American Sign Language Alphabet Classification

A. Introduction, Problem Statement and Application

An estimated 48 million Americans, or 15% of the population, suffer from hearing loss. There is increasing interest in having A.I. systems take into account the needs of the disabled, particularly those who have hearing impairments [4]. The deaf and hard-of-hearing community uses sign language to communicate and interact with the outside world. Recognition of sign language has long been a significant area of study [1]. However, due to the limited availability of interpreters and the lack of knowledge of sign language among the general population, effective communication becomes challenging. Speaking with people who use sign language in real life might be challenging. Several ASL models in use today only have an accuracy of about 60% because of hand shape, mobility, and occlusions. In American sign language (ASL), proper nouns like names, technical terms, abbreviations, or foreign phrases are often spelt using one of the 26 unique hand movements that represent the letters of the alphabet (from A to Z). ASL is used predominantly in the United States and in many parts of Canada. ASL is accepted by many high schools, colleges, and universities in fulfillment of modern and “foreign” language academic degree requirements across the United States [7]. This project aims for ASL alphabet classification to facilitate better understanding and interaction with the ASL community. The goal of the research is to examine the various deep learning models in order to identify the model that is most optimised for handling the classification challenge. The goal is to develop a reliable and efficient tool for ASL alphabet recognition, which can bridge the communication gap between individuals with hearing impairments and those who do not know sign language. The proposal showcases the importance of the problem, the dataset selection, possible methodology, and the expected results.

B. Proposed Methodology

Perform necessary prepossessing steps on the dataset to increase the robustness, which includes augmentation techniques like resizing to 256 x 256, horizontal flip, and noise reduction with the help of Gaussian blur with a radius of 2 . The dataset is divided into split is 80% for training , 10% for testing and 10% for validation. The goal is to use supervised and semi-supervised learning classifications on the ASL alphabets.

The supervised methodology for decision tree classification involves using preprocessed images as input features and image labels as the target variable. The model’s performance is evaluated on the testing set using assessment metrics like accuracy, precision, recall, F1-score, and confusion matrix analysis. These metrics provide insights into the model’s effectiveness and generalizability. Additionally, the impact of dimensionality reduction through PCA can be assessed using the same evaluation criteria. PCA helps reduce dimensionality while retaining important features, enhancing efficiency and potentially improving model performance. Fine-tuning of the model can be done based on the evaluation results, allowing for informed decisions about performance improvement.

For Semi-supervised Learning Classification with decision trees involves a two-step process. Firstly, the Decision Tree classifier is trained using labeled data. Secondly, the predicted labels from the unlabeled data are combined with the labeled data to create an augmented dataset. Techniques like GridSearchCV are used to fine-tune the hyperparameters and optimize the model’s performance. The model is then evaluated on the testing set using various evaluation metrics. Additionally, Principal Component Analysis (PCA) is applied for dimensionality reduction. The evaluation metrics provide insights into the model’s accuracy, precision, recall, F1-score, and its ability to handle ASL alphabet recognition tasks effectively. This allows for informed decisions regarding the model’s performance and potential improvements.

For Supervised Learning Classification with DNN, the proposed methodology involves using the ASL dataset, which consists of preprocessed images representing ASL signs. The dataset is then divided into separate sets for training, validation, and testing. To classify the ASL signs, the ResNet-18 architecture is employed, leveraging its residual blocks composed of multiple convolutional layers. The model’s performance is assessed by evaluating its training accuracy, test accuracy, precision, recall, and F1-score. The utilization of deep neural networks like ResNet-18 allows for the capture of complex patterns and more precise predictions particularly when handling high-dimensional data. Overall, this proposed methodology involves preprocessing the ASL dataset, training the ResNet-18 model on the training set using appropriate parameters, and evaluating its performance using various metrics. This approach is expected to yield superior accuracy in ASL alphabet recognition compared to decision tree-based models.

C. Attempts at solving the Methodology

Data Prepossessing: Initially we decrease the image size to 256x256 from 400x400. After scaling the images, using a Gaussian blur with a radius of 2, the amount of image noise is decreased.In addition, gray scale conversion, scaling of the pixel values, and noise removal are carried out. After that, for further use, we keep both the labels and the processed photos.

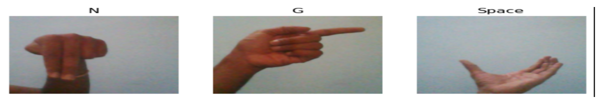


Figure 1. Before Processing

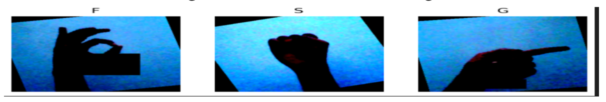


Figure 2. After Processing

C.1. Supervised learning with Decision Trees

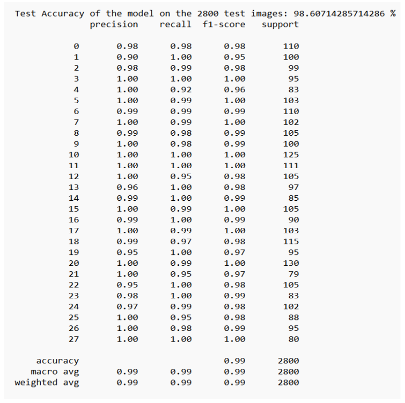
In our methodology, we trained a Decision Tree classifier using preprocessed images and corresponding ASL alphabet labels. Hyperparameter tuning was performed using GridSearchCV, exploring different combinations of hyperparameters like max-depth, Gini index, and entropy. We found that the optimal depth of the tree was between 8 to 12, as increasing depth led to overfitting. This approach resulted in promising results, with an accuracy of approximately 88% achieved after a CPU training that took 16 hours. To further improve performance and reduce training time, we implemented Principal Component Analysis (PCA) for dimensionality reduction. This technique reduced the dimensionality of the image data while retaining important features. By feeding the model with the reconstructed images, we achieved better performance with 92% accuracy and a 50% reduction in training time.

C.2. Semi-supervised learning with Decision Trees

In our semi-supervised learning methodology, we first divided the dataset into labeled and unlabeled data. The labeled data contained instances with their ASL alphabet labels, while the unlabeled data consisted of images without any assigned labels. We trained an initial Decision Tree classifier using the labeled data, which made predictions based on the provided labels. The labeled data was used only 20% for training. In the iterative training phase, we combined the labeled data with the predicted labels for the unlabeled data, creating an augmented dataset. GridSearchCV was employed to fine-tune the hyperparameters, such as max-depth, Gini index, and entropy, to optimize the model's performance. Using the augmented dataset and the optimized hyperparameters, we trained a new Decision Tree classifier iteratively, refining the predictions based on the combined labeled and predicted labels. Additionally, Addition of PCA on the augmented dataset to reduce its dimensionality while preserving important features.

C.3. Supervised learning Classification with a DNN model

We used ResNet 18 which is built using residual blocks, which are modules composed of multiple convolutional layers. During the training process, the model used a learning rate of 0.0001, a batch size of 32, cross-entropy loss as the loss function, and the Adam optimizer. The model was run for 10 epochs With these specifications, we were able to achieve training accuracy of 99.1% and test accuracy of 98.6% (The model Precision is 0.99 , Recall is 0.99 and F1-Score is 0.99). With CNN model, we observed that we were able to achieve more accuracy(99.1%) than decision trees(92%), as neural networks can capture complex patterns and make more precise predictions in various tasks as compared to decision trees which are limited by their shallow structure which causes issue in handling high dimensional data.



```
Test Accuracy of the model on the 2800 test images: 98.60714285714286 %
              precision    recall  f1-score   support

           0       0.98      0.98      0.98       110
           1       0.90      1.00      0.95       100
           2       0.98      0.99      0.98        99
           3       1.00      1.00      1.00        95
           4       1.00      0.92      0.96        83
           5       1.00      0.99      1.00       103
           6       0.99      0.99      0.99       110
           7       1.00      0.99      1.00       102
           8       0.99      0.98      0.99       105
           9       1.00      0.98      0.99       100
          10       1.00      1.00      1.00       125
          11       1.00      1.00      1.00       111
          12       1.00      0.95      0.98       105
          13       0.96      1.00      0.98        97
          14       0.99      1.00      0.99        85
          15       1.00      0.99      1.00       105
          16       0.99      1.00      0.99        90
          17       1.00      0.99      1.00       103
          18       0.99      0.97      0.98       115
          19       0.95      1.00      0.97        95
          20       1.00      0.99      1.00       130
          21       1.00      0.95      0.97        79
          22       0.95      1.00      0.98       105
          23       0.98      1.00      0.99        83
          24       0.97      0.99      0.98       102
          25       1.00      0.95      0.98        88
          26       1.00      0.98      0.99        95
          27       1.00      1.00      1.00        80

    accuracy                           0.99      2800
   macro avg       0.99      0.99      0.99      2800
weighted avg       0.99      0.99      0.99      2800
```

Figure 3. Accuracy plot with DNN

D. Future plans and improvement

By combining these improvements, including model comparison, hyperparameter tuning, GPU acceleration, and deployment on AWS EC2 instance to enchance the computation power, we expect to achieve higher accuracy, faster execution, and improved real-time performance for our ASL alphabet classification project. The visualizations generated during the analysis will provide valuable insights into the effectiveness of these improvements and will be included in our final report. Evaluating the metrics like precision, recall, F1-score for the decision trees. Further for CNN model, to improve accuracy we are planning adjusting batch size and learning rate, increasing the number of epochs, utilizing t-SNE for visualization, and implementing feature engineering techniques to reduce the number of input features (e.g., pixels). These strategies aim to enhance the model's performance and optimize its ability to extract meaningful patterns from the data. The visualizations generated during the analysis will provide valuable insights into the effectiveness of these improvements and will be included in our final report.

Bibliography

[1] Asl alphabet recognition. https://pubmed.ncbi.nlm.nih.gov/34502747//.

[2] Hyperparameter. https://www.section.io/engineering-education/hyperparmeter-tuning/.

[3] Machine learning. https://www.simplilearn.com/tutorials/machine-learning-tutorial/principal-component-analysis.

[4] The medium article. https://medium.com/mlearning-ai/american-sign-language-alphabet-recognition-ec286915df12.

[5] Optimize hyperparameters. https://www.projectpro.io/recipes/optimize-hyper-parameters-of-decisiontree-model-using-grid-search-in-python.

[6] Pytorch tutorial: Torchvision object detection finetuning tutorial. https://pytorch.org/tutorials/intermediate/torchvision_tutorial.html.

[7] Title of the arxiv pape. https://www.nad.org/resources/american-sign-language/what-is-american-sign-language/.

[8] CuPY. Cupy. https://pypi.org/project/cupy/.

[9] Kapillondhe. American sign language. https://www.kaggle.com/datasets/kapillondhe/american-sign-language.

[10] Scikitlearn. Random forest. https://scikitlearn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.

Supplementary Material

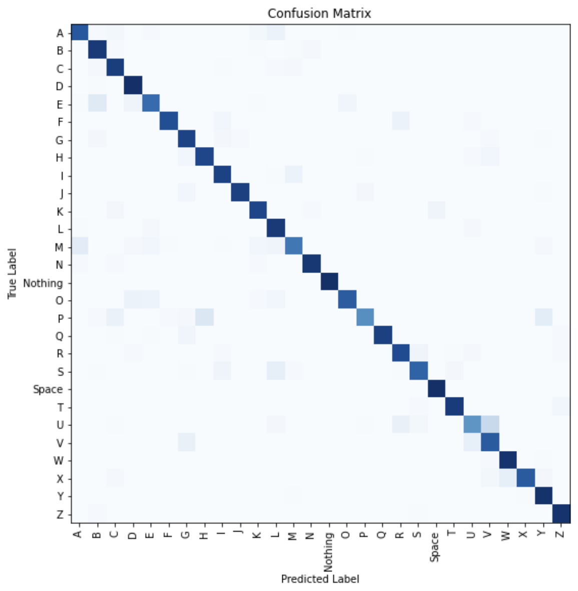


Figure 4. Confusion Matrix for Decision Tree

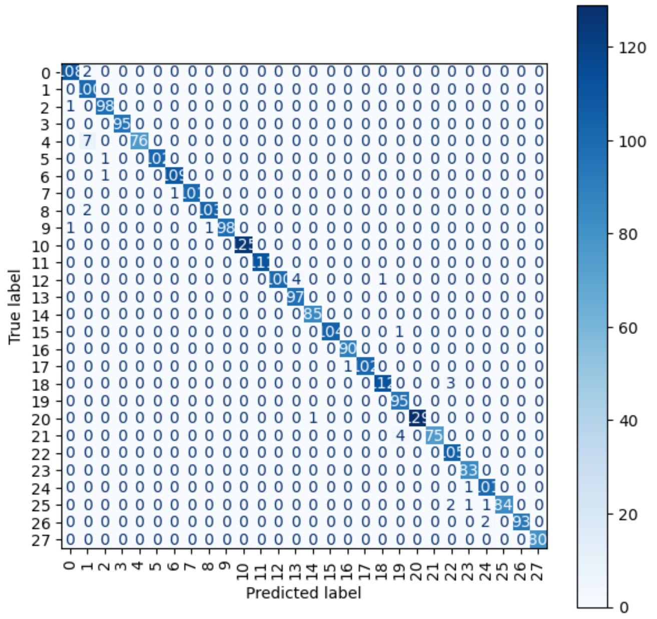


Figure 5. Confusion Matrix for DNN

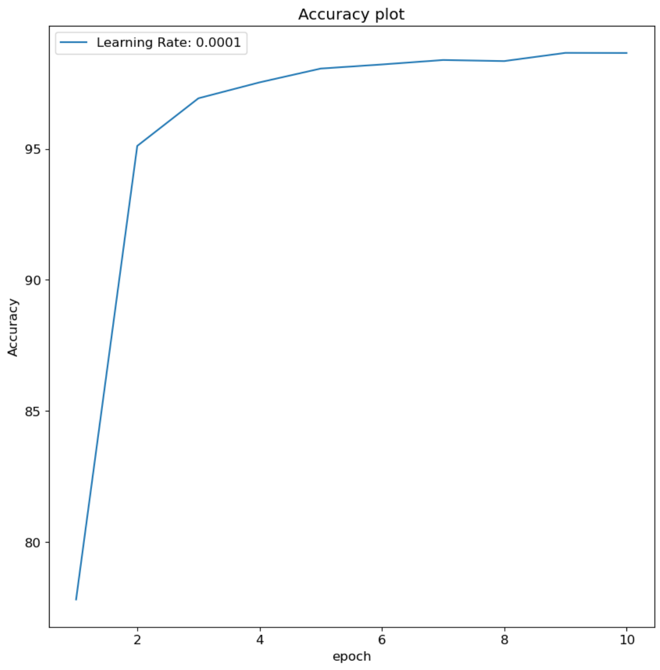


Figure 6. Accuracy plot with DNN