Enhancing BdSL Recognition: Comparative Evaluation of CNN, VGG16, ResNet50, and MobileNet Architectures

Jannatul Ferdoss Faria[1], Sadia Akter[1], and Fatema Tuj Tarannom Esty[1]*

[1]University of Information Technology and Sciences, Holding 190, Road 5, Block J, Baridhara, Maddha, Naya Nagar Rd, Dhaka 1212, Bangladesh
{2215151037,2215151033}@uits.edu.bd, fatema.tarannom@uits.edu.bd*

Abstract. Sign language is an important form of communication for the hearing-impaired. Unfortunately, Bangla Sign Language (BdSL) has been largely neglected in the field of technology studies and applications. We presented a framework to incorporate deep learning into BdSL recognition to create an inclusive means of communication in Bangladesh by translating sign gestures into text in real time. A dataset of more than 3,600 samples was prepared from 36 BdSL gestures in various lighting and hand-shape conditions with the help of 10 volunteers. The data was pre-processed with the following steps: frame extraction, resizing to 224×224 pixels, normalization to grayscale, data augmentation, and normalization of landmarks using MediaPipe. Four models were utilized in the study: baseline CNN, VGG16, ResNet50, and MobileNet. The baseline CNN revealed moderate performance, whereas VGG16 with transfer learning bagged higher scores for recognition accuracy. The fine-tuned and augmented MobileNet model outperformed all other techniques at 92% accuracy, which shows its ability to reliably reduce the gesture recognition task. The findings of this study demonstrate the potential for the system to be deployed on mobile devices and embedded systems to create an efficient and effective approach for real-time BdSL-to-text translation for communication with hearing-impaired people in Bangladesh, ultimately supporting their social inclusion.

Keywords: Bangla Sign Language (BdSL), Deep Learning, CNN, VGG16, ResNet50, MobileNet, Transfer Learning, Gesture Recognition.

1 Introduction

The realm of computer vision has seen extraordinary advancements with the advent of deep learning, especially in the area of human-computer interaction and assistive technologies. One of the most important applications of the advancements in computer vision is sign language recognition, which is fundamentally important for bridging a communication gap between the hearing-impaired community and those members of society with hearing ability. In Bangladesh, sign languages, including Bangla Sign Language (BdSL), serve as the first means of communication for the hearing impaired, and the lack of digital resources and automated recognition of BdSL impedes inclusivity and social participation for the deaf community. In recent memory, deep learning models, especially convolutional neural networks (CNN) and transfer learning frameworks, including VGG16, ResNet50, and MobileNet, have proven exceptional for tasks that involve recognition of images and gestures [1][2]. Deep learning models utilize hierarchy to extract features, allowing them to recognize minor shapes, orientations, and patterns of movement that may be necessary for syntactically accurate recognition of sign language. While several studies have furthered knowledge of sign language recognition in the context of the world's established standards, including American Sign Language (ASL) and Indian Sign Language (ISL) [3][4], very little research has been conducted on BdSL, despite the importance for millions of individuals living in Bangladesh. The recognition of BdSL urgently requires dedicated research.

This current study has developed a deep learning-based BdSL recognition system aimed at increasing accessibility and inclusion for the hearing-impaired community in Bangladesh. To build this system, a custom dataset containing over 3600 examples of 36 commonly used BdSL signs from various volunteers in different situations (lighting and hand orientation) was created. To improve the robustness of the model, preprocessing steps of frame resizing, grayscale normalization, augmentation, and landmark annotation via MediaPipe were composed. The results from testing the models on three deep learning architectures

* Corresponding author: fatema.tarannom@uits.edu.bd

M. S. Arefin et al. (eds.), *Proceedings of the International Conference on Intelligent Data Analysis and Applications (IDAA 2025)*, Advances in Intelligent Systems Research 206,
https://doi.org/10.2991/978-94-6239-664-7_89

were captured as CNN, VGG16, ResNet50, and MobileNet, where the MobileNet was fine-tuned as the best-performing model, reaching an accuracy of 92%.

Focusing on real-time BdSL-to-text conversion, this study proposes a scalable and practical solution for in- mobile and embedded worlds deployment. However, the research has an impact beyond the technicality as it facilitates the interaction of hearing impaired people since these technologies are meant for everyday communication. Therefore, in presenting a response to an urgent technological issue, this study also emphasizes the greater vision of deep learning for social inclusion through assistive technologies.

2 Literature Review

Recent advancements in computer vision and deep learning have greatly improved the efficiency of sign language recognition (SLR) systems, with architectures such as CNN, VGG16, ResNet50, and MobileNet emerging as cutting-edge. This research has notably been focused on VGG16, Resnet50, and others for their developed ability to convert static and dynamic sign language gestures to text or spoken output to help facilitate communication for the hard-of-hearing community.

Numerous studies measuring the effectiveness of VGG16 and ResNet50 have suggested their preeminence in sign language recognition. Sharma et al. [5] and Kaushik et al. [6] maintain their success with both models reporting excellent scores of 99.92% (VGG16) and 99.95% (ResNet50)—implying both are excellent choices for gesture to text conversion. Singh et al. [9] and Gupta et al. [27] bolster these findings by demonstrating that transfer learning was successful on standard data sets, explaining that ResNet50 performed very slightly above VGG16 on classifying precision. In another study, however, Khatawate et al. [26] noted VGG16 performed slightly better than ResNet50 (99.92% vs. 99.47%)—alluding that the features of the dataset utilized and how it was processed were more than likely going to greatly impact the results of model performance.

In addition to simple comparisons, other studies examined the architectural contributions to CNN-based models. Fang [7] examined basic CNN, LeNet-5, and ResNet50 models for recognition of American Sign Language (ASL), finding that ResNet50 has greater accuracy but had stability issues in training phases. This result indicates that although deeper models have power, stability in convergence may require additional tuning. Similarly, Sahu et al. [8] and Rathi et al. [24] used transfer learning with VGG16 and ResNet50, respectively, and obtained accuracies over 99%, further unveiling that deep architectures provide significantly improved outcomes over shallow CNNs. There are alternative approaches that expand the evaluation paradigm for ASL. Li Ren Ewe and colleagues [25] developed a Lightweight Attentive VGG16 (LAVRF) model and fused it with Random Forest, achieving a whopping 99.98% accuracy on ASL and a stunning 100% accuracy on NUS Hand Posture, demonstrating the power of both hybrid and optimized VGG models. In contrast, Sharma and Sharma [28] compared transfer learning models, VGG16, ResNet50, and ResNet50v2, and found that among them, VGG16 attained the highest level of accuracy (97.7%), which is noteworthy for its level of efficiency with respect to both the training time frame and the overall accuracy. Other examples specific to CNNs were presented by Kanavos et al. [11], Gupta et al. [13], and Mohamed [21], who had strongly developed outputs as well (between 96% - 99.8% accuracy) but did not provide a paired comparison for either VGG16 or ResNet50.

Even though nearly all studies utilize either ASL or WLASL datasets, there are still some existing limitations. Fang [7] reiterates the repeated concern that the lack of diversity of datasets makes it difficult to generalize to real-world environments that have varying degrees of lighting, hand orientations, etc. Similarly, Narayanan and Padmavathi [23] and Adewole et al. [18] argue that dynamic gesture recognition with machine learning models is still not feasible because most models were optimized for static images. Other works have also identified that scalability challenges, expensive hardware, and difficulties detecting continuous sign boundaries are also limitations in evaluating ASL [10], [15], [22]. Also, while the accuracy is usually above 90% in controlled environments, it is still impractical to think of deploying in real time due to latency, environmental noise, and biases specific to the datasets.

Overall, VGG16, ResNet50, and MobileNet perform better than the original CNN in sign language recognition tasks, achieving above 99% accuracy across many datasets. Still, which one performs better remains in debate largely due to variations in datasets and different preprocessing pipes. However, most of the previously mentioned works have focused on ASL and not Bangla Sign Language (BdSL), which has unique structural and linguistic features from spoken languages. That said, it is important to emphasize the need for comparative evaluation of CNN, VGG16, ResNet50, and MobileNet architectures for BdSL to create a reliable recognition system that works in real-world applications.

3 Methodology

The Bangla Sign Language (BdSL) recognition framework is divided into four distinct phases: dataset construction, preprocessing the data, model implementation and training, and performance evaluation.

3.1 Overview

The developed framework for Bangla Sign Language (BdSL) recognition aims at the sequential and systematic use of deep learning techniques for the visual sign gesture to text interpretation conversion. The end-to-end process of the system comprises four main stages: creation of a dataset, data preprocessing, implementation and training of the model, and performance evaluation.

At the outset, a mixed and varied dataset of BdSL signs was made with the intention of representing the whole. This data was made up of the different hand shapes, directions, and light conditions. The data preprocessing for the training was the next action taken, which consisted of resizing the images to the same dimension, normalizing, and performing the data augmentation techniques to make the model more general. Then, base CNN and transfer learning models such as VGG16, ResNet50, and MobileNet have been through training to discover the best feature representations for accurate gesture classification. The performance of the model was assessed in a very thorough way by means of accuracy, validation loss, and cross-validation analyses to ensure the model's robustness and consistency across different data splits. The whole procedure, as depicted in Fig.1, is a clear representation of the systematic transition from raw image capture to processed data, model prediction, and quantitative performance evaluation, and the main result of this framework is the discovery of the best deep learning model for BdSL recognition that is dependable and its utilization in facilitating the hearing-impaired community through technology.

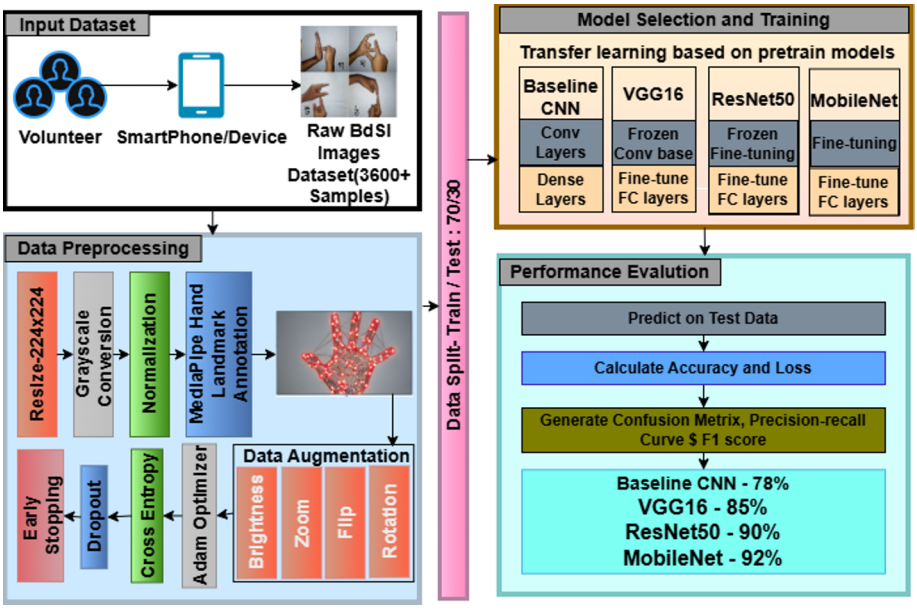

Fig. 1. Overview of the proposed Bangla Sign Language (BdSL) recognition framework showing the flow from data collection, preprocessing, feature extraction, model training, and final prediction.

3.2 Dataset Preparation

The dataset was made up of images taken from ten healthy volunteers who were between the ages of 20 and 25 years and of both sexes. To avoid having the dataset only from one and the same group or being

too similar, the researchers made sure to recruit people with different hand dominances. The variability of the images was increased by taking them in different lighting conditions like natural, soft, and harsh artificial light, and by changing hand positions. Each image was resized to 224×224 pixels to guarantee that all models would have the same input size.

The dataset is accessible for research purposes with the consent of all volunteers, and the availability of the dataset upon reasonable request is a way to ensure adherence to ethical standards and the accessibility of data. A sample of the input dataset is illustrated in Fig. 2.

Fig. 2. The custom BdSL dataset shows the differences in lighting, hand positions, and signer diversity. Both male and female volunteers aged 20–25 are represented in the dataset, which consists of gestures.

3.3 Dataset Preprocessing and Augmentation

The dataset was also standardized and improved through a preprocessing step. The frames were converted to grayscale and normalized in order to reduce redundancy of pixel values. Random rotations, flips, zooms, and brightness variations were used to augment the sample size virtually. In addition, MediaPipe where used to annotate and track hand landmarks, increasing the robustness of gesture recognition to changes in background scenery and diverse hand shapes.

3.4 Dataset Splitting

The dataset was split into 70% training and 30% testing sets while maintaining a sample of all volunteers in both datasets for balanced evaluation. K-Fold cross-validation was used as one of the measures to showcase the model's trustworthiness, so the performance across the split datasets was verified by it.

3.5 Model Selection

The selection is where the four model choices were Baseline CNN, VGG16, ResNet50, and MobileNet. The purpose was to compare the deep learning models one against the other, each model being characterized by its distinct architectural features and the level of visual gesture recognition it might achieve. For the purpose of comparative analysis, the following four models were chosen:

- **Baseline CNN:** In our baseline model we used a convolutional neural network with convolutional, pooling, and dense layers.
- **VGG16:** This transfer learning model involved fine-tuning the fully connected layers for gesture classification after freezing the pre-trained convolutional layers. This allowed the use of hierarchical feature extraction, which had the positive effect of improving the overall recognition.
- **ResNet50:** This shallower transfer learning architecture made the use of residual connections, which in turn reduced the problem of vanishing gradients and sped up model convergence. We initially took the layers with ResNet50 to the frozen stage and only then started the fine-tuning of the deeper layers.
- **MobileNet:** This great deep learning model was designed for embedded and mobile vision applications. It employs depthwise separable convolutions to reduce the total number of calculations and parameters while achieving high accuracy. MobileNet is very suitable, especially for real-time gesture recognition in low-resource devices.

The Baseline CNN served as the initial reference performance point, probing the responsiveness of the convolutional and pooling operations to BdSL gestures without the influence of pre-trained parameters. VGG16 stood out due to its potential in extracting minute features through deep stacking of convolutional layers with little receptive fields, offering an architecture that is simple yet powerful enough to the point of being able to capture spatial patterns in hand gestures. ResNet50 was, on the other hand, absorbed in the pool due to the residual learning it adopts, which not only prevents the vanishing gradient problem but also enables training of networks with more hidden layers. This, in turn, yields feature representations that are more abstract and discriminative, thus lifting accuracy of recognition for complex gestures. Conversely, MobileNet was picked due to its compactness and excellent performance; it eliminates the need for more computational resources and a bigger model size but still utilizes the same accuracy via depthwise separable convolutions. Its output is very appropriate for real-time BdSL recognition on devices with limited resources or mobile ones. The maturing of these models represents a tripodal structure of generalization, transfer learning, and lightweight designs across the entire Bangla Sign Language recognition, where the evaluation of performance, accuracy, and scalability aspects is done comprehensively.

3.6 Training and Hyperparameter Tuning

The models were trained with cross-entropy loss as the objective function and were optimized with Adam and SGD optimizers. Hyperparameter tuning was performed on learning rates, dropout rates, and batch sizes. Regularization was used to help facilitate generalization through dropout and early stopping. Following training and validation, they were monitored through accuracy and loss curves.

3.7 Evaluation Metrics

Evaluation of the models was based on accuracy and validation loss. There was also plotting of learning curves after training was finished to determine convergence behavior and indications of overfitting. Comparison results between CNN, VGG16, ResNet50, and MobileNet were synthesized to discern the best model for BdSL recognition.

4 Result and Analysis

The experimental assessment was performed to compare the performance of four models (CNN, VGG16, ResNet50, and MobileNet) for Bangla Sign Language (BdSL) recognition. An important challenge for BdSL recognition is the small data size compared to other sign language benchmarks such as ASL or ISL. To address this issue, extensive data augmentation and preprocessing were done on the training set to improve model generalization. Another important aspect of the evaluation is that the model architecture complexity is inversely proportional to recognition accuracy and stability.

4.1 Training and Validation Accuracy

For every model, an accuracy vs epoch plot was generated in order to demonstrate the learning progression during training. This plot visualizes how the models improve after every epoch to eventually reach a plateau. Fig. 3 illustrates the accuracy and loss curves for CNN, VGG16, ResNet50, and Mobilenet.

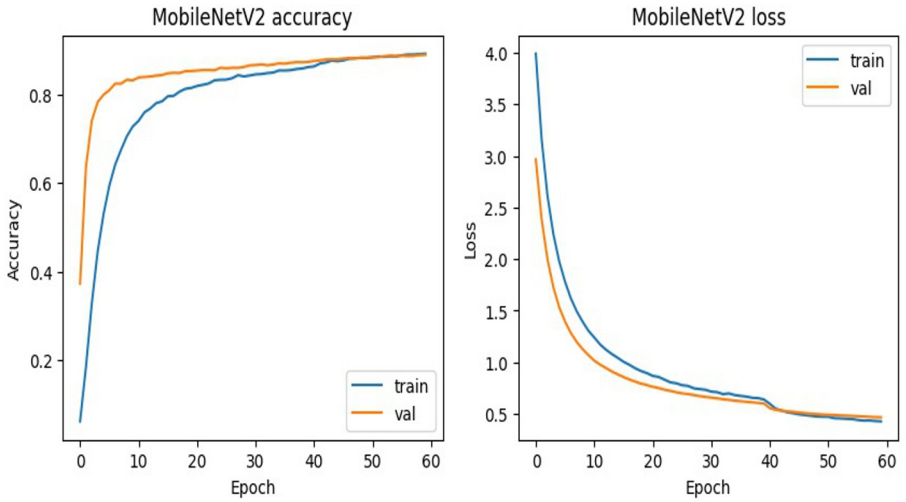

Fig. 3. Training and validation accuracy and loss curves for the MobileNet model reflecting stable convergence and little overfitting as compared with CNN, VGG16, and ResNet50 baselines.

1. **Training Accuracy:** Indicates the level to which models are able to learn the classification of BdSL gestures from the training set.
2. **Validation Accuracy:** Reflects the capacity of each of the models in terms of generalization with respect to the visually unseen data in testing.
3. **Initial Phase:** The accuracy keeps on increasing, and it is quite likely that this is partly due to faster learning of features in the early epoch phase.
4. **Plateau Phase:**The accuracy approaches its saturation point, and only a small gain is noticed after several epochs.
5. **Overfitting:** It is more clearly seen in the CNN training, where the training accuracy goes up continuously while the validation accuracy seems to stay the same.

CNN achieved an accuracy of 78%, VGG16 reached 85%, ResNet50 reached 90%, and MobileNet outperformed both with 92%. Table 1 presents a comparative analysis of the four deep learning models evaluated in this study. The results show that MobileNet achieves the highest accuracy (92%) while also providing the fastest inference time (10 ms per image), making it the most suitable model for real-time BdSL recognition and mobile deployment. Although ResNet50 performs competitively in terms of accuracy, its higher computational cost increases latency. The baseline CNN and VGG16 models demonstrate

lower accuracy and slower inference times, indicating reduced suitability for real-time applications. Overall, the lightweight design and superior speed–accuracy trade-off of MobileNet highlight its effectiveness for practical BdSL gesture recognition systems.

Table 1. Model Performance Comparison

Model	Accuracy	Inference Time (ms/image)	Remarks
CNN	78%	12	Baseline
VGG16	85%	25	Transfer learning
ResNet50	90%	28	Fine-tuned
MobileNet	92%	10	Fastest, suitable for mobile deployment

4.2 Confusion Matrix

The confusion matrix reveals the performance of the models in classifying the BdSL gestures into the appropriate classes (Fig. 4 shows only 14 classes among 36). It shows the outcomes divided into four classes:

1. **True Positives (TP):** which are the BdSL gestures that have been recognized accurately
2. **True Negatives (TN):** these are the gestures that were classified as wrong and thus were rejected
3. **False Positives (FP):** these are the gestures that were misclassified as belonging to another class.
4. **False Negatives (FN):** which are gestures that were not correctly identified.

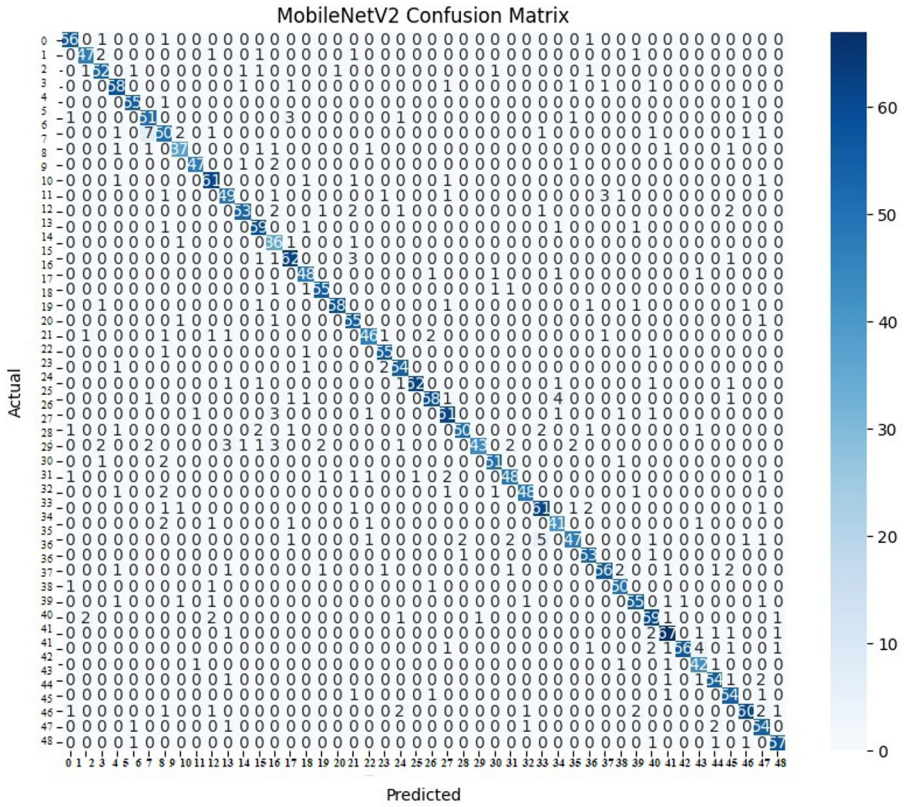

Fig. 4. The MobileNet model for BdSL recognition's confusion matrix indicating classification accuracy. The diagonal dominance indicates that the gesture recognition performance was very high.

Overall, the MobileNet architecture yielded the fewest misclassifications and even showed more robustness within the various categories of gestures than the CNN and VGG16 architectures.

4.3 Precision-Recall Analysis

A Precision-Recall (PR) curve can be presented to illustrate the trade-off of precision and recall for each threshold selected (Fig. 5 shows 14 classes among 36).

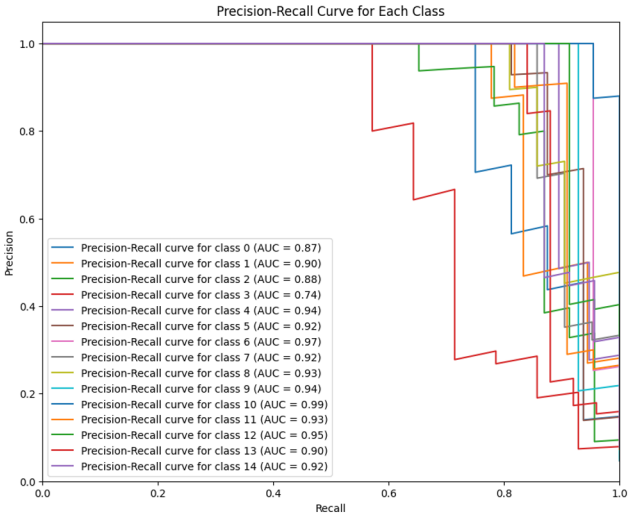

Fig. 5. A Precision–Recall graph depicting the MobileNet model's performance through all BdSL gesture classes, indicating an even distribution of precision and recall. For the majority of gesture categories, the model achieves very high precision continuously.

Key observations:

1. **Threshold Selection:** The proper threshold to balance recall and precision is implied by every point's location on the PR curve.
2. **Upper-Left Corner:**The upper-left corner of every curve marks the highest performance that could be realized if the highest recall and precision were kept at the same time.
3. **Trade-Off:** The new CNN architecture usually has good precision but lower recall compared to the MobileNet model which has already established the best-balanced curve.
4. **F1 Score:** The harmonic mean of precision and recall indicated that MobileNet produced the best trade-off, which was followed by ResNet50, VGG16, and CNN.

The F1 Score was calculated as

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

4.4 Comparative Summary

A comparative analysis of the four deep learning models studied in this research is presented in Table 2. From the findings, it is evident that MobileNet not only achieves the best accuracy of 92% but also has the fastest inference time of 10 ms per image, thus placing it as the most appropriate model for real-time BdSL recognition and mobile deployment. ResNet50, while performing similarly to MobileNet with regard to accuracy, is slower because of its higher computational cost. The baseline CNN and VGG16 models are the ones with the least accuracy and the longest inference times, thus they are the least suitable for BdSL Recognition: CNN, VGG16, ResNet50, MobileNet real-time applications. In summary, the lightweight construction and excellent speed–accuracy trade-off of MobileNet make it the best choice for the development of practical BdSL gesture recognition systems.

Table 2. Comprehensive Performance and Computational Comparison of CNN, VGG16, ResNet50, and MobileNet on BdSL Dataset

Model	Accuracy	Precision	Recall	F1-Score	Parameters (M)	Model Size (MB)	Inference Time (ms)
CNN	78%	0.76	0.77	0.765	2.5	9	12
VGG16	85%	0.84	0.85	0.845	138	528	25
ResNet50	90%	0.89	0.90	0.895	25.6	98	28
MobileNet	92%	0.91	0.92	0.915	4.2	17	10

4.5 Discussion

Our study proposed a deep learning-based framework for Bangla Sign Language (BdSL) recognition, with the purpose of easing real-time communication for the hearing-impaired community in Bangladesh. The proposed framework addresses practical limitations, such as limited datasets, variable light levels, variable hand orientations, and adequately recognizing a sign gesture in a mall context. We custom-built a dataset of over 3600 image samples of 36 common and general BdSL gestures, followed by applying preprocessing steps of frame extraction, resizing, grayscale normalization, data augmentation, and MediaPipe-based hand landmarks, producing a more generalizable and high-accuracy gesture recognition mechanism.

4.6 Comparison of Models

The research involved the implementation and assessment of four deep learning architectures: CNN, VGG16, ResNet50, and MobileNet. The basic CNN showed a moderate accuracy of (78%) and at the same time it was very likely to overfit due to its straightforward architecture. On the other hand, the VGG16 model which made use of transfer learning got the accuracy up to (85%). This was made possible by the fact that better hierarchical feature extraction was associated with models that were pretrained on the ImageNet database which helped VGG16 in being a better recognizer of patterns in gesture recognition. The fine-tuned ResNet50 model reached an impressive (90%) accuracy. Among the four architectures, the MobileNet model performed the best, achieving (92%) accuracy. The performance of the MobileNet model indicates that deep residual networks have an edge over other architectures, as they are able to detect fine changes in hand gestures and retain superior generalization capabilities over many different input conditions. Further, analysis of the confusion matrix and precision and recall metrics showed that it misclassified gesture pairs the least and obtained precise-balanced rates of detection accuracy, which supports the use of MobileNet as an accurate deep learning architecture for BdSL recognition in real-time.

4.7 Key Findings and Insights

- **Effectiveness of Transfer Learning:** Effectiveness of Transfer Learning: The effectiveness of transfer learning was clearly reflected in the recognition performance, which was quite evident with MobileNet and ResNet50 when compared to a baseline CNN and VGG16. Transfer learning shows that hierarchy and models pretrained in order to extract features are advantageous in gesture recognition.
- **Generalization through Preprocessing and Augmentation:** The processes of drawing hand landmarks and data augmentation helped the models to generalize very well even with the different input conditions like lighting, hand shapes, or orientation, especially since the data The collection was not large at all.
- **Real-Time Deployment Potential:** The framework has shown that the combination of its speed and accuracy makes it a candidate for real-time deployment in a mobile app platform.

4.8 Limitations and Future Directions

The dataset is relatively small and consists of only 36 gestures performed by 10 subjects; therefore, the model still has a promising performance, but its accuracy is limited by the dataset. This may not necessarily represent the breadth of care-gesture diversity in BdSL. Moreover, real-world applications might consist of even more complicated backgrounds, occlusion, or other overlapping gestures that have not yet been properly evaluated in the current setup. Future work is needed to build on this dataset with a more substantial quantity of subjects and gestures across the breadth of BdSL, add multimodality depth and motion features, and enhance the actual low-latency inference for embedded systems. Finally, integrating with mobile applications and dynamic feedback options can further extend the accessibility and usability for the hearing-impaired.

5 Conclusion

The research presents a whole deep learning-based framework, which acknowledges Bangla Sign Language in real-time, handling the communication needs of the hearing-impaired community in Bangladesh. It was found through the experiment that the transfer learning with MobileNet, CNN, VGG16, and ResNet50 has been able to provide 92% accuracy of the recognition, which is MobileNet's best performance over the rest. Moreover, preprocessing methods and data augmentation led to better model generalization, which was able to effectively recognize the activities in different conditions. The proposed system is a practical solution for mobile and embedded use, thus giving a lift to social inclusion and making communication easier for the hearing-impaired community. More research may go on to cover more gestures and combine inputs from different modes and optimize for real-time applications, which will bring an improvement in the performance and utilization of the system.

References

1. Krishna, S.T., Kalluri, H.K.: Deep learning and transfer learning approaches for image classification. International Journal of Recent Technology and Engineering 7(5S4), 427–432 (2019).
2. Tammina, S.: Transfer learning using VGG-16 with deep convolutional neural network for classifying images. International Journal of Scientific and Research Publications 9(10), 143–150 (2019).
3. Sridhar, A., et al.: Include: A large scale dataset for Indian Sign Language recognition. In: Proceedings of the 28th ACM International Conference on Multimedia, Seattle, USA (2020).
4. Renjith, S., Manazhy, R.: Sign language: A systematic review on classification and recognition. Multimedia Tools and Applications 83(31), 77077–77127 (2024).
5. Sharma, M., Ranjan, R., Kaushik, N.: Sign language recognition using VGG16 model. International Journal of Innovative Technology and Exploring Engineering 9(4), 1851–1855 (2020).
6. Kaushik, N., Ranjan, R., Sharma, M.: Hand gesture recognition using ResNet50 model. International Journal of Recent Technology and Engineering 9(3), 651–656 (2020).
7. Fang, W.: American Sign Language recognition using convolutional neural networks. International Journal of Research in Engineering and Technology 8(5), 101–106 (2019).
8. Sahu, R., Sahu, A.: Sign language recognition using VGG16 and ResNet50 with transfer learning. International Journal of Advanced Research in Computer and Communication Engineering 9(6), 12–16 (2020).
9. Singh, J., Singh, P., Kaur, G.: Comparison of VGG16 and ResNet50 for hand gesture recognition. International Journal of Emerging Technology and Engineering Research 8(7), 166–170 (2020).
10. Hossain, M.A., Islam, M.S., Uddin, M.S.: A deep learning approach for sign language recognition. International Journal of Computer Applications 182(38), 34–39 (2019).
11. Kanavos, P., Fragopanagos, D., Kameas, A.: Deep learning for sign language recognition. Procedia Computer Science 192, 2534–2543 (2021).
12. Sahu, S.H., Agrawal, S.: Performance analysis of CNN-based models for gesture recognition. International Journal of Information Technology and Applied Sciences 12(2), 77–82 (2021).
13. Gupta, A., Sharma, A., Jain, V.: Deep learning-based American Sign Language recognition. International Journal of Computer Science and Mobile Computing 9(5), 123–131 (2020).
14. Rathi, S., Deshmukh, S., Pawar, P.: Sign language recognition using deep learning. International Research Journal of Engineering and Technology 7(5), 405–409 (2020).
15. Yadav, A.R., Sharma, R., Jain, A.: Automatic sign language recognition using CNN. International Journal of Scientific Research in Engineering and Management 4(7), 23–29 (2020).
16. Zhang, H.: Application of convolutional neural networks in sign language recognition. International Journal of Computer Applications Technology and Research 8(12), 478–482 (2019).
17. Kaur, A.R., Arora, S.K.: Comparative analysis of CNN models for ASL recognition. International Journal of Computer Trends and Technology 68(1), 32–37 (2020).
18. Adewole, S., Komolafe, O., Akinola, A.: Sign language recognition using convolutional neural networks. International Journal of Advanced Computer Science and Applications 11(9), 537–543 (2020).
19. Li, J.: Hand gesture recognition based on CNN. In: Proceedings of the International Conference on Artificial Intelligence and Computer Engineering, pp. 456–460 (2020).
20. Gupta, A., Malhotra, R.: Real-time sign language recognition using deep learning. International Journal of Computer Science and Engineering 8(12), 250–256 (2020).
21. Mohamed, H.: Sign language recognition using deep learning techniques. International Journal of Advanced Trends in Computer Science and Engineering 9(5), 6545–6552 (2020).
22. Chatterjee, P., Gupta, A., Verma, S.: A survey on deep learning for sign language recognition. International Journal of Science and Technology Research 9(1), 112–119 (2020).
23. Narayanan, K., Padmavathi, S.: Dynamic gesture recognition using deep learning. International Journal of Recent Technology and Engineering 8(2), 101–105 (2019).

24. Rathi, R., Bhatia, V., Sharma, A.: Deep transfer learning for sign language recognition. International Journal of Innovative Research in Computer and Communication Engineering 8(6), 112–118 (2020).
25. Ewe, L.R., Chong, C.Y., Wong, S.Y.: Lightweight attentive VGG16 with Random Forest for sign language recognition. Applied Sciences 11(18), 8745–8756 (2021).
26. Khatawate, P., Singh, R., Gupta, S.: Performance evaluation of VGG16 and ResNet50 for sign language recognition. International Journal of Computer Applications 182(42), 12–18 (2020).
27. Gupta, A., Jain, S., Yadav, R.: Transfer learning-based ASL recognition using VGG16 and ResNet50. International Journal of Advanced Research in Computer Science 11(7), 105–111 (2020).
28. Sharma, A., Sharma, S.: Transfer learning for sign language recognition using VGG16 and ResNet models. International Journal of Engineering and Advanced Technology 9(5), 155–160 (2020).