# A Deep Learning–Based Framework for Correcting Erroneous Character-Level Bengali Sign Images

Sameen Islam[1], Mohammed Ayman[2], S.M. Tawsif Islam[3] and Md. Tanzim Reza[4]
[1,2,3,4]Department of Computer Science and Engineering, BRAC University
Kha 224 Pragati Sarani, Merul Badda, Dhaka 1212, Bangladesh
Email: [1]sameen.islam@g.bracu.ac.bd, [2]mohammed.ayman@g.bracu.ac.bd, [3]sm.tawsif.islam@g.bracu.ac.bd, [4]tanzim.reza@bracu.ac.bd

***Abstract*—People who cannot hear or speak rely on sign language as the primary source of communication. Sign language is generally expressed at a fast pace, therefore errors and miscommunication may happen frequently. Our thesis introduces a deep learning–based framework for correcting erroneous character-level Bengali sign language images. The main focus of our research is correctly mapping incorrect bengali character level sign gestures to their closest semantically accurate signs. A dataset consisting of 36 classes for both correct and potential incorrect hand signs were generated for bengali characters. The proposed framework utilizes convolutional neural networks(CNNs) along with triplet loss to extract discriminative embeddings. These embeddings are later represented on a vector space where metric distance from incorrect images are used to map them to the correct gestures. Our primary goal is to develop a system where the intended meaning is correctly delivered even in case of improper hand gestures. Overall, 9 models were used to assess the proficiency of the idea for correcting erroneous bengali sign characters. ResNet50 delivered the most excellent results with an accuracy of 97.6% on a 200 epoch experiment. Further ablation studies also resulted in other models performing well. VGG16 had an accuracy of 94.4% and EfficientNetB0 delivered an accuracy score of 96.4%. The research concluded that longer epochs provided significant increases in accuracy for both black and normal backgrounds. Both KNN and centroid based distance were used as similarity based mapping approaches in order to compare the accuracy of the models under various changes in hyper-parameters. The centroid performed better throughout all the experiments, Thus, the overall results highlight the successful mapping of incorrect sign gestures to their appropriate correct classes.**

***Index Terms*—Triplet Loss, Convolutional Neural Networks, Metric Distance, Embedding**

## I. Introduction

Sign language is a crucial way of communication for people that have hearing or vocal difficulties. By using gestures, facial expressions and body positions, sign language enables a person to express their emotion, communicate their ideas and thoughts. Similar to any language, sign languages have their own grammatical structures and vocabulary. And just like any language, these structures vary depending upon the geographical location and culture. In order to sustain and manage the Bengali deaf community, character-level sign language is crucial and unavoidable. Nonetheless, character-level sign language does pose challenges due to the regional variations. Character-level sign language relies on accurate and quick hand gestures. It has little time for proper implementation in real time. Communication errors may arise due to muscle fatigue, poor interpretation devices and inadequate proficiency in sign language. These errors may lead to losing self-esteem, restricting opportunities and obstacles in other fields requiring communication. Solutions to these problems may arise from technology based applications that facilitate proper translation of these visual gestures.

In our work, we propose an image processing method which will help to identify and correct Bengali character-level sign language errors. Our system will accept input images and generate improved outputs. This will help understand and enhance the accuracy of character-based sign expressions. The model is trained using a dataset where there are multiple character-level sign examples. By training on this dataset, the model will be able to properly detect and correct errors by utilizing adaptive learning algorithms and neural network optimization. Our ultimate goal is to construct a robust and comprehensive system that allows better and proper access to sign language for Bengali-speaking communities. The research can further be expanded beyond the scope of Bengali sign language and delve into a broader research category of assistive technologies. It will promote more inclusivity as enhanced detection and correction of sign languages would lessen the gap in communication between the disabled community. Furthermore, the research and findings of this paper can also serve as a backbone for other similar developments, independent of geographical location. But as our paper focuses on the character-level sign language delivery and correction of Bengali sign language, we are constricted by the limitations of our dataset. Despite the drawbacks of limited data, this work tries to create an effective and efficient system in order to make communication more accessible and correct.

### A. Motivation

Sign language remains to be an important medium for bridging the communication gap between the hearing and speech impaired community. Although there exists sufficient work on Bengali Sign Language, most of the focus is biased towards more widely studied languages like American Sign Language (ASL). Existing research on Bengali Sign Language is predominantly focused on error detection and classification rather than correction. The limited attention for the Bengali Sign Language community encouraged us to deliver our own

contribution towards helping these unfortunate individuals. Our research aims to help these people by developing a translational model for Bengali character-level sign language error correction. Our work will hopefully lay a foundation for further development and attention towards helping the speech and hearing impaired community.

## II. Related Work

Sign language recognition has evolved significantly with the introduction of deep learning architectures that enable representation learning through embeddings. Early work by Rivera-Acosta et al. [1] implemented a real-time ASL translation system combining YOLO-based spatial detection and LSTM-based temporal modeling, demonstrating how hybrid deep architectures can capture both spatial and sequential dependencies. However, their approach lacked metric-based embedding alignment, which limits correction and similarity-based retrieval.

Schroff et al. [2] introduced FaceNet, a landmark model that leveraged triplet loss to learn a compact embedding space for face recognition, where intra-class distances were minimized while inter-class distances were maximized. This principle later inspired sign language systems that map gestures to embedding manifolds, enabling more effective correction of visually similar but semantically different signs. Mopidevi et al. [3] extended this concept with a multiview meta-metric learning framework for sign language recognition, applying triplet loss across multiple camera perspectives to enhance robustness and generalization—an idea closely aligned with our embedding-based Bengali sign correction framework.

Similarly, Gong et al. [4] proposed E2GAN, an efficient image-to-image translation GAN that optimizes embedding consistency while reducing computational cost. Although primarily focused on generative efficiency, their work underscores the importance of preserving semantic fidelity in latent spaces—a concern also addressed by our model through centroid-guided correction. Lin [5] compared Pix2Pix and CycleGAN for image-to-image translation, concluding that supervised GANs like Pix2Pix preserve spatial structure better when paired with aligned datasets, offering insight into potential extensions of our work for synthetic sign data augmentation.

Haque et al. [6] developed a convolutional framework for recognizing Bangladeshi Sign Language (BdSL) words, highlighting the benefits of deep CNNs in extracting fine-grained hand features, yet acknowledging the limitations of purely classification-based models in handling error correction. Raihan et al. [7] introduced Bengali-Sign, a machine learning–based interpretation system emphasizing accessibility but focusing primarily on direct recognition rather than embedding-level mapping. Finally, Natarajan et al. [8] proposed an end-to-end framework combining recognition, translation, and video generation, illustrating the potential of multimodal integration, which motivates future expansion of our embedding-based model toward multimodal correction and synthesis.

Together, these works establish a trajectory from classification-based recognition toward embedding-oriented and translation-aware frameworks. Our approach builds upon this progression by employing triplet loss–driven embeddings and centroid-based correction to enhance recognition accuracy and resilience against visually similar sign confusions.

## III. Methodology

The proposed system is designed to learn discriminative embeddings for Bangla Sign Language (BdSL) gestures at the character level using a triplet-loss-based convolutional neural network (CNN). The model was trained and evaluated on images featuring both standard and black backgrounds, separately, supporting comprehensive assessment across diverse visual conditions.

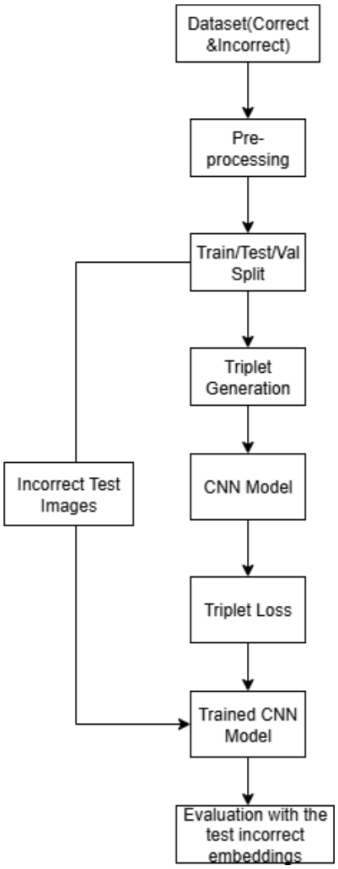


Fig. 1. Data Pre-Process Flowchart

The workflow is organized into three sequential modules: **Data Preprocessing and Organization:** Input images are prepared, augmented, and split into training, validation, and test sets to ensure balanced class representation and robust model learning. **Embedding Model Construction:** Multiple pre-trained CNN backbones—including ConvNeXt-Tiny, DenseNet121, ResNet50, EfficientNetB0, MobileNetV2, VGG16, Xception, CustomCNN and ConvNet—are used to generate compact latent feature representations.
**Metric-Based Evaluation:** Learned embeddings are analyzed

using k-nearest neighbors (k-NN) and centroid-based distance metrics to assess intra-class similarity and inter-class separability.

**Custom CNN:** A custom CNN model was generated to compare how it would perform against pre-trained backbones. The CNN backbone consisted of 7 layers which included 2 Convolutional( 32 and 64), 2 Max pooling, 1 Flatten, 1 Dense and 1 L2 normalization layer.

Each of the imported CNN backbone was loaded with ImageNet weights (excluding top layers) and connected to a dense projection layer of 128 units, followed by batch normalization, dropout (0.3), and L2 normalization. Triplet loss optimized embedding distances. Model training used a batch size of 16 triplets with checkpointing for weight saving on validation loss.The design emphasizes a metric learning paradigm rather than a conventional classification pipeline. Instead of predicting class labels directly, the model learns to project gesture images into an embedding space where distances reflect semantic similarity. This allows incorrect gestures to be mapped to their correct counterparts using geometric relationships in the embedding space.

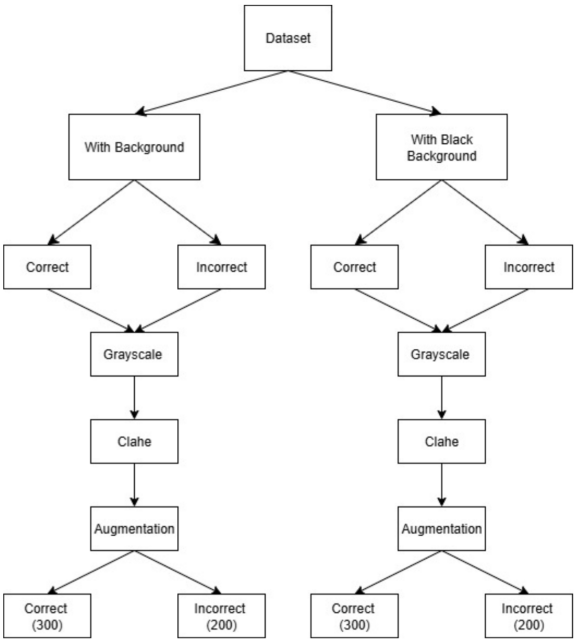


Fig. 2. Data Pre-Process Flowchart

## IV. Dataset Description

The data used for this study is branched into two parts: a correct set and an incorrect set of Bengali character level sign language gestures. Each set contained a total of 36 classes which highlighted hand signs corresponding to Bengali characters. All the images were taken under a consistent background with sufficient lighting so that the focus would be on accurate hand gestures. This was done in order to reduce external factors and to enable the model to only focus on hand signs. The images were taken in RGB format which was crucial in order to distinguish subtle variations in hand shape. The whole dataset maintained a consistent resolution across all the images in order to enable a uniform pre-processing function and feature extraction on the entire dataset. The controlled data setup assured that it contained gestures of uniform visual quality but diverse in sign representations. Each of the characters were then manually mapped to a specific class label in order to produce a detailed 20 comparison of classwise accuracy.

| | | | | | | |
|---|---|---|---|---|---|---|
| 0<br>অ | 1<br>আ | 2<br>ই | 3<br>উ | 4<br>ঊ | 5<br>ঋ | 6<br>ক |
| 7<br>খ | 8<br>গ | 9<br>ঘ | 10<br>চ | 11<br>ছ | 12<br>জ | 13<br>ঝ |
| 14<br>ট | 15<br>ঠ | 16<br>ড | 17<br>ঢ | 18<br>ত | 19<br>থ | 20<br>দ |
| 21<br>ধ | 22<br>প | 23<br>ফ | 24<br>ব | 25<br>ভ | 26<br>ম | 27<br>য |
| 28<br>র | 29<br>ল | 30<br>ন | 31<br>স | 32<br>হ | 33<br>ড় | 34<br>ং |
| 35<br>ঃ | | | | | | |

Fig. 3. Bengali Characters with Class Labels

The majority of the pictures were of the correct classes. The distribution of the correct classes is shown as follows:

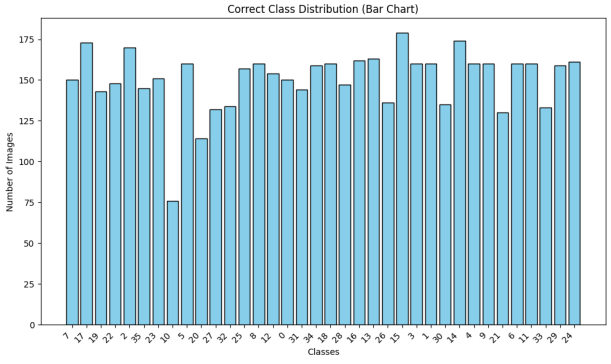


Fig. 4. Correct Class Distribution

The number of images belonging to the incorrect character set was comparatively lower. Variations and intentional inaccuracies were included to improve the model's robustness:

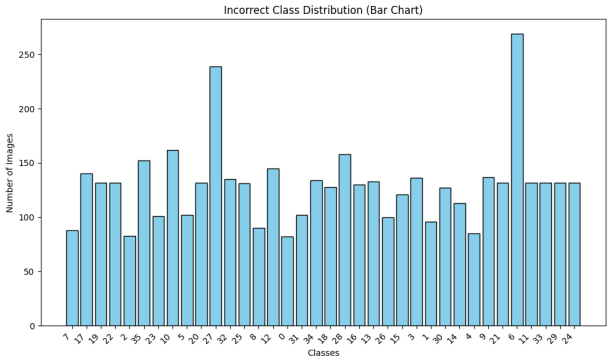


Fig. 5. Incorrect Class Distribution

## V. Data Pre-Processing

The data cleaning process focused on converting the image in such a way that the models could more accurately discriminate and correct the negative images. The images were skimmed through to remove any images that displayed excess pixelations, blurriness, or gestures where certain parts of the hands were out of frame. Additionally, images that had completely overlapped with signs from other classes were removed to maintain the integrity of the dataset. The SHAP library installation confirmed dependencies like scipy, scikit-learn, tqdm, and numba were satisfied.

In order for data to be consistent and have proper quality before model training, a comprehensive pre-processing and transformation pipeline was executed. All the images were resized to 224×224 pixels. This provided consistent input for all models and ensured compatibility with CNN architectures. To reduce any effect of color during model training, the images were converted to grayscale and then restored to a three-channel format. Additionally, Contrast Limited Adaptive Histogram Equalization (CLAHE) was applied to enhance local contrast and fine-grained details. Finally, the images underwent class-wise balancing: correct samples were augmented to 300 images per class, and incorrect samples to 200 images per class.

TABLE I
Pre-Processing Framework

| **Pre-processing Type** | **Description** | **Reasoning** |
|---|---|---|
| Convert to Grayscale | Images converted to grayscale and restored to 3-channel format | Reduces complexity and emphasizes hand shape rather than color. |
| CLAHE Enhancement | Applied CLAHE to improve contrast | Enhances local contrast, hand contour visibility, and gesture edges. |
| Resize | All images resized to 224×224 pixels | Ensures consistent input dimensions. |
| Augmentation | rotation = $\pm$15, width shift = $\pm$0.1, height shift = $\pm$0.1, zoom = $\pm$0.1, horizontal flip | Simulates real-world variations. Dataset balanced to 300 correct and 200 incorrect images per class. |

The dataset was further duplicated into two parts: one with a natural background and one with a uniform black background. The purpose of this was to analyze how background would affect the model's performance. Sample pre-processed images for both backgrounds are shown in Figure

Fig. 6. Image representation before and after pre-processing

## VI. Triplet Network

The triplet-loss framework is central to the learning process. Each training triplet consists of an anchor (incorrect gesture), a positive sample (correct gesture of the same class), and a negative sample (correct gesture from a different class). By

minimizing intra-class distances and maximizing inter-class distances, the model learns a robust embedding space that generalizes to unseen gestures.

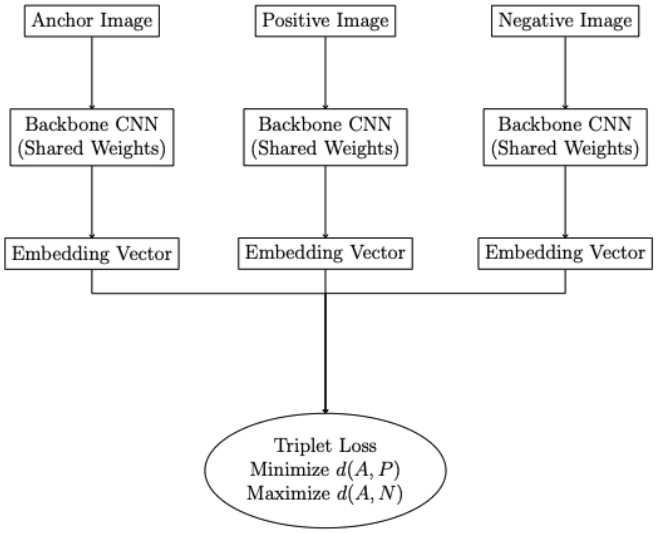


Fig. 7. Triplet Network Architecture: Anchor, Positive, Negative inputs and Embed- ding Output

Then the images are delivered to the CNN model to generate the embeddings. The embeddings are computed using the triplet loss function and depending upon a certain margin the weights of the model are updated. So, the triplet loss is configured in such a way that it tries to minimize the distance between anchor and positive, while also increasing the distance from the negative.

## VII. Embedding Generation

Each CNN backbone was employed as a feature extractor with its classification head removed. The extracted features were projected into a 128-dimensional embedding space via a fully connected dense layer. Batch normalization and dropout were applied to improve generalization, and L2 normalization ensured embeddings had unit length. Training used triplets of anchor, positive, and negative samples with the triplet loss function:

$$L = \max(d(a, p) - d(a, n) + m, 0) \tag{1}$$

where d(a, p) and d(a, n) are squared Euclidean distances between anchor–positive and anchor–negative embeddings, respectively, and m is a fixed margin (0.5). The embedding based approach has various advantages over typical classification. First of all, in normal classification, the model learns strict class boundaries. Meanwhile, for metric learning, it learns a space where it can bring similar gesture embeddings close together and dissimilar ones further apart. Additionally, the embeddings can be used for error correction by comparing the embeddings of an error sign with all the class centroids of the correct class

## VIII. Performance Evaluation

### A. k-NN Accuracy

The K Nearest Neighbors (k-NN) classification was utilized during the mapping of the incorrect test images. A $k = 3$ nearest neighbors with euclidean distance was utilized for all the models throughout all the experiments. For a test sample $i$, let $\hat{y}_i$ denote its predicted class and $y_i$ its true class. The k-NN accuracy is calculated by:

$$\text{Accuracy}_{\text{kNN}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}(\hat{y}_i = y_i) \tag{2}$$

where $N$ is the total number of test samples, and $\mathbf{1}(\cdot)$ is the indicator function, returning 1 if the condition is true and 0 otherwise. This metric directly evaluates how well the learned embeddings preserve class neighborhoods in the feature space.

### B. Centroid Accuracy

Centroid accuracy is another method by which the embeddings were properly mapped to their correct class. The formula for centroid accuracy is as follows:

$$\mathbf{c}_k = \frac{1}{n_k} \sum_{i=1}^{n_k} \mathbf{e}_i \tag{3}$$

where $n_k$ is the number of training samples in class $k$, and $\mathbf{e}_i$ represents the embedding of a sample. The distance from the sample incorrect image to the class centroids were computed to find the overall accuracy. This approach highlights how well embeddings cluster around their true class centers, which is crucial for mapping incorrect signs to their correct forms.

### C. Macro Precision, Recall, and F1-Score

To account for class imbalance and evaluate the model performance across all classes uniformly, macro-averaged precision, recall, and F1-score were computed. For class $k$, the metrics are defined as:

$$\begin{aligned} \text{Precision}_k &= \frac{TP_k}{TP_k + FP_k}, \\ \text{Recall}_k &= \frac{TP_k}{TP_k + FN_k}, \\ F1_k &= \frac{2 \cdot \text{Precision}_k \cdot \text{Recall}_k}{\text{Precision}_k + \text{Recall}_k} \end{aligned} \tag{4}$$

where $TP_k$, $FP_k$, and $FN_k$ represent true positives, false positives, and false negatives for class $k$, respectively. The macro-averaged F1-score is:

$$\text{Macro-F1} = \frac{1}{K} \sum_{k=1}^{K} F1_k \tag{5}$$

These metrics provide a balanced view of performance across all 36 Bengali character classes, preventing dominance by more frequent classes.

### D. Intra-class and Inter-class Distances

The embedding space was further analyzed by computing intra-class and inter-class distances to assess cluster compactness and class separability. For class $k$ with $n_k$ embeddings $\mathbf{e}_i$, the average intra-class distance is:

$$\text{Intra-class distance}_k = \frac{2}{n_k(n_k-1)} \sum_{i<j} \|\mathbf{e}_i - \mathbf{e}_j\|_2 \quad (6)$$

The inter-class distance between class $k$ and class $l$ centroids is:

$$\text{Inter-class distance}_{k,l} = \|\mathbf{c}_k - \mathbf{c}_l\|_2 \quad (7)$$

Small intra-class distances indicate tight clustering of embeddings belonging to the same class, while large inter-class distances ensure clear separation between different classes. Visualization techniques such as t-SNE and UMAP were employed to illustrate these relationships, providing qualitative confirmation of embedding quality.

## IX. Result Analysis

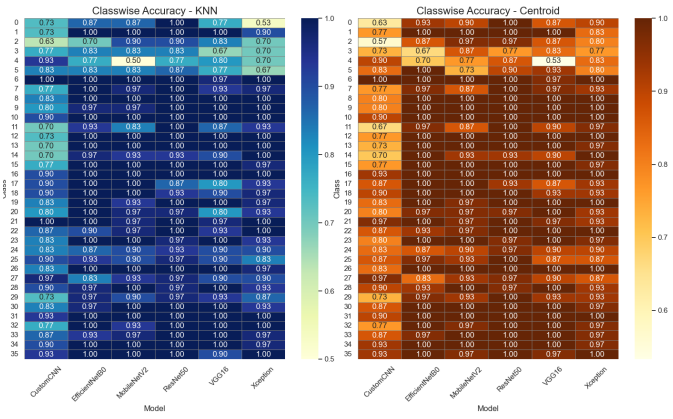


Fig. 8. Class-wise Accuracy of best 6 model

In order to have a better understanding about the best-performing models, the classwise accuracy was investigated using KNN and Centroid based evaluation. The KNN method used the local neighborhood relationships between samples, meanwhile the centroids used global classwise compactness for each.

Some classes that had better performance from both methods included 6,9,10,15,18,21,28,34 and 35. This showcase that features of these classes included unique structure patterns like finger separation and hand orientation which were more effectively captured by the CNN backbones. Thus, as these form tight clusters, it becomes easier for both KNN and Centroid methods to yield the best accuracy among these classes.

On the other hand, certain classes like 0,2,4,11 showed fluctuating accuries especially for the KNN method. These variations in accuracy show how these classes contain similarities between themselves such as similar finger positions and overlapping hand structures.

In general, the centroid based method accuracy still demonstrated better accuracy which highlights the fact that although there may exist local confusions or outliers, the global representation of these clusters in the vector space is still meaningful.

## X. Ablation Study

TABLE II
Hyperparameter Test Configurations for Triplet Network Experiments

| Test | Batch | Epoch | Steps per Epoch | Validation Step | Margin | Embedding Size |
|---|---|---|---|---|---|---|
| 1 | 16 | 100 | 30 | 10 | 0.5 | 128 |
| 2 | 16 | 100 | 100 | 50 | 0.5 | 128 |
| 3 | 16 | 200 | 100 | 50 | 0.5 | 128 |
| 4 | 16 | 150 | 100 | 10 | 0.5 | 128 |
| 5 | 16 | 100 | 100 | 10 | 0.3 | 128 |
| 6 | 16 | 100 | 100 | 10 | 0.7 | 128 |
| 7 | 16 | 100 | 100 | 10 | 0.5 | 256 |

Increasing the number of epochs and steps per epoch consistently improved accuracy for both KNN and centroid-based methods across all backbones. For example, Test 3 achieved the best performance with an accuracy of 97.6% for ResNet50, 96.4% for EfficientB0, 95.8% of MobileNetV2, 95.5% for VGG16, 92.0% for DenseNet121 and 81.5% for custom CNN with normal background based on centroid distancing. Similarly, an accuracy of 95.8% for ResNet50, 94.8% for EfficientB0, 94.0% of MobileNetV2, 93.2% for VGG16, 88.0% for DenseNet121 and 81.3% for custom CNN were achieved with normal background based on KNN=3. Increasing the embedding size from 128 to 256 slightly improved class separability for some models. Shallow models such as CustomCNN and ConvNet showed very limited improvement. Pretrained models such as ResNet50, MobileNetV2, EfficientNetB0, and DenseNet121 consistently outperformed CustomCNN and other shallow models. ResNet50 achieved the highest overall accuracy of 97.6%.

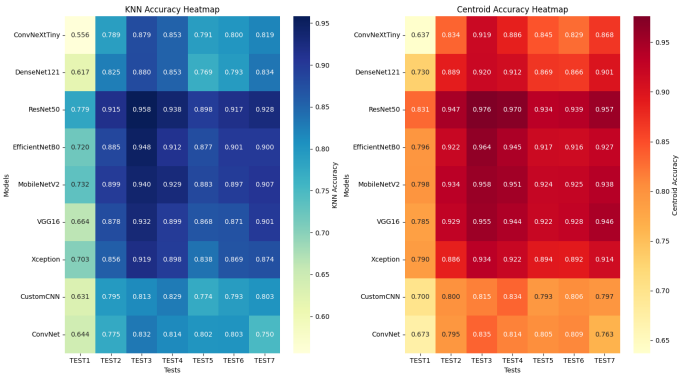


Fig. 9. Class-wise Accuracy of best 6 model

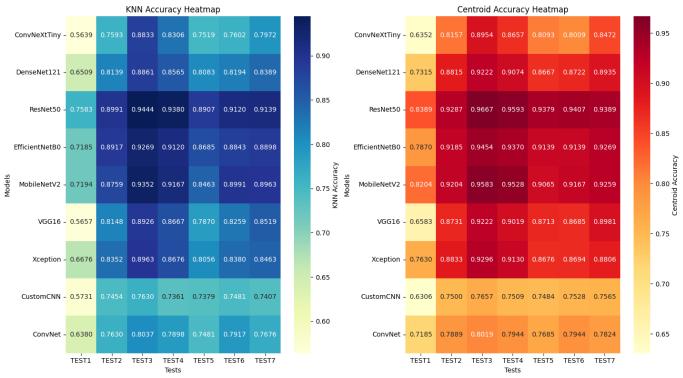


Fig. 10. Class-wise Accuracy of best 6 model

## XI. Conclusion and Future Works

Multiple experiments were conducted across nine deep learning backbones, each yielding promising results. The black background dataset demonstrated high intra-class compactness and clear separability between gesture classes. Increasing the number of epochs to 200 and improving the number of steps per iteration further enhanced embedding quality and classification accuracy. Between the two evaluation strategies, the Centroid-based method proved more stable and effective than kNN, offering better inter-class discrimination and overall accuracy. Among the backbones, ResNet50 consistently achieved over 90% accuracy across experi- ments, demonstrating strong generalization and stable embedding structures. Although other pretrained models such as MobileNetV2 and EfficientNetB0 also performed well, ResNet50 exhibited the most reliable classwise accuracy. In contrast, models like CustomCNN and ConvNet, which were trained from scratch, showed lower performance due to their inability to extract high-quality embeddings. Overall, these findings confirm that the triplet-loss-based framework effectively learns a discriminative embedding space capable of mapping incorrect gestures to their correct representations. The system achieved compact intra-class embeddings and strong inter-class separations, validating the proposed method as a robust approach for gesture correction.

Based on our research, future research can be branched into multiple directions. For example, a larger dataset with greater variations can improve model generalization and capability. Additionally, integrating this framework into real-world applications can help bridge communication gaps between deaf and hearing impaired communities. Media pipe integration can allow more accurate focus on hand gestures leading to better results. Other frameworks such as quadruplet loss and self-supervised learning can improve the model's robustness.

## References


[1] M. Rivera-Acosta, J. M. Ruiz-Varela, S. Ortega-Cisneros, J. Rivera, R. Parra-Michel, and P. Mejia-Alvarez, "Spelling Correction Real-Time American Sign Language Alphabet Translation System Based on YOLO Network and LSTM," *Electronics*, vol. 10, no. 9, pp. 1035, Apr. 2021.

[2] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015, pp. 815–823.

[3] S. Mopidevi, M. V. D. Prasad, and P. V. V. Kishore, "Multiview Meta-Metric Learning for Sign Language Recognition Using Triplet Loss Embeddings," *Pattern Analysis and Applications*, vol. 26, no. 3, pp. 1125–1141, 2023.

[4] Y. Gong, Z. Zhan, Q. Jin, Y. Li, Y. Idelbayev, X. Liu, A. Zharkov, K. Aberman, S. Tulyakov, Y. Wang, and J. Ren, "E2GAN: Efficient Training of Efficient GANs for Image-to-Image Translation," in *Proc. 41st Int. Conf. Mach. Learn. (ICML)*, Honolulu, USA, May 2024.

[5] E. Lin, "Comparative Analysis of Pix2Pix and CycleGAN for Image-to-Image Translation," *Highlights in Science, Engineering and Technology*, vol. 39, pp. 915–925, 2023.

[6] A. Haque, R. A. Pulok, M. M. Rahman, S. Akter, N. Khan, and S. Haque, "Recognition of Bangladeshi Sign Language (BdSL) Words Using Deep Convolutional Neural Networks (DCNNs)," *Emerging Science Journal*, vol. 7, no. 6, pp. 2183–2201, Dec. 2023.

[7] M. J. Raihan, M. I. Labib, A. A. J. Jim, J. J. Tiang, U. Biswas, and A.-A. Nahid, "Bengali-Sign: A Machine Learning-Based Bengali Sign Language Interpretation for Deaf and Non-Verbal People," *Sensors*, vol. 24, no. 16, pp. 5351, Aug. 2024.

[8] B. Natarajan et al., "Development of an End-to-End Deep Learning Framework for Sign Language Recognition, Translation, and Video Generation," *IEEE Access*, vol. 10, pp. 104358–104371, 2022.