

Real-Time Sign Language Recognition and Speech Conversion Using Mobilenetv2 Transfer Learning

Rakshitha N¹, Mrs. Vinodhini²

¹Department of Computer Applications [PG] Jyoti Nivas College Autonomous

²Assistant Professor, Department of computer Applications [PG] Jyoti Nivas College Autonomous

Abstract—Communication between hearing-impaired individuals and the general public is often challenging due to limited understanding of sign language. This paper presents a real-time Sign Language Recognition (SLR) system that recognizes American Sign Language (ASL) hand gestures using deep learning and converts them into text and speech. The proposed system employs MobileNetV2 transfer learning for gesture classification and OpenCV for real-time image acquisition and preprocessing. Live video frames captured through a webcam are processed by extracting the Region of Interest (ROI), resizing the image to 224×224 pixels, and feeding it into the trained deep learning model. The recognized gesture is displayed as text on a Flask-based web interface and can be converted into speech using a text-to-speech module. To improve prediction stability, a confidence threshold and stable-frame validation mechanism are incorporated. Experimental evaluation on the ASL dataset achieved an overall classification accuracy of 95%, demonstrating that the proposed approach provides reliable and efficient real-time recognition. The developed system offers an affordable communication aid for hearing-impaired individuals and can be extended for continuous sentence recognition and multilingual translation.

Index Terms—Sign Language Recognition, MobileNetV2, Deep Learning, Transfer Learning, OpenCV, Flask, Computer Vision, Speech Synthesis.

I. INTRODUCTION

Sign language is widely used by people with hearing or speech impairments, but communication can become difficult when the other person has no knowledge of signing. This challenge can occur in educational institutions, hospitals, workplaces, and public service settings where an interpreter may not always be available. Recent developments in artificial intelligence and computer vision have made

automated hand-gesture interpretation increasingly practical. Such systems can identify visual patterns and transform recognized gestures into forms that are easier for non-signers to understand.

Deep-learning approaches have transformed image classification by learning useful visual representations directly from training data. For real-time applications, MobileNetV2 offers a practical trade-off between computational demand and recognition capability. Its compact design also makes it suitable for deployment on devices with restricted processing resources while retaining effective prediction performance.

This study develops a real-time sign language recognition application based on MobileNetV2 transfer learning. A webcam supplies hand-gesture frames, OpenCV prepares the input, and the trained classifier identifies the corresponding ASL character. The predicted character is presented through a Flask web interface and can be converted into speech with a text-to-speech component. Confidence filtering and consecutive-frame verification are used to limit unstable outputs during continuous recognition.

Testing on the ASL dataset resulted in 95% classification accuracy, showing that the approach can support practical gesture-based communication. The implementation is lightweight, provides a simple user interface, and operates in real time, making it applicable to educational and accessibility-oriented scenarios.

II. LITERATURE REVIEW

Research on sign language recognition has progressed across both computer vision and artificial intelligence. Initial systems often used instrumented gloves containing flex sensors or accelerometers to capture hand movement. While such hardware-based

approaches could provide accurate measurements, they introduced additional equipment, cost, and implementation complexity.

As camera-based computer vision matured, recognition systems increasingly moved away from wearable sensing devices. OpenCV provided a practical means of acquiring and preparing visual data, while Convolutional Neural Networks (CNNs) enabled automatic feature learning rather than dependence on manually designed image descriptors. Transfer learning is widely used for gesture classification because pretrained visual models can be adapted when task-specific training data are limited. Architectures including VGG16, ResNet50, EfficientNet, and MobileNetV2 have been explored for this purpose. MobileNetV2 is especially appropriate for real-time use because its compact architecture requires relatively modest computational resources.

Deep-learning recognition systems have also been combined with web frameworks such as Flask to provide interactive interfaces that display predicted gestures and generate speech. However, recognition quality can still vary with illumination, background conditions, and frame-to-frame prediction instability. The present work addresses these issues by pairing MobileNetV2 transfer learning with confidence filtering and consecutive-frame validation.

III. PROPOSED METHODOLOGY

The proposed application converts American Sign Language (ASL) gestures into text and speech through a sequence of image capture, preprocessing, classification, text formation, and speech synthesis stages. Figure 1 presents the overall processing architecture.

Initially, live video is captured using a webcam connected to the system. OpenCV continuously reads the video frames and extracts a predefined Region of Interest (ROI) where the user places the hand gesture. Restricting recognition to the ROI minimizes background interference and improves computational efficiency.

Before inference, the captured hand region is scaled to 224×224 pixels, normalized, and arranged in the input representation expected by MobileNetV2. The resulting image is then supplied to the trained network for gesture classification.

The prediction with the highest probability is selected as the output. To improve stability during real-time recognition, a confidence threshold of 0.85 is applied. Predictions below this threshold are ignored. Additionally, a stable-frame mechanism confirms a gesture only after it has been detected consistently for multiple consecutive frames. This reduces incorrect predictions caused by hand movement or temporary occlusion.

The recognized gesture is displayed as text on a Flask-based web interface. Special gesture classes such as Space and Delete enable users to form complete words and edit generated text. Finally, the recognized sentence is converted into speech using a text-to-speech module, allowing non-sign language users to understand the communication.

IV. SYSTEM ARCHITECTURE

The proposed architecture integrates computer vision, deep learning, and web technologies into a single real-time application.

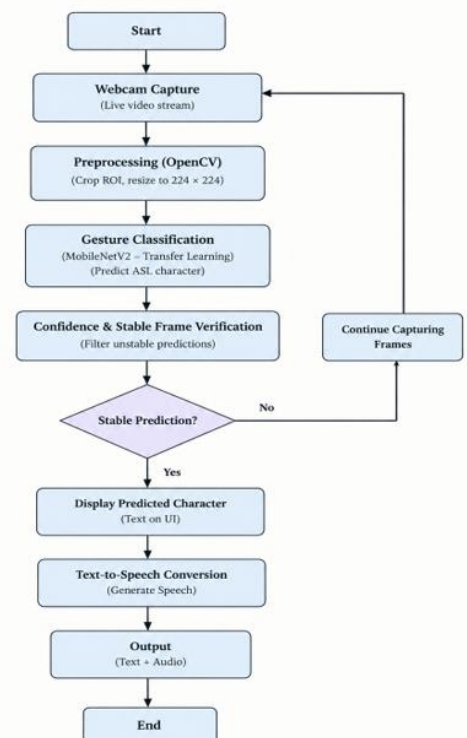


Figure 1. System Architecture.

The major components of the system are described below.

A. Image acquisition

The webcam continuously captures live video frames. Each frame is horizontally flipped to provide a natural user experience and then processed for gesture recognition.

B. Image preprocessing

Image preprocessing improves recognition accuracy before classification. The preprocessing stage includes:

- ROI extraction
- Rescaling the image to 224×224 pixels
- Pixel normalization
- Tensor conversion

These operations reduce unnecessary background information and prepare the image for deep learning inference.

C. Mobilenetv2 gesture classification

The classifier uses a MobileNetV2 network adapted through transfer learning and trained with the ASL alphabet data. Its compact design and relatively low computational requirement make it appropriate for real-time recognition, while the network predicts the gesture class with the greatest confidence.

- Alphabet A–Z
- Space
- Delete
- Nothing

The predicted class with the highest confidence score is selected as the output.

D. Flask-Based Deployment

The trained model is integrated into a Flask application to enable real-time prediction. Flask handles communication between the deep learning model and the user interface by providing APIs for:

- Live prediction
- Recognition history
- Confidence score
- Generated sentence
- Recording controls

E. User interface

The frontend is developed using HTML and CSS. It displays:

- Live webcam feed
- Predicted alphabet
- Confidence percentage

- Recognition history
- Generated sentence
- Recording status

The interface also provides buttons to start recognition, stop recognition, clear the generated text, and convert the recognized sentence into speech.

V. DATASET DESCRIPTION

Training was carried out with the American Sign Language (ASL) Alphabet Dataset, a labeled collection of images covering different hand gestures. It contains 29 categories: the 26 English alphabet classes plus Space, Delete, and Nothing.

The images represent different lighting settings and hand orientations, providing variation for learning visual patterns. Prior to training, each image was resized to 224×224 pixels and normalized to support stable optimization.

Data augmentation techniques were applied during training to increase dataset diversity and improve the model's ability to generalize to unseen hand gestures.

VI. MODEL TRAINING

The MobileNetV2 model was initialized using pretrained ImageNet weights. Transfer learning was adopted to reduce training time and improve classification performance. The original classification layer was replaced with a custom output layer corresponding to the 29 ASL gesture classes.

Training was performed in Google Colab using TensorFlow/Keras. Categorical cross-entropy was used as the optimization loss, while validation accuracy was monitored throughout training to assess generalization and limit overfitting.

After training, the optimized model was saved and integrated into the Flask application for real-time prediction. During deployment, each webcam frame undergoes preprocessing before being passed to the trained model for classification.

VII. EXPERIMENTAL RESULTS AND DISCUSSION

The sign language recognition application was evaluated with the ASL alphabet dataset. Model development was completed in Google Colab with

TensorFlow/Keras, and live recognition was tested through the Flask webcam interface. The evaluation considered training and validation accuracy, loss curves, the confusion matrix, precision, recall, and F1-score.

The adapted MobileNetV2 network reached an overall classification accuracy of 95%. During webcam testing, predictions remained stable with low response delay, supporting the intended real-time communication use case.

A. Training and validation accuracy

Figure 2 shows the accuracy recorded during training and validation. Both curves improve across the training epochs, while the validation curve remains close to the training curve. This behaviour suggests that the learned model generalizes reasonably well without substantial overfitting. The final validation accuracy is approximately 95%, supporting the use of transfer learning for the task.

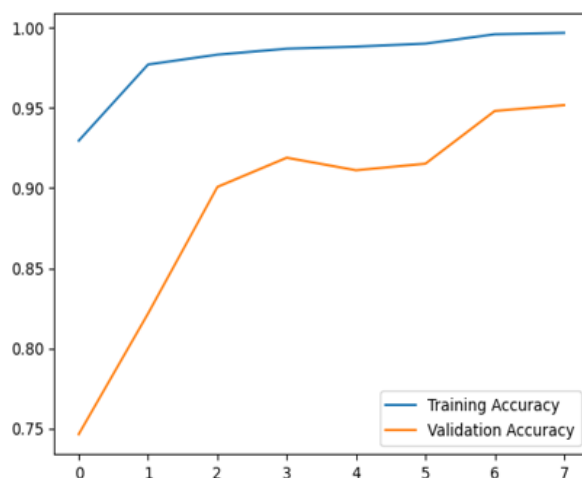


Figure 1. Training and Validation Accuracy Curves

B. Training and validation loss

Figure 3 presents the loss behaviour for the training and validation sets. The two loss values generally decline as optimization progresses, indicating that the network parameters are being learned successfully. Small changes in validation loss appear near the final epochs, but the overall pattern remains stable, suggesting satisfactory convergence and generalization.

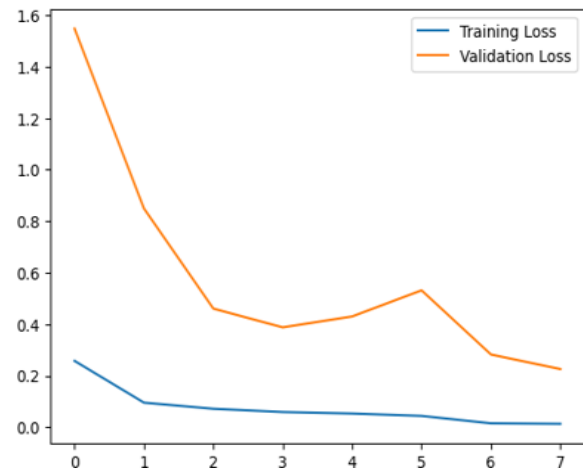


Figure 3. Training and Validation Loss Curves

C. Confusion matrix analysis

Figure 4 shows the confusion matrix produced by the trained classifier. The strongest counts appear along the main diagonal, showing that most ASL alphabet categories are recognized correctly. The remaining errors are concentrated among visually similar gestures, where finger arrangements can be difficult to distinguish.

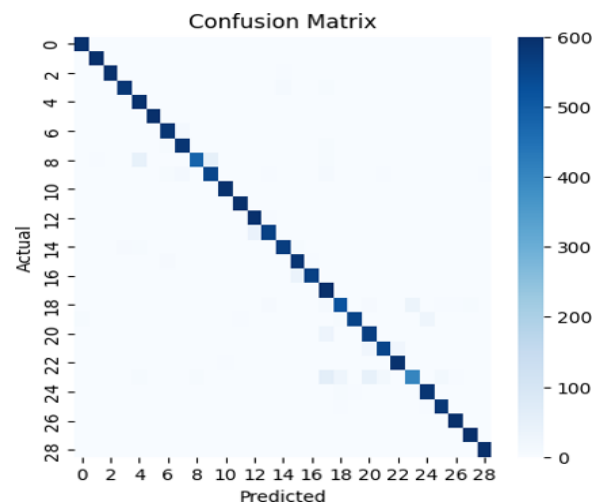


Figure 4. Confusion Matrix Analysis

Overall, the confusion matrix confirms that the proposed model performs reliably across different gesture categories and provides accurate real-time predictions.

D. Precision, recall and f1-score

Figure 5 compares precision, recall, and F1-score across the evaluation. The metrics remain relatively

close to one another, indicating a balanced relationship between false-positive and false-negative outcomes. Together, these measures provide evidence of consistent classification behaviour for the MobileNetV2-based system.

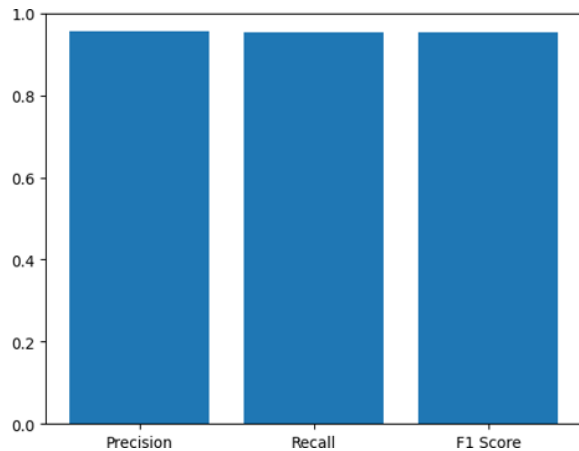


Figure 5. Precision, Recall and F1-Score Metrics

E. Real-time performance

The real-time implementation successfully integrates webcam image acquisition, gesture recognition, and speech generation into a single application. The confidence threshold of 0.85 and stable-frame validation mechanism significantly reduce incorrect predictions during continuous recognition. The Flask-based interface provides smooth interaction by displaying the live video feed, recognized alphabet, confidence score, prediction history, and generated sentence. The integrated text-to-speech module further enhances accessibility by converting recognized text into audible speech.

VIII. ADVANTAGES OF THE PROPOSED SYSTEM

The proposed system offers several practical advantages over conventional sign language recognition methods:

- Lightweight MobileNetV2 architecture enables efficient real-time inference.
- Webcam-based recognition eliminates the need for wearable sensor gloves.
- High recognition accuracy of approximately 95%.
- User-friendly web interface for live interaction.
- Integrated text-to-speech functionality improves accessibility.

- Confidence filtering reduces unstable predictions.
- Cost-effective implementation using open-source technologies.

IX. CONCLUSION

This work developed a real-time sign language recognition and speech conversion application based on MobileNetV2 transfer learning. The framework combines computer vision, deep learning, and web technologies to identify ASL hand gestures from webcam input and produce text and speech. OpenCV performs the image preparation stage, while TensorFlow/Keras supports classification with the pretrained MobileNetV2 network.

Confidence filtering and stable-frame verification help make live predictions more consistent. The experimental evaluation produced an overall classification accuracy of 95%, indicating that the system can support real-time gesture recognition. The Flask interface provides an accessible interaction layer and allows the application to be used in educational, healthcare, and public-service contexts.

The developed application demonstrates how artificial intelligence can contribute to improving accessibility and reducing communication barriers for individuals with hearing and speech impairments.

X. FUTURE SCOPE

Although the proposed system performs effectively for ASL alphabet recognition, several enhancements can further improve its functionality:

- Formation of complete words and continuous sign-language sentences.
- Support for dynamic hand gestures involving motion.
- Mobile application development for Android and iOS devices.
- Integration with multilingual translation systems.
- Cloud-based deployment for remote accessibility.
- Improved robustness under varying lighting conditions.
- Automatic hand detection without requiring a fixed Region of Interest.

These enhancements can expand the usability of the proposed framework in real-world assistive communication systems.

REFERENCES

- [1] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [2] Howard, A. G., et al. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications.
- [3] TensorFlow Developers. *TensorFlow Documentation*. TensorFlow Documentation
- [4] OpenCV Developers. *OpenCV Documentation*. OpenCV Documentation
- [5] Flask Development Team. *Flask Web Framework Documentation*. Flask Documentation
- [6] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- [7] Szeliski, R. (2022). *Computer Vision: Algorithms and Applications*. Springer.
- [8] Kaggle. *ASL Alphabet Dataset*.
- [9] Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization.
- [10] Chollet, F. (2018). *Deep Learning with Python*. Manning Publications.