convolutional operations to scan the input image with learnable filters, capturing local patterns and features.

b) Pooling Layers: These layers reduce the spatial dimensions of the input, preserving important information while discarding less relevant details.

c) Fully connected layers: Finally, these layers combine high-level features learned by the previous two layers to make predictions.

Although CNNs have been successful in image classification tasks, as they go deeper, they face challenges such as vanishing gradients and degradation in training accuracy. In this regard, ML experts use ResNet152 architecture, short for Residual

Network with 152 layers to address the issue of vanishing gradient particularly.

2) *Architecture : ResNet152*

ResNet152, a specific variant of CNNs was proposed by Kaiming He et al. [2] in their paper "Deep Residual Learning for Image Recognition." The fundamental innovation of ResNet is the introduction of residual blocks that allow the network to learn residual functions. Below, we list key features of ResNet152 architecture:

a) Residual blocks: Residual blocks contain shortcut connections, allowing the network to learn residual functions. The input to a residual block is added to its output, facilitating the flow of gradients during backpropagation.

b) Deep architecture: ResNet152 is exceptionally deep, with 152 layers. The depth is made possible by the use of residual connections, enabling the network to be more easily optimized and reducing the likelihood of vanishing gradients.

c) Global average pooling: Instead of relying on fully connected layers at the end, ResNet152 architectures use global average pooling, which helps reduce overfitting and parameter count.

Given these key features of a CNN combined with ResNet152 architecture, we explain how this combination helps in image classification and improves its efficiency. Given limited computation capacity and other constraints, we choose this method to ensure quick results while maintaining the accuracy of the results.

In this regard, ResNet152 does the following:

a) Addresses the issue of vanishing Gradients: The residual connections in ResNet152 mitigate the vanishing gradient problem. The gradient can flow more easily through the network during backpropagation, enabling the training of very deep networks without loss of information.

b) Improves training convergence: This factor is important to us given the constraints under which we are executing our project. Deeper networks often suffer from slower convergence during training. ResNet152's architecture helps in faster convergence, allowing the model to learn complex representations efficiently.

c) Ensure better feature extraction: The deeper architecture enables ResNet152 to capture intricate features and patterns in images, making it highly effective in tasks where fine details are crucial, such as image classification.

d) High accuracy: Due to its deep structure and advanced residual connections, ResNet152 achieves state-of-the-art performance in image classification benchmarks. It consistently outperforms shallower networks on such tasks where data has a high diversity and has large size. Our dataset of hand images has diverse backgrounds often hampering the effective learning in the model and ResNet152 overcomes this particular hurdle.

Therefore, we use this ResNet152 architecture in our image classification to ensure the accuracy of our results where detailed feature extraction is essential while maintaining deeper learning in the model.

IV. Unknown Class classification (Extra credit)

For the classification of unknown classes, a threshold-based approach was employed. In this method a specific threshold value for the classifier's confidence level is set. The class is categorized as -1 If the classifier's confidence in predicting any of the known classes falls below this threshold. This strategy effectively filters out predictions with lower certainty, reducing misclassification of signs that the model is not trained to recognize. The threshold set in the model according to the graph is 0.7.

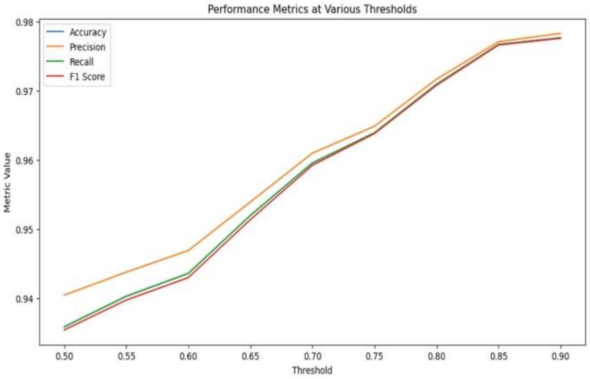
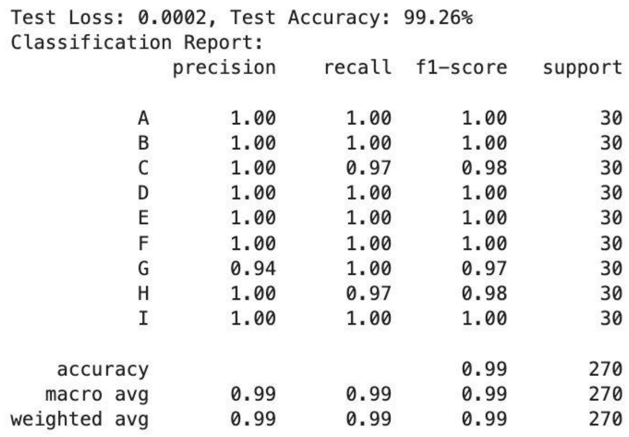


Fig. 3. Threshold graph

V. Results

Before the developed machine learning model exhibited a remarkable capacity for the accurate classification of American Sign Language (ASL) images, as evidenced by a high test accuracy rate of 99.26% and an exceedingly low test loss of 0.0002. The model's precision and recall were consistently high across all ASL letters tested, with many categories achieving a score of 1.00, indicating a very low occurrence of both false positives and false negatives. The F1-scores, which harmonize the precision and recall, maintained strong values for each letter, confirming the model's balanced and reliable classification abilities.

Uniformly high scores in the classification report for precision, recall, and F1-score across all letter categories underscore the model's effective generalization capabilities.



```
Test Loss: 0.0002, Test Accuracy: 99.26%
Classification Report:
              precision    recall  f1-score   support

           A       1.00      1.00      1.00        30
           B       1.00      1.00      1.00        30
           C       1.00      0.97      0.98        30
           D       1.00      1.00      1.00        30
           E       1.00      1.00      1.00        30
           F       1.00      1.00      1.00        30
           G       0.94      1.00      0.97        30
           H       1.00      0.97      0.98        30
           I       1.00      1.00      1.00        30

    accuracy                           0.99       270
   macro avg       0.99      0.99      0.99       270
weighted avg       0.99      0.99      0.99       270
```

Fig. 4. Classification Report

VI. Future Work

The successful identification of ASL's initial nine letters provides a solid platform for the project's progression. In future, the application will extend the dataset to encompass the entire ASL alphabet, aiming for an all-encompassing depiction of its manual communication system. The goal is to refine our model to not only detect each letter with precision but also to decode the sequences of finger-spelled words and phrases.

The project will also evolve to include the recognition of dynamic gestures, moving beyond static images to interpret the continuous flow of sign language gestures through sequential image processing or video analysis. We plan to investigate advanced temporal neural network models, including Long Short-Term Memory (LSTM) networks and 3D Convolutional Neural Networks (CNNs), renowned for their proficiency in managing sequential data, to enable this functionality.

VII. Conclusion

In this paper, we execute an image classification exercise applying Convoluted Neural Networks (CNNs) with ResNet152 architecture. Using this architecture in tandem with CNN addresses the issues associated with traditional CNN methods while ensuring that there is no tradeoff between convergence speed and accuracy. We apply this method to a diverse dataset sourced from students from various countries and backgrounds, thus ensuring a better learning process. Our results show that performance metrics such as recall, precision, and f1-score have a good score for most of the classes. We further calculate weighted metrics to find the average performance. This study adds to the well-established literature on CNNs that has provided evidence of CNN's efficiency in image classification tasks.

Acknowledgment

We would like to thank Prof. Catia Silva for providing this oppurtunity to do this exciting exercise of image classification. We would also like to thank numerous TAs and hipergtaor team for providing us the valuable inputs and computing resources .

References

[1] Gour, M., Jain, S., & Sunil Kumar, T. (2020). Residual learning based CNN for breast cancer histopathological image classification. *International Journal of Imaging Systems and Technology*, *30*(3), 621-635

[2] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778)J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.