Advancing Sign Language Recognition: A Comprehensive Study on Machine Learning Models for American and Indian Sign Languages

Kunjan Vaghela
Rutgers University
kunjan.vaghela@rutgers.edu

Manad Desai
Rutgers University
md1713@scarletmail.rutgers.edu

Saurabh Kamble
Rutgers University
sk2675@scarletmail.rutgers.edu

Sanket Dalvi
Rutgers University
sd1482@scarletmail.rutgers.edu

Swapnil Verlekar
Rutgers University
sv725@scarletmail.rutgers.edu

ABSTRACT

Sign Language Recognition (SLR) plays a vital role in enabling communication for the deaf and hard-of-hearing, using machine learning to interpret American Sign Language (ASL) and Indian Sign Language (ISL). This study delves into the intricate challenges of deciphering dynamic hand gestures in ASL and ISL, conducting a comprehensive comparative analysis of various machine learning models. Evaluating techniques like K-Nearest Neighbor (KNN), Support Vector Machine (SVM), DeepASLR, and Transfer Learning, the research explores their strengths and limitations. Leveraging augmentation strategies to reinforce datasets, insights emerge from ASL's smaller dimension (28, 28, 1) with 27,455 training samples, contrasting with ISL's larger (400, 400, 3) dataset of 13,689 samples. Results highlight the effectiveness of different models and transfer learning using pre-trained VGG16 and InceptionV3, enhancing accuracy but facing challenges in mitigating misclassifications among visually similar gestures. These findings emphasize diverse datasets, augmentation methods, and model selection, fostering technological inclusivity for the deaf and hard-of-hearing community.

KEYWORDS

KNN (K-Nearest Neighbor), SVM (Support Vector Machine), DeepASLR, Transfer Learning, SLR (Sign Language Recognition), ASL (American Sign Language), ISL (Indian Sign Language)

1 INTRODUCTION

Our project explores Sign Language Recognition (SLR) for the deaf and hard of hearing, focusing on American Sign Language (ASL) and Indian Sign Language (ISL), highlighting its importance in bridging communication gaps. We study the dynamic, nuanced nature of these languages, which use complex hand gestures to convey a wide range of concepts. The study employs advanced machine learning models like K-Nearest Neighbor (KNN), Support Vector Machine (SVM), DeepASLR, and Transfer Learning with models such as VGG16 and InceptionV3, tailored to address the distinct characteristics of ASL and ISL. Key to our research is the comparative analysis of model performance across ASL's compact dataset and ISL's extensive one. Our findings aim to improve SLR technologies, bridging the communication gap for those relying on ASL and ISL and advancing inclusivity in technological solutions.

Conference'17, Washington, DC, USA

DOI: 10.1145/nnnnnnn.nnnnnnn

2 PROBLEM DESCRIPTION

Sign language is used by deaf and hard-of-hearing individuals as their primary means of communication. Automated sign language recognition could help bridge the communication gap between signers and non-signers. However, existing sign language datasets and models tend to have limited skin tone variation, often focusing only on light skin tones [1].

In this project, we aim to develop a sign language recognition system that works well across skin tones. Our training data will consist of both American Sign Language (ASL) and Indian Sign Language (ISL) images depicting signs performed by signers with light, medium, and dark skin tones.

We trained several models on this multi-ethnicity dataset to recognize signs. During training, we monitored performance across skin tones. The goal is to achieve high accuracy across all skin tones represented in the training data.

After training the model, we tested it on a holdout test set that also contained varied skin tones. This allowed us to evaluate real-world performance for signers of different ethnic backgrounds. Metrics like accuracy, precision, and recall will be tracked separately for light, medium, and dark skin tones.

In summary, we aim to develop an automated sign language recognition system that achieves both high overall performance and fairness across ethnicities. The multi-ethnicity training data and skin-tone specific evaluation are key to avoiding biased models.

3 DATASET

3.1 ASL - American Sign Language

The American Sign Language (ASL) dataset utilized in this research encompasses alphabet classes from A to Z, excluding J and Z due to their reliance on hand movements. This dataset, which is made up of 7,172 test cases and 27,455 training cases, is displayed as

grayscale images with an aspect ratio of (28, 28, 1) that depict ASL hand gestures. The dataset shows different class distributions, and in order to improve model training and accuracy, data augmentation techniques were used to augment the dataset.

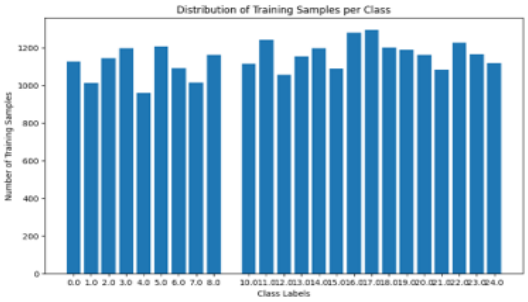


Figure 1: ASL classwise data distribution

Figure 2: ASL classwise data sample

3.2 ISL - Indian Sign Language

The Indian Sign Language (ISL) dataset that was used in this research includes numerical values ranging from 0 to 9 and alphabets from A to Z. 13,689 examples for training, 1,502 examples for validation, and 1,486 examples for testing make up this extensive dataset. A coloured image with dimensions of (400, 400, 3) is displayed for each data instance, containing the visual representations of alphanumeric characters and ISL gestures.

The dataset displays discernible variations in the distribution of data instances among its classes, signifying varying frequencies or occurrences of each class in the dataset. Data augmentation techniques were used to improve the efficacy of model training and address potential class imbalance. With the help of these augmentation techniques, which involved transforming the original ISL images in different ways, including rotation, scaling, and flipping, the dataset was increased, guaranteeing a more complete and varied training set.

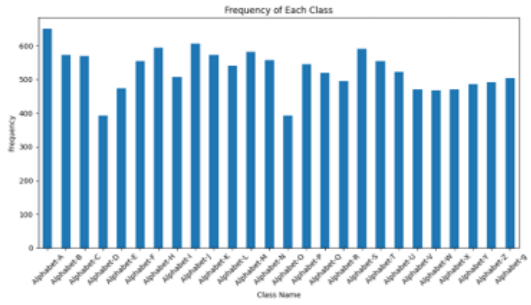


Figure 3: ISL classwise data distribution

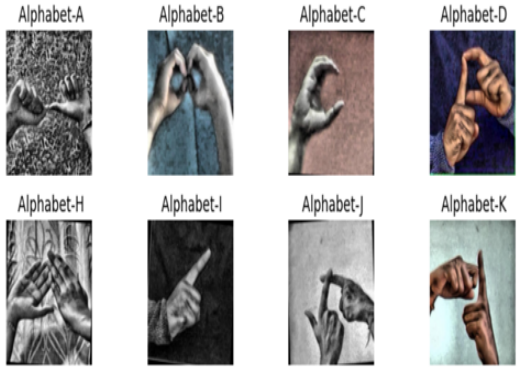


Figure 4: ISL classwise data sample

4 METHODOLOGY OF OUR WORK

We implemented Sign Language detection for ASL and ISL using various machine learning methodologies for better analysis.

4.1 KNN

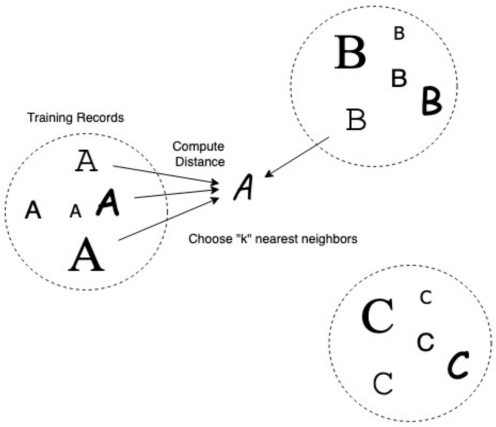
The approach utilizes a K-Nearest Neighbors (KNN) classifier to perform sign language recognition on the ASL and ISL dataset. This dataset consists of images of hands forming various sign language letters and numbers.

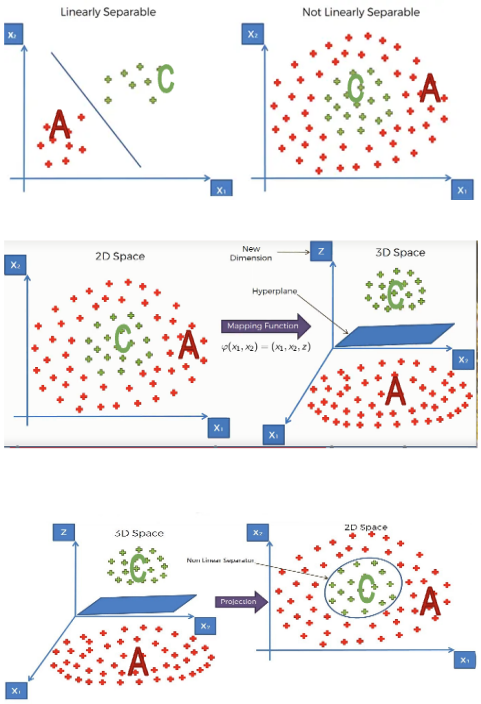
The sign images are pre-processed by resizing and flattening into one-dimensional feature vectors. These raw pixel features for the training data are then min-max scaled to normalize the value range.

A KNN model is initialized and fit on the processed training data, with 5 neighbors and using a Euclidean (L2 norm) distance metric. This allows finding the 5 most similar training images for any new test image.

For evaluating the method, additional test sign image data is procured. This data undergoes the same preprocessing and scaling pipelines to achieve compatibility with the trained KNN classifier.

The trained model is then used to predict labels for the test data. Various evaluation metrics are computed by comparing predictions





to the ground truth labels, including accuracy, precision, recall and F1 score.

Additionally, test set performance is monitored on factors like skin tone to detect any biases. Fairness is encouraged through the multi-ethnicity SIGN dataset containing high diversity.

In summary, this KNN classification approach combined with scaled raw pixel features aims to accurately recognize sign language gestures while maintaining unbiased predictions across demographics.

4.2 SVM

The application of Support Vector Machine (SVM) for sign language recognition involves leveraging the ASL and ISL datasets, comprising hand gesture images representing various sign language letters and numbers. Similar to KNN, the sign images undergo preprocessing steps. They are resized and flattened into one-dimensional feature vectors. These raw pixel features undergo min-max scaling, ensuring a normalized value range, enabling effective SVM training. The SVM model is constructed and trained on the preprocessed data. Utilizing both linear and non-linear kernels, SVM aims to define an optimal hyperplane for class separation in the high-dimensional feature space.

Hyperparameter tuning is conducted, exploring the parameter space via techniques like GridSearchCV to identify the most suitable hyperparameters for optimal SVM performance. Following training, the model is evaluated using additional test sign images subjected to the same preprocessing and scaling pipelines. Evaluation metrics, including accuracy, precision, recall, and F1 score, are computed to assess the model's performance.

The SVM model's performance on the test set is scrutinized for biases related to factors such as skin tone. The aim is to ensure fairness across demographics, aligning with the multi-ethnicity focus of the SIGN dataset. The SVM classification method, with its ability to define effective separation boundaries, strives to accurately recognize sign language gestures. It aims to maintain unbiased predictions and cater to diverse demographics, contributing to equitable communication accessibility for all.

4.3 DeepASLR

A Convolutional Neural Network (CNN) is a widely used deep learning approach for analyzing visual data, particularly images. It requires minimal preprocessing and automatically learns filters, reducing the need for manual engineering. The CNN consists of four key operations: convolution, pooling, flattening, and fully connected layers.

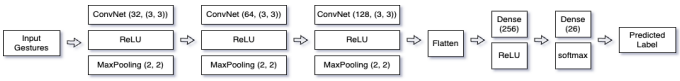
Convolution Layer: Captures low-level features like color, edges, and gradient orientation.

Pooling Layer: Reduces spatial dimensions, aiding computational efficiency and preserving dominant features.

Flattening: Converts processed image into a 1-D vector for further analysis.

Fully Connected Layers: Follow flattening, aiding in feature extraction for classification. ReLu activation functions and SoftMax output layers are commonly used.

In our exploration of CNN models for Sign Language detection, we chose to evaluate the DeepASLR model due to its impressive accuracy. DeepASLR, a CNN-based model, exhibited outstanding performance across three different datasets, achieving an average accuracy of approximately 99.42% during testing. This exceptional accuracy surpassed that of previous Sign Language detection models, highlighting the effectiveness of the DeepASLR CNN model in achieving high-performance results. Architecture of DeepASLR:



- Input layer: Accepts images with dimensions 64 × 64 × 3 (representing sign language frames).
- Feature extraction: Three convolutional layers (Conv1, Conv2, Conv3) with 3 × 3 filter dimensions. Filter numbers are 32, 64, and 128 for ConvNet1, ConvNet2, and ConvNet3, respectively. Each convolution is followed by a rectified linear unit (ReLu) activation.
- MaxPooling: Applied with 2 × 2 dimensions after ReLu to prevent information loss.
- Flattening: Transforms features for the classification stage.
- Classification: Implemented with fully connected layers, ReLu activation, and a SoftMax output layer for probability-based object predictions.

The system is implemented using Keras, Tensorflow, and OpenCV libraries. The proposed CNN structure is designed to effectively process sign language frames, utilizing convolutional and pooling layers for feature extraction and classification, demonstrating the power of deep learning in visual analysis.

4.4 Transfer Learning

Transfer learning involves adopting a pre-trained machine learning model, originally designed to tackle a specific problem, and applying it to a related but distinct task. For instance, the insights gained from training a basic classifier to identify backpacks in images could be utilized to recognize other items like suitcases.

In computer vision, neural networks commonly identify edges in lower levels, shapes in middle layers, and task-specific features in upper layers. Transfer learning focuses on reusing the early and middle layers, while retraining only the later layers. This method utilizes labeled data from the initial training of the model.

One of the key advantage of transfer learning is its ability to save time in training, enhance the performance of neural networks in most cases, and require less data. Typically, training a neural network from scratch demands a substantial amount of data, which might not always be accessible. This is where transfer learning proves useful—enabling the construction of robust machine learning models with comparatively smaller training datasets due to the pre-trained nature of the model.

This approach is especially valuable in natural language processing, where creating vast labeled datasets often requires expert knowledge. Moreover, it significantly reduces training time, which could otherwise span days or weeks, particularly when training deep neural networks on complex tasks.

The core principle of transfer learning involves leveraging a network's learned weights from one task (referred to as "task A") to enhance performance in a different task (termed as "task B"). Its primary aim is to utilize the knowledge gleaned from a task that boasts a large labeled training dataset to improve generalization in a separate task with limited data. Rather than starting from scratch, transfer learning capitalizes on previously learned patterns from a related or similar task to expedite the learning process. Applications of transfer learning find widespread use in domains like computer vision and sentiment analysis within natural language processing, particularly benefiting tasks that demand significant computational resources.

In this project, we are trying to leverage the pre-trained model of VGG16 and InceptionV3 to our dataset. The pre-trained models are originally trained on imagenet**??** dataset which contains 14M images and around 1000 classes.

4.4.1 VGG16. The VGG16 convolutional neural network architecture are made up of three fully connected layers after 13 convolutional layers. With its consistent architecture and simple 3x3 convolutional filters used across the network, VGG16 is well known for its simplicity. Because of the way it is made, it can identify complex patterns and characteristics in pictures.

There are VGG16 pre-trained models available that were trained on massive image datasets such as ImageNet. In computer vision problems involving transfer learning, these pretrained models are useful. These pretrained models, which make use of the learnt representations from VGG16's training on ImageNet, provide a basis for a variety of image recognition tasks, enabling quicker convergence and enhanced performance in novel image classification or feature extraction tasks.

In this project, we have exculded the last output layer of the VGG16 and added our output layer for the classification. In this project, we are not freezing the initial layers of VGG16, but we are fine-tuning it on our dataset for the better classification accuracy.

4.4.2 InceptionV3. InceptionV3 is a convolutional neural network architecture developed by Google's research team. Its deep architecture and intricate design are what define it; they are intended to achieve high accuracy with computational efficiency. InceptionV3's main innovation is how it uses "Inception modules," which are blocks with different convolutions of different sizes (1x1, 3x3, 5x5) and pooling operations. This allows the network to capture a variety of features at different scales within the same layer. Compared to previous models, this architecture maintains or even improves performance while drastically reducing the computational load. Deeper networks are made possible without appreciably raising computational requirements by factorization techniques like 1x1 convolutions, which lower computational complexity and dimensions.

We used the InceptionV3 architecture in our project to accomplish our goal. Rather than leaving the final layer in place, we eliminated it and adjusted the model so that it could be trained on our dataset to acquire task-specific features. Importantly, during this fine-tuning phase, we decided not to freeze the InceptionV3 network's initial layers. The model could adjust and update its learned representations from earlier layers to better fit the intricacies of our particular task by unfreezing these initial layers, which would improve performance and adaptability. By using this approach, the network was able to improve its comprehension of the data while applying the pre-trained knowledge derived from the architecture of InceptionV3.

5 RESULTS

5.1 KNN

(1) KNN Accuracy comparison for varying K on ASL and ISL dataset.

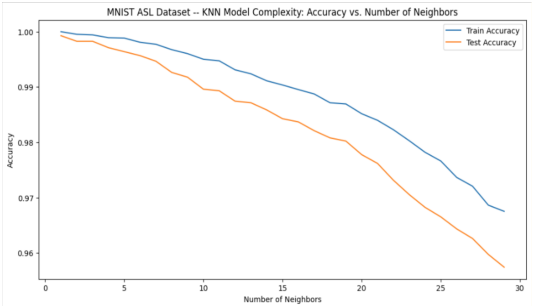


Figure 5: KNN Accuracy for varying K for ASL dataset

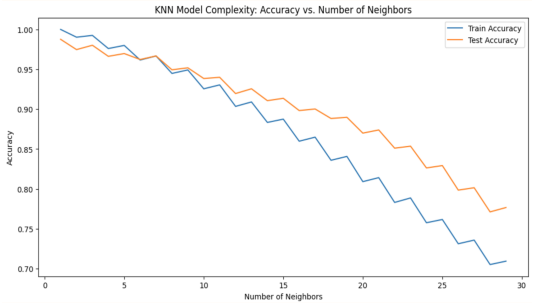


Figure 6: KNN Accruancy for varying K for ISL dataset

The optimal K value balances model accuracy and generalization capability. Smaller K values of 5-7 achieved maximal accuracy for ASL (99.64%) and ISL (98.41%) but risk overfitting. While ASL maintained accuracy up to K=10, ISL performance fluctuated sharply with K. Thus, K=7 strikes the best fit, converging train and test accuracy for ISL while retaining strong ASL performance. Additionally, the test accuracy for ISL dataset surpasses train accuracy K=7 indicating overfitting due to smaller dataset size. Overall, K value selection requires balancing precision and avoidable overfitting through the accuracy trend across K.

(2) KNN Precision and Recall comparison for ASL and ISL dataset.

The KNN model achieved strong overall precision and recall for both ASL and ISL recognition. However, performance dropped for certain easily confusable signs with similar gestures - ASL alphabets U, V, W and ISL digits

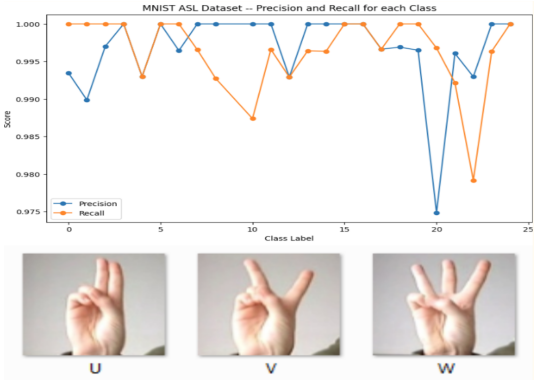


Figure 7: Precision and Recall for all classes for ASL dataset

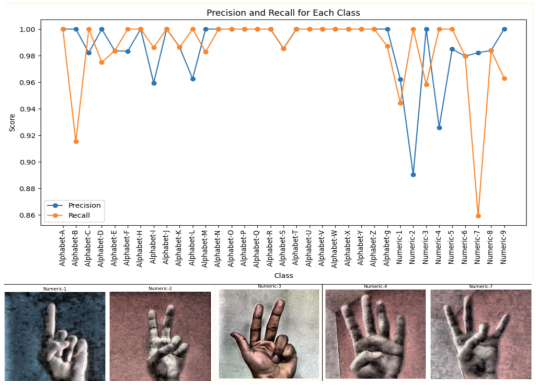


Figure 8: Precision and Recall for all classes for ISL dataset

1, 2, 3, 4, 7. Similarity in the hand shapes for these signs increases misclassifications. Expanding the diversity and size of the training data for these classes could mitigate such errors. Nonetheless, the high precision and recall demonstrates robust sign recognition capabilities, despite intrinsic challenges in for certain subtle sign differences.

(3) Impact on KNN accuracy based on dataset size for ASL and ISL dataset.

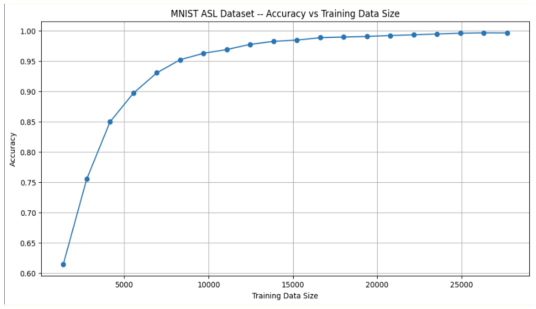


Figure 9: Training accuracy v/s dataset size for ASL dataset

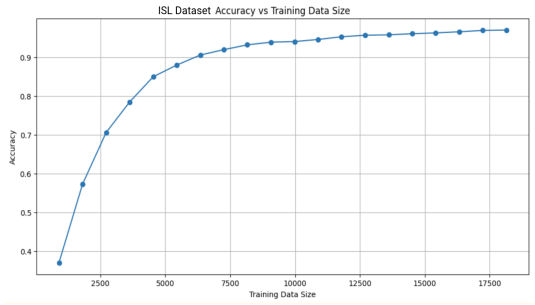


Figure 10: Training accuracy v/s dataset size for ISL dataset

The KNN model accuracy improves with more training data, as additional examples enhance representation of the target class distribution. For both ASL and ISL test sets, accuracy increased monotonically up to respective peaks utilizing full dataset sizes of 20k and 12.5k samples, affirming larger data benefits model performance. However, beyond these points, additional data provided minimal accuracy gains, indicating saturation where further samples add little new information. High ultimate accuracy levels nearing 99% (ASL) and 98% (ISL) signify the model's ability to generalize reliably for sign recognition without requiring ever-larger quantities of data.

5.2 SVM

(1) Observed results for SVM model on ASL and ISL dataset.

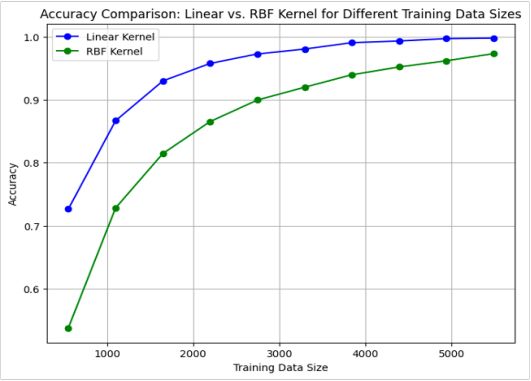


Figure 11: Linear vs RBF Kernel for different training dataset sizes.

Advantages and Disadvantages Reflected in Results
The SVM model's advantages are evident in its ability to achieve high accuracy and robustness in recognizing sign language gestures across different datasets. However, challenges persist in precision and recall for signs with subtle similarities, suggesting a need for continual improvement

Table 1: Score for SVM models

	ASL Linear Kernel	ASL RBF Kernel	ISL Linear Kernel	ISL RBF Kernel
Accuracy	99.79%	97.34%	96.67%	98.06%
Precision	99.79%	97.35%	96.75%	98.13%
Recall	99.79%	97.32%	96.67%	98.06%
F-score	99.79%	97.33%	96.67%	98.06%

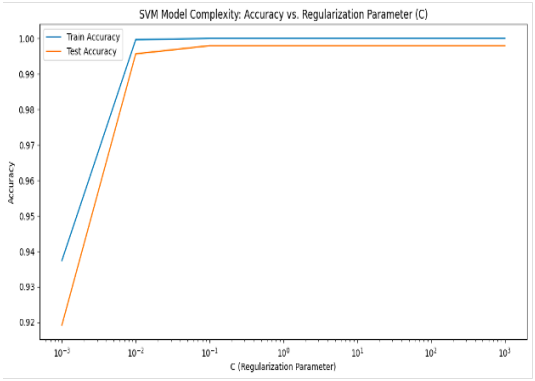


Figure 12: Training and testing accuracy based on Regularization Parameter value (C)

in training data diversity and size for enhanced accuracy in recognizing such gestures. The model showcases efficiency in generalization without an insatiable need for an exceedingly large dataset, affirming its practicality for real-world applications.

5.3 DeepASLR model

(1) ASL dataset

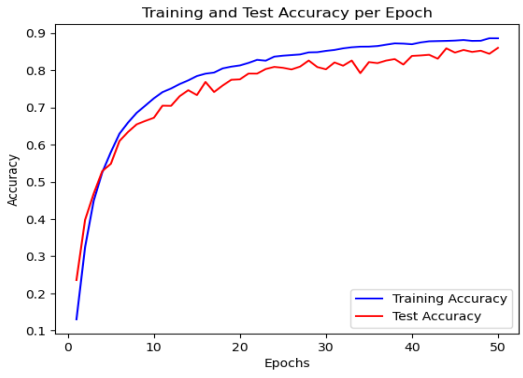


Figure 13: Training vs Test Accuracy for DeepASLR for ASL dataset

Figure 14: Training vs Test Loss for DeepASLR for ASL dataset

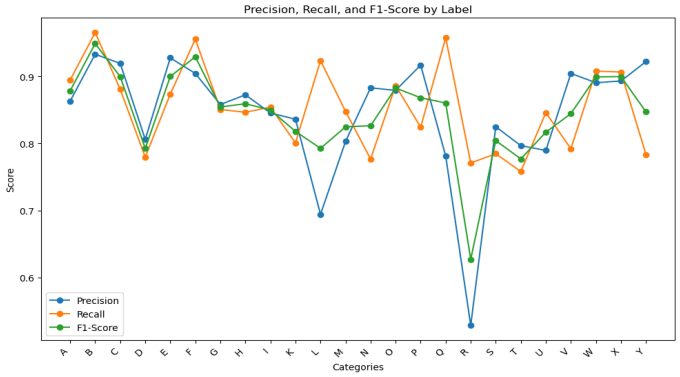


Figure 16: Performance Metrics for DeepASLR for ASL dataset

The DeepASLR model, applied to the ASL dataset, achieves an overall accuracy of 85%. It demonstrates diverse precision, recall, and F1 scores across classes, with individual class performances ranging in precision from 0.47 to 0.96, in recall from 0.75 to 0.96, and in F1 score from 0.59 to 0.93. The model exhibits a balanced trade-off between precision and recall, shedding light on its strengths and areas for improvement in recognizing various American Sign Language gestures.

It's noteworthy that the DeepASLR Model required 50 epochs to achieve an accuracy of 85%, whereas other models achieved significantly higher accuracies, exceeding 97%, within only 25 epochs on the same dataset. This difference in performance may be attributed to the identification of fewer trainable parameters during the training of the CNN model for this particular dataset.

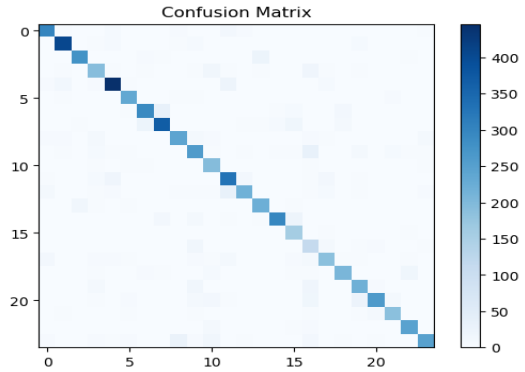


Figure 15: Confusion Metrics for DeepASLR for ASL dataset

Upon reviewing the Performance Metrics (Figure 15) and Confusion Metrics (Figure 16) for individual classes, it becomes evident that certain classes, such as R, exhibited lower performance compared to others. This could be attributed to the higher visual complexity involved in distinguishing the sign for R from other gestures, such as D and L.

Similarly, some classes, such as B and F, demonstrated outstanding performance compared to others. This can be attributed to the distinctiveness of their gestures when compared to signs for the remaining classes.

(2) ISL dataset

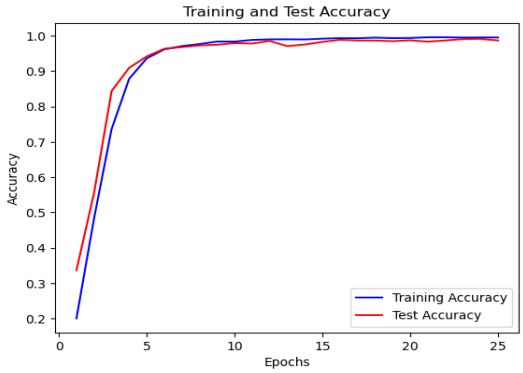


Figure 17: Training vs Test Accuracy for DeepASLR for ISL dataset

In the evaluation of the ISL dataset using the DeepASLR model (refer to Figure 17 and Figure 18), substantial advancements in performance metrics are evident compared to the ASL dataset. The model achieves an impressive overall accuracy exceeding 98%, underscoring its adeptness in recognizing Indian Sign Language gestures. Precision, recall, and F1 scores consistently approach or reach their maximum values, indicating a robust ability of the DeepASLR model to recognize diverse Indian Sign Language

Figure 18: Training vs Test Loss for DeepASLR for ISL dataset

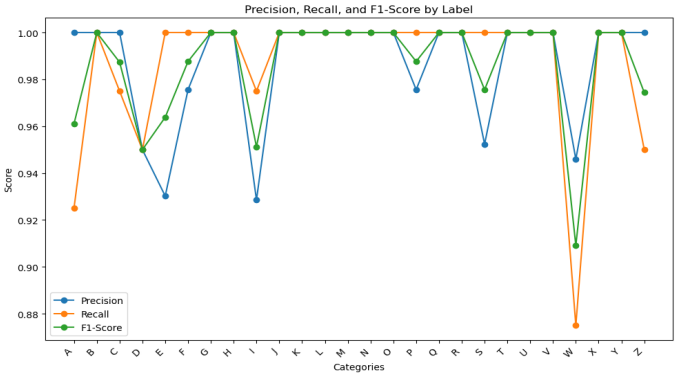


Figure 20: Performance Metrics for DeepASLR for ISL dataset

gestures. Across individual classes, the model exhibits precision ranging from 0.93 to 1.00, recall from 0.88 to 1.00, and F1 scores from 0.91 to 1.00.

Notably, DeepASLR achieves this exceptional performance within a mere 25 epochs, showcasing its efficiency in learning and adapting to the nuances of the ISL dataset, a notable improvement compared to the ASL dataset.

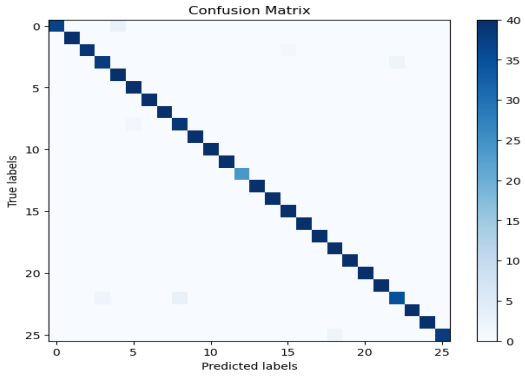


Figure 19: Confusion Metrics for DeepASLR for ISL dataset

In contrast to the ASL dataset, where certain classes like R displayed lower performance due to visual complexity, the ISL dataset consistently showcases high performance across all classes (refer to Figure 19 and Figure 20). This consistency suggests a dataset-specific adaptability of the DeepASLR model to the unique visual characteristics of Indian Sign Language. Additionally, a noteworthy aspect is the presence of double-handed gestures in the ISL dataset, contributing to a higher visual diversity in deciphering hand gestures. This feature likely contributes to the elevated performance of the ISL dataset on the DeepASLR model compared to the ASL dataset.

In summary, the DeepASLR model demonstrates exceptional performance on the ISL dataset, achieving an accuracy exceeding 98%. The comparison with the ASL dataset underscores the model's adaptability to diverse sign languages and implies its effectiveness in capturing language-specific nuances.

5.4 VGG16 and InceptionV3 models

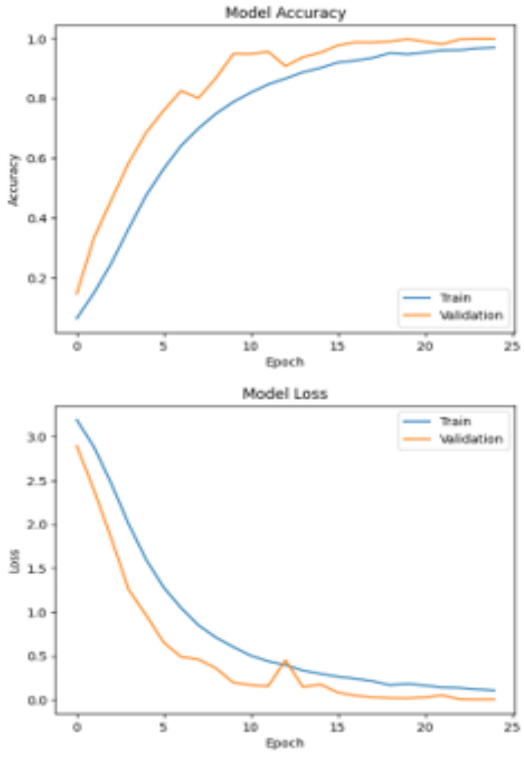


Figure 21: Training and testing Accuracy and Loss for VGG16 on ASL Dataset

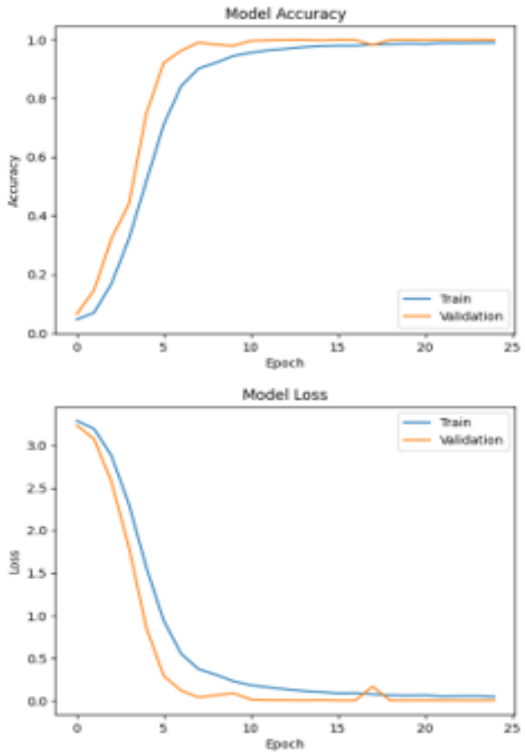


Figure 22: Training and testing Accuracy and Loss for InceptionV3 on ASL Dataset

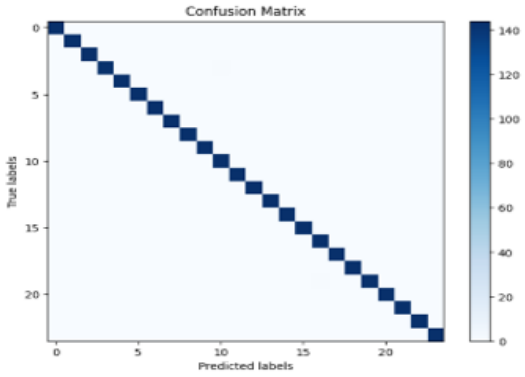


Figure 23: Confusion matrix for VGG16 on ASL Dataset

(1) Result of VGG16 and InceptionV3 model on ASL dataset
The VGG16 and InceptionV3 models perform exceptionally well in our analysis of the ASL dataset, with accuracies as high as 99.9%. But when it comes to precision, recall, and F1-score, InceptionV3 performs better than VGG16. It achieves consistent, higher scores—99.7%—in all three categories. This indicates that even though both models achieve exceptionally high accuracy rates, InceptionV3 is more capable of accurately identifying and classifying

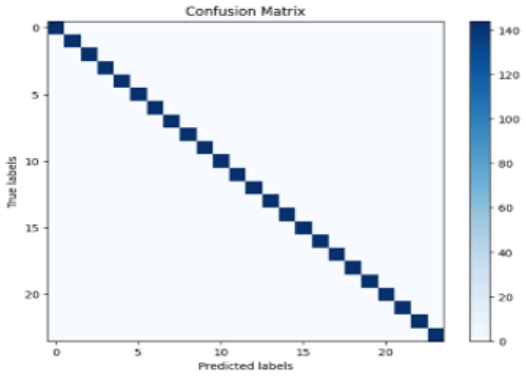


Figure 24: Confusion matrix for InceptionV3 on ASL Dataset

American Sign Language gestures than VGG16. These findings demonstrate InceptionV3's improved recall and precision, suggesting that it may be better in practical uses for ASL recognition tasks.

(2) Result of VGG16 and InceptionV3 model on ISL dataset
Both the VGG16 and InceptionV3 models show good performance metrics when tested on the ISL dataset. Although VGG16 receives praise for its performance in terms of accuracy, precision, recall, and F1-Score, InceptionV3 performs marginally better overall. With a 98.54% accuracy rate and a 98.56%, 98.53%, and 98.53% precision, recall, and F1-Score, respectively, higher than VGG16, InceptionV3 shows a slightly better capacity to identify and classify Indian Sign Language (ISL) gestures. These outcomes highlight the marginally better performance of InceptionV3 and its possible applicability for practical ISL recognition applications.

As evident from the data presented in Figure 28, a detailed examination of the confusion matrix indicates that the VGG16 model exhibits suboptimal performance specifically concerning the classification of two particular classes. Consequently, in response to this identified limitation, our approach involved leveraging the enhanced feature extraction capabilities inherent in the InceptionV3 model. This decision was made based on InceptionV3's architecture, known for its ability to extract a diverse range of intricate features from images. By employing the InceptionV3 model, we aimed to capitalize on its multi-faceted feature extraction mechanism, strategically seeking to enhance the model's capacity to discern and classify intricate nuances within the data.

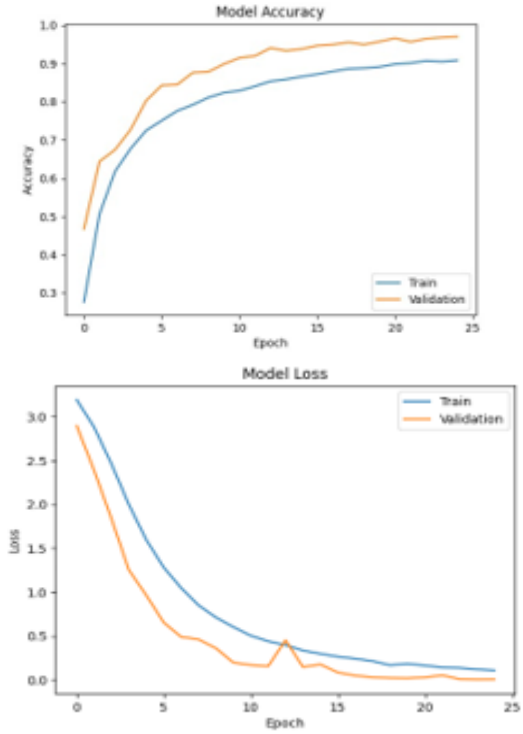


Figure 25: Training and testing Accuracy and Loss for VGG16 on ISL Dataset

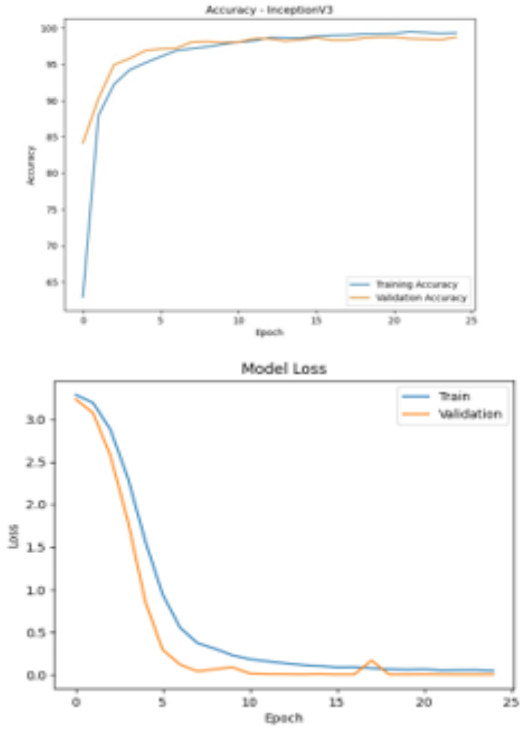


Figure 26: Training and testing Accuracy and Loss for InceptionV3 on ISL Dataset

6 CONCLUSION

In conclusion, our study on Sign Language Recognition (SLR) for American Sign Language (ASL) and Indian Sign Language (ISL) unveils valuable insights into the challenges and advancements in machine learning models. We explored the effectiveness of various models, including K-Nearest Neighbor (KNN), Support Vector Machine (SVM), DeepASLR, and Transfer Learning using VGG16 and InceptionV3.

The KNN model demonstrated high accuracy, precision, and recall, showcasing its capability for sign language recognition. However, challenges arose with visually similar gestures, emphasizing the importance of diverse datasets and continuous improvement.

The SVM model excelled in accuracy and robustness, particularly in recognizing diverse sign language gestures. Despite challenges in precision and recall for certain signs, the model demonstrated efficiency without an insatiable need for an extensive dataset.

DeepASLR, a Convolutional Neural Network (CNN)-based model, exhibited impressive accuracy, reaching 85% for ASL and exceeding 98% for ISL. The model's adaptability to the unique visual characteristics of ISL underscored its effectiveness in capturing language-specific nuances.

Transfer Learning using VGG16 and InceptionV3 yielded outstanding results, with InceptionV3 demonstrating slightly better performance in both ASL and ISL recognition tasks. These pre-trained models showcased their ability to expedite the learning process and achieve high accuracy with limited data.

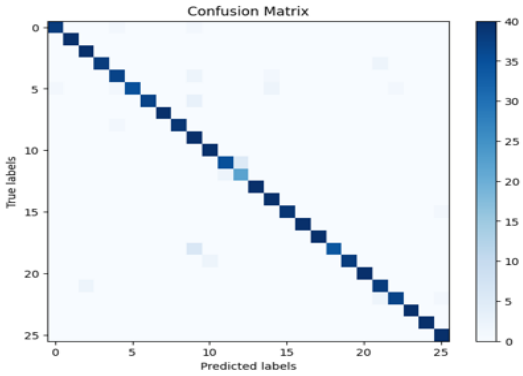


Figure 27: Confusion matrix for VGG16 on ISL Dataset

In summary, our research emphasizes the significance of diverse datasets, model selection, and continuous improvement in advancing technological inclusivity for the deaf and hard-of-hearing community. The findings contribute to the development of effective SLR systems, bridging communication gaps and fostering accessibility for individuals relying on ASL and ISL.

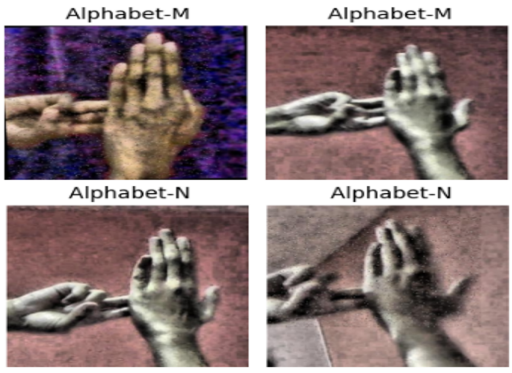


Figure 28: Observation of Alphabate-M and Alphabate-N

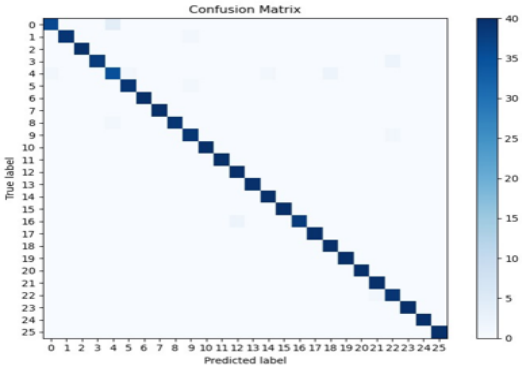


Figure 29: Confusion matrix for InceptionV3 on ISL Dataset

7 REFERENCES

(1) Madhiarasan, M., Roy, P. P. (2022). A Comprehensive Review of Sign Language Recognition: Different Types, Modalities, and Datasets.

(2) ISL Dataset: https://universe.roboflow.com/sanjai-kumar-xexo7/isl-45ew9/dataset/2

(3) Kasapbasi, A., Elbushra, A., Al-Hardanee, O., and Yilmaz, A. Deep-ASLR: A CNN based human computer interface for American Sign Language recognition for hearing-impaired individuals. In Computer Methods and Programs in Biomedicine Update, 2, p.1000482, 2022. https://www.sciencedirect.com/science/article/pii/S2666990021000471

(4) ASL Dataset: https://www.kaggle.com/datasets/datamunge/sign-language-mnist

(5) Sharma, M., Pal, R., Sahoo, A. K. (2014). Indian Sign Language Recognition Using Neural Networks and KNN Classifiers. Department of Computer Science and Engineering, School of Engineering and Technology, Sharda University, Greater Noida, India

(6) Hosna, A., Merry, E., Gyalmo, J. et al. Transfer learning: a friendly introduction. J Big Data 9, 102 (2022). https://doi.org/10.1186/s40537-022-00652-w

(7) Töngi, R. (2021). Application of transfer learning to sign language recognition using an inflated 3D deep convolutional neural network. arXiv preprint arXiv:2103.05111.

(8) Shitole, A., Kulkarni, K., Bathe, P., Patil, R., Deshmane, S. VGG-16 BASED INDO-PAKISTANI SIGN LANGUAGE INTERPRETER.

(9) Mahyoub, M., Natalia, F., Sudirman, S., Mustafina, J. (2023, January). Sign Language Recognition using Deep Learning. In 2023 15th International Conference on Developments in eSystems Engineering (DeSE) (pp. 184-189). IEEE.

(10) Saini, B., Venkatesh, D., Chaudhari, N., Shelake, T., Gite, S., Pradhan, B. (2023, July). A comparative analysis of Indian sign language recognition using deep learning models. In Forum for Linguistic Studies (Vol. 5, No. 1, pp. 197-222).

(11) Bani Baker, Q., Alqudah, N., Alsmadi, T., Awawdeh, R. (2023). Image-Based Arabic Sign Language Recognition System Using Transfer Deep Learning Models. Applied Computational Intelligence and Soft Computing, 2023.