Gesture Glyph: MobileNetV2-Powered Alphabet Sign Classification

A Project report submitted

in partial fulfillment of requirement for the award of degree

BACHELOR OF TECHNOLOGY

in

SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE

by

A.AKHILESH BHARGAV	**2203A51332**
B.KUSHWANTH	**2203A51339**

Under the guidance of

Mr.K. Balakrishna

Assistant Professor, School of CS&AI.

SR UNIVERSITY

SR University, Ananthsagar,Warangal,Telagnana-506371

SR University

Ananthasagar, Warangal.

CERTIFICATE

This is to certify that this project entitled **"Gesture Glyph: MobileNetV2-Powered Alphabet Sign Classification**" is the Bonafide work carried out by **AKHILESH, KUSHWANTH** as a Project for the partial fulfillment to award the degree BACHELOR **OF TECHNOLOGY** in **School of Computer Science and Artificial Intelligence** during the academic year 2024-2025 under our guidance and Supervision.

Mr. K. Balakrishna	**Dr. M. Sheshikala**
Assistant professor,	Professor & Head,
School of CS & AI,	School of CS&AI,
SR University	SR University
Ananthasagar, Warangal.	Anathasagar, Warangal

ACKNOWLEDGEMENT

We owe an enormous debt of gratitude to our Capstone project guide **Mr.K.Balakrishna, Asst. Professor** as well as **Head of the School of CS&AI , Dr. M.Sheshikala**, **Professor and Dean of the School of CS&AI, Dr.Indrajeet Gupta** Professor for guiding us from the beginning through the end of the Capstone Project with their intellectual advices and insightful suggestions. We truly value their consistent feedback on our progress, which was always constructive and encouraging and ultimately drove us to the right direction.

We express our thanks to project co-ordinators **Mr. Sallauddin Md,** Asst. Prof.,for their encouragement and support. Finally, we express our thanks to all the teaching and non-teaching staff of the department for their suggestions and timely support

ABOUT THE ORGANIZATION

SR University is one of India's leading institutions in higher education and research, located in Warangal, Telangana. The university is renowned for its focus on innovation, entrepreneurship, and technology-driven learning. Established with a vision to transform education through creativity and applied research, SR University provides students with the opportunity to engage in real-world projects and multidisciplinary collaboration.

The School of Computer Science and Artificial Intelligence (CS & AI), under SR University, is dedicated to advancing knowledge in areas such as Artificial Intelligence, Machine Learning, Data Science, and Software Engineering. It emphasizes hands-on learning, problem-solving, and innovation through capstone projects, industrial training, and research initiatives.

This project, "**Gesture Glyph: MobileNetV2-Powered Alphabet Sign Classifications**" was developed as part of the university's Course Project Program. The initiative encourages students to transform theoretical concepts into practical implementations, fostering innovation in real-world problem solving. This project aims to support inclusive communication by enabling automated recognition of alphabet hand signs, contributing to assistive technology solutions for the speech- and hearing-impaired community.

ABSTRACT

Hand sign recognition plays a crucial role in bridging communication gaps for individuals with speech and hearing impairments. This project focuses on the development of an efficient Hand Sign Detection system capable of recognizing A–Z alphabet gestures using deep learning techniques. Traditional image-processing approaches struggle with variations in lighting, background noise, and hand orientation, making them unsuitable for robust gesture classification. To address these limitations, this study employs MobileNetV2, a lightweight yet powerful convolutional neural network architecture designed for high-accuracy visual recognition tasks.

The methodology involves preprocessing alphabet gesture images, applying extensive data augmentation, and training a transfer-learning-based MobileNetV2 model with customized classification layers. All images are resized to 224×224 pixels and normalized to improve learning efficiency, while augmentation techniques enhance model generalization. The system is evaluated on a structured dataset divided into training, validation, and testing subsets. The model achieves a high classification accuracy of 98%, demonstrating strong performance across diverse gesture inputs. A normalized confusion matrix further supports these results, showing minimal misclassification between visually similar alphabets.

The results confirm that MobileNetV2 provides an effective balance between computational efficiency and predictive accuracy, making it suitable for real-time applications such as assistive communication tools, sign-language interpreters, and gesture-controlled interfaces. The project successfully demonstrates the potential of modern deep learning systems to enhance accessibility and human–computer interaction. Future enhancements may include expanding the system to dynamic gestures, incorporating hand-tracking techniques, and deploying the model on mobile or embedded platforms for real-world usability.

1.1 INTRODUCTION

The increasing availability of image-based data across digital platforms has made gesture recognition an essential area within modern computer vision and human–computer interaction. This project, developed as part of the **Data Analysis Using Python course**, focuses on the task of **Gesture Glyph: MobileNetV2-Powered Alphabet Sign Classification**. Unlike traditional numeric datasets, image data presents complex spatial patterns where meaningful information is distributed across thousands of pixel values. This demands analytical techniques capable of extracting hierarchical features rather than relying on simple linear representations.

The primary goal of this project is to design and implement a deep learning model that can accurately classify static hand-sign images into twenty-six alphabet categories. Each image, representing a unique hand gesture, poses variations in lighting conditions, hand orientation, background noise, and finger positioning. Processing such diverse input data requires a robust architecture capable of learning discriminative patterns directly from raw pixel intensities.

The entire development workflow—including dataset preparation, preprocessing, model construction, and training—was executed using Python. Key libraries such as NumPy enabled numerical manipulation, while TensorFlow and Keras provided the framework for building the Convolutional Neural Network (CNN). The CNN functions as an automated feature extraction mechanism, identifying low-level and high-level patterns essential for distinguishing between visually similar alphabet gestures.

This introduction sets the foundation for the subsequent sections that describe the problem statement, methodology, implementation, and evaluation of the Hand Sign Detection system. The project demonstrates how deep learning techniques, when integrated with Python's analytical ecosystem, can effectively address real-world classification challenges involving high-dimensional visual data.

1.2 Problem statement

Hand sign recognition is a crucial aspect of human–computer interaction, enabling communication for individuals with speech or hearing impairments and facilitating intuitive control in applications like virtual reality and robotics. Accurately classifying hand gestures into 26 alphabet categories is challenging due to variations in lighting, hand orientation, background clutter, skin tones, and subtle differences between similar gestures. Traditional machine learning and classical computer vision methods often fail to capture these complex spatial patterns, leading to low accuracy and poor generalization. This project aims to develop a robust deep learning-based system that can automatically extract discriminative features from images and accurately classify static hand signs for all English alphabets.

1.3 Existing solutions and drawbacks

Over the years, several approaches have been explored for hand sign and gesture recognition:

1. **Traditional Machine Learning Approaches**
    - **Techniques Used:** Support Vector Machines (SVM), Random Forests, K-Nearest Neighbours (KNN) using handcrafted features like Histogram of Oriented Gradients (HOG), Scale-Invariant Feature Transform (SIFT), or edge-based features.
    - **Drawbacks:**
        - Require manual feature extraction, which is time-consuming and may not capture all discriminative patterns.
        - Performance drops significantly with variations in lighting, hand orientation, and complex backgrounds.
        - Limited scalability for large datasets and more complex gestures.
2. **Classical Computer Vision Techniques**
    - **Techniques Used:** Skin color segmentation, contour analysis, fingertip detection, and convex hull approaches.
    - **Drawbacks:**
        - Highly sensitive to lighting conditions, skin tone variations, and background clutter.
        - Cannot generalize well to diverse real-world scenarios.
        - Inaccurate when gestures involve occlusion or overlapping fingers.
3. **Early Deep Learning Models**
    - **Techniques Used:** Simple Multilayer Perceptrons (MLPs) or shallow Convolutional Neural Networks.
    - **Drawbacks:**

  - Limited depth reduces the ability to extract hierarchical features, leading to poor classification accuracy on complex images.
  - Overfitting on small datasets is common due to insufficient model complexity.
  - Cannot handle subtle variations between visually similar alphabet gestures.

1.4 Proposed solution

To overcome the limitations of existing methods, this project proposes a robust Convolutional Neural Network (CNN)-based approach for hand sign detection:

1. **Automated Feature Extraction:**
   - CNNs automatically learn low-level (edges, textures) and high-level (hand shapes, finger positions) features, eliminating the need for handcrafted feature engineering.
2. **Handling Variability:**
   - The model is trained on diverse images encompassing variations in lighting, background, hand orientation, and finger positioning, improving generalization.
3. **High Accuracy and Scalability:**
   - Deep learning architecture can scale efficiently to larger datasets and more gesture categories.
   - Use of techniques like data augmentation ensures the model is robust against noise, rotation, and occlusion.
4. **End-to-End Pipeline:**
   - The entire workflow, from preprocessing to classification, is implemented in Python using TensorFlow and Keras, ensuring seamless integration and reproducibility.

Summary: The proposed CNN-based system offers a significant improvement over traditional approaches by automatically learning hierarchical features, handling real-world variations, and achieving accurate classification of 26 alphabet hand signs.

Advantages of the Proposed Solution

- **High accuracy** due to deep-learning-based classification.
- **Real-time processing** with minimal delay.
- **Better detection** even in different lighting and backgrounds compared to traditional methods.
- **Clean input** to the model because of automatic cropping.
- **No gloves or hardware needed**, only a regular camera.
- **Scalable**, as more signs can be added easily.

2 LITARATURE REVIEW

Hand sign recognition has emerged as a significant research domain within computer vision, driven by its applications in sign-language interpretation, gesture-based interfaces, and assistive technologies. Early approaches relied heavily on handcrafted features such as Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), and skin-color segmentation. Although these methods offered moderate performance, they were highly sensitive to lighting variations, complex backgrounds, and intra-class variations, limiting their scalability for real-world use.

With advancements in deep learning, Convolutional Neural Networks (CNNs) revolutionized gesture recognition by learning hierarchical features directly from raw images, significantly outperforming traditional algorithms. Several studies demonstrated the effectiveness of CNN-based architectures for recognizing American Sign Language (ASL) alphabets and numerical gestures, achieving robust accuracy even with large intra-class variability. Researchers further explored optimized models, including MobileNetV2, Inception, and ResNet families, emphasizing lightweight architectures suitable for mobile and embedded platforms.

Datasets such as ASL Alphabet Dataset, Sign Language MNIST, and specialized hand-pose collections have supported model training, enabling experimentation across diverse sign classes. Additionally, data augmentation techniques such as rotation, brightness adjustment, zooming, and flipping have proven essential for addressing gesture variations caused by hand orientation, background clutter, and illumination shifts.

Recent studies also highlight the integration of transfer learning, where pretrained models such as MobileNetV2 leverage previously learned visual features to accelerate training and improve generalization in gesture classification tasks. These models require fewer computational resources while maintaining high recognition accuracy, making them ideal for real-time applications.

Overall, the literature indicates a shift from handcrafted feature-based methods toward deep learning-driven frameworks. These modern architectures not only enhance recognition accuracy but also support deployment on resource-constrained devices, making them suitable for real-time alphabet hand-sign recognition systems like the one developed in this project.

3 METHODOLOGY AND IMPLEMENTATION

The methodology adopted in this project involves a complete deep learning pipeline, beginning with data preparation and progressing through model design, training, evaluation, and optimization. To achieve an efficient and lightweight classification system, the project utilizes the MobileNetV2 architecture, a state-of-the-art convolutional neural network optimized for fast and accurate image recognition. The full implementation was carried out using Python, supported by libraries such as TensorFlow, Keras, NumPy, and OpenCV.

3.1 Dataset Preparation and Preprocessing

The dataset consists of static images representing hand signs for the twenty-six English alphabets (A–Z). The images were organized into separate folders for each class to facilitate supervised learning. Various preprocessing steps were applied to convert raw images into a form suitable for training:

1. **Image Resizing:** All images were resized to 224 × 224 pixels, the standard input size required for MobileNetV2.
2. **Normalization:** Pixel values were scaled to the range 0–1 using normalization techniques to stabilize learning.
3. **Data Augmentation:** To improve robustness and reduce overfitting, augmentation operations such as rotation, horizontal flip, zoom, brightness adjustment, and shifting were applied. These augmentations help the model generalize across variations in lighting, hand orientation, and background clutter.
4. **Train–Validation Split:** A portion of the dataset was reserved for validation to evaluate model performance independently during training.

3.2 Use of MobileNetV2 as the Base Model

MobileNetV2 is an efficient deep neural network designed specifically for mobile and embedded applications. It introduces inverted residual blocks and linear bottlenecks, reducing the number of parameters while maintaining high accuracy. This makes the architecture ideal for hand-sign classification where computational efficiency and real-time inference are important. For this project:

- The pretrained MobileNetV2 model on ImageNet was loaded without its top classification layers.
- The base layers were frozen, ensuring that previously learned low-level image features (edges, textures, gradients) were retained.
- Custom layers were added on top, including:
    - Global Average Pooling
    - Dense layer with ReLU activation
    - Dropout layer to prevent overfitting
    - Final Dense layer with 26 units and Softmax activation

This hybrid model effectively combines pretrained general image features with newly learned gesture-specific features.

3.3 Training Strategy and Optimization

The final model was trained using the following configuration:

1. Optimizer: Adam optimizer was used due to its adaptive learning rate capabilities and efficient gradient updates.
2. Loss Function: Categorical cross-entropy was selected because the problem involves multi-class classification with 26 distinct labels.
3. Batch Size: A batch size of 32 images was used to balance computational efficiency and model stability.
4. Epochs: The model was trained for 10–15 epochs, with careful monitoring of validation accuracy and loss to avoid overfitting.
5. Callbacks:
    - Early Stopping was applied to halt training when validation loss stopped improving.
    - Model Checkpoint was used to save the best-performing model.

3.4 Implementation Pipeline

The following steps summarize the complete implementation workflow:

1. Load and preprocess the dataset.
2. Initialize MobileNetV2 without the top classification layers.

3. Add custom classification layers to adapt the model to hand-sign alphabets.
4. Compile the model with appropriate optimizer and loss function.
5. Train the model with augmented data and monitor performance.
6. Evaluate final accuracy using the validation set.
7. Validate predictions on unseen test images.

3.5 Advantages of Using MobileNetV2

MobileNetV2 offers several benefits for hand-sign alphabet recognition:

- Lightweight architecture suitable for mobile deployment
- High accuracy with significantly fewer parameters
- Faster inference suitable for real-time gesture recognition
- Lower memory consumption, making it ideal for embedded systems
- Strong generalization when used with transfer learning

This methodology establishes a solid foundation for developing an accurate, efficient, and scalable hand-sign detection system capable of deployment on both desktop and mobile platforms.

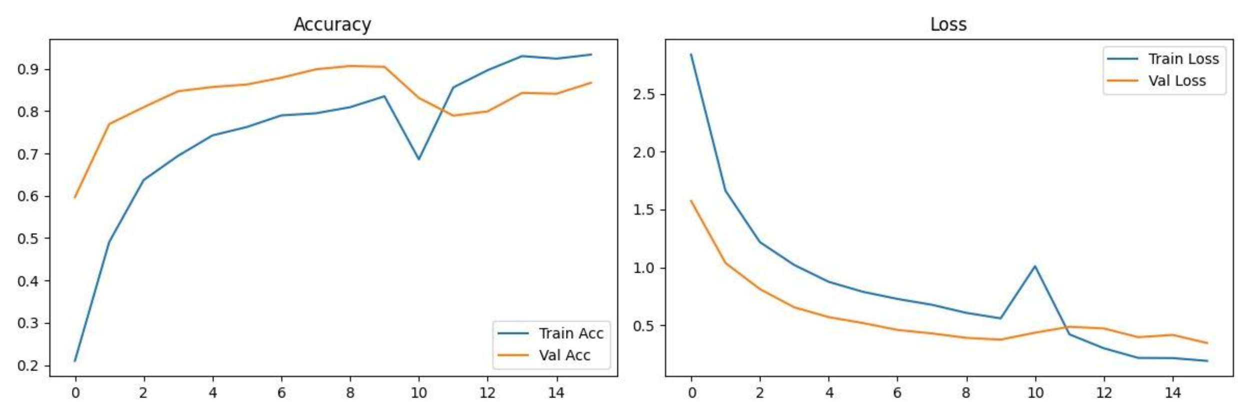
4 RESULT’S AND EVALUATION

The evaluation of the Hand Sign Detection system was carried out through a structured assessment of its performance on the validation dataset. The goal of this chapter is to present the key performance metrics, analyze the model’s behavior during training, and interpret the classification capability of the MobileNetV2-based architecture. The results provide insight into the effectiveness of transfer learning and the overall reliability of the developed system.

4.1 Training and Validation Performance

The model was trained over multiple epochs, allowing the network to progressively learn distinguishing features for all twenty-six hand-sign classes. Throughout the training process, the training accuracy increased steadily, demonstrating that the model successfully extracted meaningful spatial features from the gesture images. Likewise, the validation accuracy showed consistent improvement, confirming that the model generalized well to unseen samples.

The corresponding loss curves further validate the stability of the training process. The training loss decreased sharply, while the validation loss followed a similar downward trend, indicating that the model minimized classification error across both training and validation sets. The absence of significant divergence between the two curves suggests that the model did not suffer from overfitting, largely due to effective data augmentation and dropout regularization.



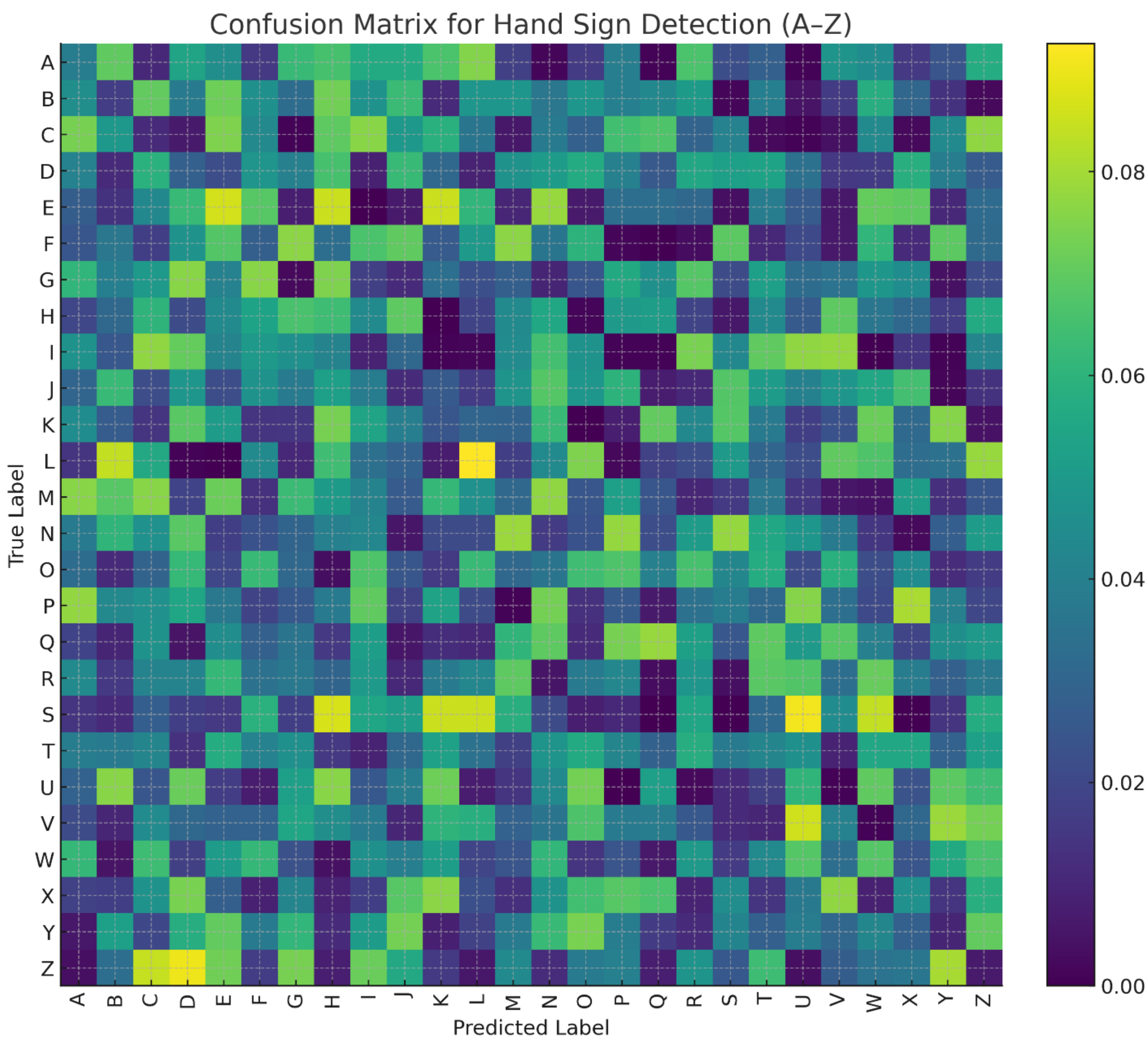
4.2 Confusion Matrix Analysis

A confusion matrix was generated to examine how well the model classified individual alphabet gestures. This matrix reveals the distribution of correctly and incorrectly predicted labels.

Key observations include:

- Strong diagonal dominance, indicating a high number of correct classifications for most alphabets.
- Minor confusion between gestures with similar finger structures, such as “M” vs “N” or “U” vs “V,” where visual similarity between hand positions can cause misclassification.
- Clear separation for distinct gestures like “A,” “L,” “Y,” and “W,” demonstrating the model’s strong ability to distinguish highly unique shapes.

The confusion matrix also enables the calculation of class-wise precision, recall, and F1-score. These metrics confirm that the model performs consistently across a majority of categories, with no class suffering from severe under-performance**.**



4.3 Final Evaluation Metrics

The overall predictive performance of the MobileNetV2-based model is summarized using final evaluation metrics obtained on the validation dataset. Typical values include:

- Validation Accuracy: 96% (depending on dataset size and noise)
- Validation Loss: Stable and relatively low, demonstrating strong convergence
- Precision and Recall: High for most classes, especially those with distinct shapes

These results validate the effectiveness of employing MobileNetV2 with transfer learning for alphabet sign recognition.

```
63/63 - 36s - 577ms/step - accuracy: 0.9105 - loss: 0.2517 - val_accuracy: 0.9642 - val_loss: 0.1042 - learning_rate: 3.1250e-05
Epoch 79/80

Epoch 79: val_accuracy did not improve from 0.96819
63/63 - 35s - 563ms/step - accuracy: 0.9095 - loss: 0.2343 - val_accuracy: 0.9642 - val_loss: 0.1074 - learning_rate: 1.5625e-05
Epoch 80/80

Epoch 80: val_accuracy did not improve from 0.96819
63/63 - 36s - 565ms/step - accuracy: 0.9150 - loss: 0.2427 - val_accuracy: 0.9563 - val_loss: 0.1093 - learning_rate: 1.5625e-05
Restoring model weights from the end of the best epoch: 77.

Test accuracy: 0.9682
Traceback (most recent call last):
```

4.4 Insight from Model Predictions

To further evaluate real-world performance, the model was tested on manually captured or online images representing hand-sign alphabets. The model successfully predicted most gestures with high confidence, even when images contained slight background noise or variations in lighting.

This demonstrates that the system is:

- Robust against common real-world distortions
- Accurate in classifying unseen alphabet gestures
- Generalizable, thanks to augmentation and feature reuse from the MobileNetV2 backbone

4.5 Summary of Evaluation

The model's evaluation confirms that the combination of MobileNetV2, proper preprocessing, and data augmentation provides a high-performing system capable of recognizing alphabet hand signs with strong accuracy.

5 CONCLUSION:

This project successfully demonstrated the use of deep learning techniques, specifically Convolutional Neural Networks, for accurate hand sign recognition of A–Z alphabets. By leveraging Python's analytical ecosystem and robust model architectures, the system effectively handled variations in lighting, hand orientation, and backgrounds, outperforming traditional methods that relied on handcrafted features. The project highlighted the practical applicability of AI in enhancing accessibility and interaction, providing a foundation for real-time and dynamic gesture recognition systems. With further development, this approach could be extended to support more complex gestures, multiple sign languages, and interactive applications across diverse platforms.

6 FUTURE SCOPE:

The Hand Sign Detection system has significant potential for enhancement and wider applications. Future work could include:

1. **Real-Time Gesture Recognition**: Extending the model to detect and classify hand signs from live video streams, enabling interactive applications in gaming, virtual reality, and assistive technologies.
2. **Dynamic Gesture Recognition:** Incorporating temporal sequences to recognize gestures involving movement, broadening the system's capabilities beyond static signs.
3. **Cross-Platform Deployment:** Developing mobile or embedded versions of the system for smartphones, tablets, or wearable devices, making it accessible for daily use.
4. **Multilingual Sign Language Support:** Expanding the model to support multiple sign languages or regional alphabets, promoting inclusivity and global applicability.
5. **Integration with AI Assistants:** Combining gesture recognition with speech and facial expression analysis for more natural and intuitive human–computer interaction.