

**HandCode: Deep Learning-Based Recognition of American Sign Language  
Alphabet Using Convolutional Neural Networks**

In Fulfilment of the Final Alternative Assessment for the course  
Intelligent Systems (CCS 229)

Submitted to:

**Louie F. Cervantes**

Management Information System Office

West Visayas State University – La Paz, Iloilo City

**Submitted by:**

Nava, Angelika Marie B.

**BSCS 3A - AI**

May 2025

# PROJECT OVERVIEW AND OBJECTIVES

## I. Project Overview

Communication is a fundamental human need, and it serves as the pinnacle of social participation and inclusion in today's society. For members of the Deaf and hard-of-hearing (DHH) communities, sign language, particularly American Sign Language (ASL), renders itself as a primary mode of communication. However, the ever-persistent language barrier between ASL users and the broader community often poses challenges in services, social interactions, and professional opportunities [1].

Recent advancements in artificial intelligence (AI) and deep learning have revolutionized the development of assistive technologies aimed at bridging these communication gaps. Automated sign language recognition (SLR) systems, which maximize algorithms such as Convolutional Neural Networks (CNNs), have shown remarkable efficacy in interpreting static and dynamic hand gestures [2], [3]. These systems are increasingly capable of real-time translation, transforming hand poses into text or speech, therefore enhancing accessibility and fostering inclusivity for the Deaf and DHH community [1].

This project, titled **HandCode**, is situated within this rapidly evolving research landscape. Its principal objective is the design, development, and deployment of an image classification system that recognizes static hand gestures corresponding to the

ASL alphabet. The system is grounded in deep learning principles, specifically employing CNNs, which are well-suited for image processing due to their ability to automatically extract hierarchical spatial features [3], [4]. The model is trained on a publicly available ASL Alphabet Dataset, which includes thousands of labeled images representing each of the 26 letters of the English alphabet.

By integrating modern machine learning techniques with a user-friendly interface, HandCode aims to contribute to the growing field of AI-assisted communication tools. The long-term vision of such systems is to provide real-time translation and interpretation services, thus fostering inclusivity and reducing communication barriers for ASL users. This paper presents the full development lifecycle of HandCode, from dataset preparation and model training to deployment and evaluation, offering insights into both the capabilities and limitations of current deep learning approaches to sign language recognition [2], [4], [5].

## **II. Objectives**

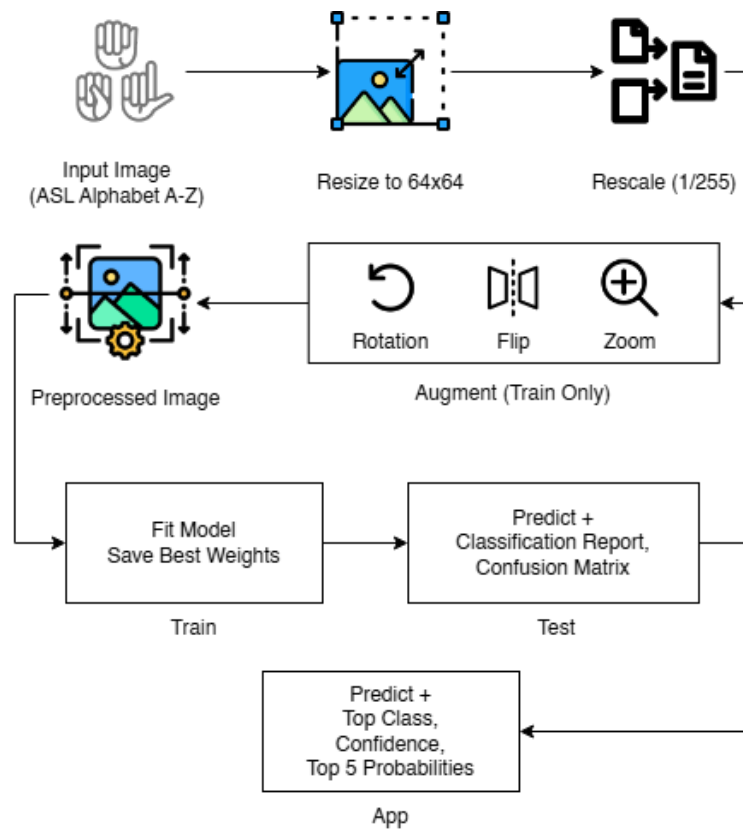
Generally, this project aims to design, develop, and deploy an image classification system capable of recognizing static hand gestures representing the ASL alphabet.

Specifically, this project's methodology is structured around four core objectives:

1. To construct a deep learning architecture capable of effectively recognizing ASL hand gestures from images with high accuracy;

2. To train, validate, and test the model using a balanced image dataset that reflects real-world variability in gesture representation;
3. To deploy the trained model in an interactive and accessible application interface using the Streamlit framework, enabling end-users to input gesture images and receive immediate classification result, and;
4. To assess the performance of the model using quantitative metrics such as training and validation accuracy, loss curves, precision, recall, and confusion matrices.

## MODEL ARCHITECTURE DIAGRAM



**Figure 1. Model Architecture Diagram**

The diagram presents a comprehensive overview of an American Sign Language (ASL) Recognition System architecture, delineating the principal stages of data preprocessing, model training, evaluation, and inference. The process initiates with an input image, which is a single frame capturing an ASL letter hand gesture. To ensure uniformity and compatibility with neural network processing, each image is first resized to a fixed dimension of 64x64 pixels. This standardization is critical for preventing input shape mismatches that could otherwise compromise model performance. Subsequently,

pixel intensity values, which originally range from 0 to 255, are normalized by dividing each value by 255, thereby constraining pixel values to the interval. This normalization step not only facilitates faster convergence during training but also stabilizes the gradient descent process, contributing to more reliable model optimization.

During the training phase, the system employs data augmentation techniques such as random rotation, zooming, and flipping of images. These transformations artificially expand the diversity of the training dataset, enabling the model to learn robust representations of ASL letters under varying orientations and scales. This approach is instrumental in enhancing the model's ability to generalize to real-world scenarios, where hand gestures may be presented from different angles or under varied lighting conditions. The preprocessed images, now augmented and normalized, are fed into a CNN, which serves as the core of the ASL recognition system. The CNN is specifically designed to extract hierarchical spatial features from images, allowing it to learn distinctive patterns associated with each ASL letter.

The model pipeline is structured into three key phases: training, testing, and inference. In the training phase, the CNN is optimized by minimizing a loss function through backpropagation, with model weights being updated based on the discrepancy between predicted and true labels. The best-performing model parameters, determined by validation accuracy or loss, are retained for subsequent use. Following training, the model is rigorously evaluated on a separate test set. This evaluation yields a classification report containing key metrics such as accuracy, precision, recall, and

F1-score, as well as a confusion matrix that visually delineates the model's performance across different ASL letters. These analyses are essential for identifying potential areas of confusion or misclassification.

In the inference phase, the system is deployed as an interactive application, where users can submit new ASL letter images for classification. For each input, the model generates a top predicted class—corresponding to the most likely ASL letter—along with a confidence score and a ranked list of the top five probable classes. This granular output not only enhances user experience but also provides developers with valuable insights into model behavior, facilitating ongoing refinement and interpretability. Collectively, the diagram encapsulates the entire ASL recognition workflow, underscoring the importance of data standardization, augmentation, and model interpretability in the development of a robust, real-world ASL recognition system.

## **DATASET DESCRIPTION**

The ASL Alphabet Dataset is a publicly available image dataset curated and uploaded by grassknotted on Kaggle [5]. It is specifically designed to support research in the field of ASL recognition, particularly for static gesture classification tasks. The dataset contains a total of 87,000 color images, each with a resolution of 200x200 pixels in RGB format, ensuring sufficient visual detail for deep learning model training and evaluation.

The dataset is organized into 29 classes, representing the 26 letters of the English alphabet (A to Z), as well as three additional classes: "del" (delete), "nothing", and "space", which are commonly used control gestures in ASL communication systems. Each class comprises approximately 3,000 images, contributing to a balanced distribution that supports effective training of machine learning models without significant class imbalance issues.

The images in the ASL Alphabet Dataset feature a variety of hand shapes, orientations, skin tones, and lighting conditions, enhancing the dataset's diversity and realism. These variations make the dataset an ideal resource for training robust models capable of generalizing across different real-world scenarios. The dataset captures hands performing static ASL signs against simple backgrounds, which facilitates focused feature learning while still presenting challenges such as subtle finger position differences and occlusions.



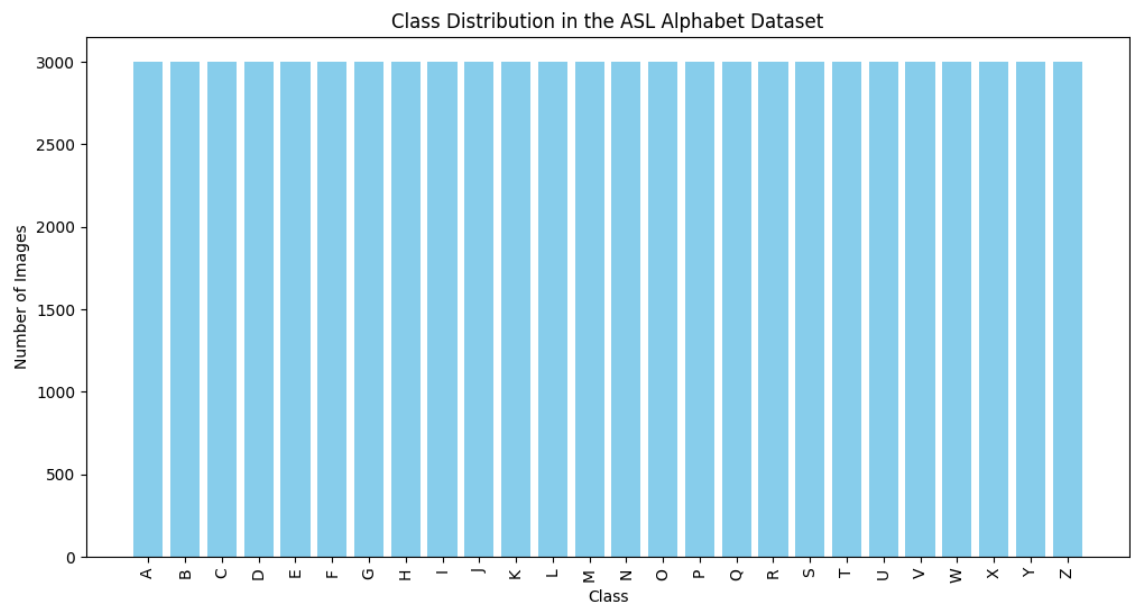
This dataset has been widely used in the computer vision and deep learning communities as a benchmark for ASL recognition systems, including Convolutional Neural Networks (CNNs) and other deep learning architectures. Its accessibility and quality make it a foundational resource for researchers and developers aiming to build ASL classification models for assistive technologies, educational tools, and communication aids for the deaf and hard-of-hearing communities.



**Figure 2. Sample images from the ASL Alphabet Dataset**

Sample images from the dataset, as shown in Figure 2, illustrate the variety of hand poses, lighting conditions, backgrounds, and skin tones captured across the dataset. Each image corresponds to a specific static sign in the ASL alphabet, with variations in hand orientation and position that reflect natural variability encountered in

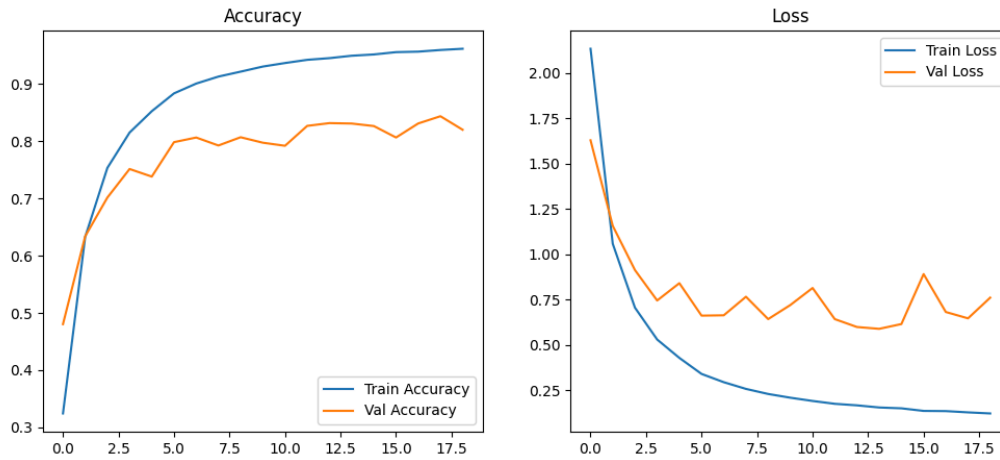
real-world scenarios. This diversity enhances the dataset's utility for training robust machine learning models capable of recognizing ASL signs under different conditions.



**Figure 3. Class distribution in the ASL Alphabet Dataset**

The class distribution within the dataset is presented in Figure 3, demonstrating an even distribution of images across all 29 classes, which includes 26 letters (A–Z) and three additional categories: "del" (delete), "nothing," and "space." Each class contains approximately 3,000 images, resulting in a total of 87,000 images. This balanced representation ensures that the dataset provides sufficient samples for each class, supporting effective model training without the risk of class imbalance that could bias model performance. Together, the sample images and class distribution illustrate the dataset's comprehensiveness and its suitability for developing ASL recognition systems.

## TRAINING LOGS AND CHARTS



**Figure 4. Model Training Logs and Charts**

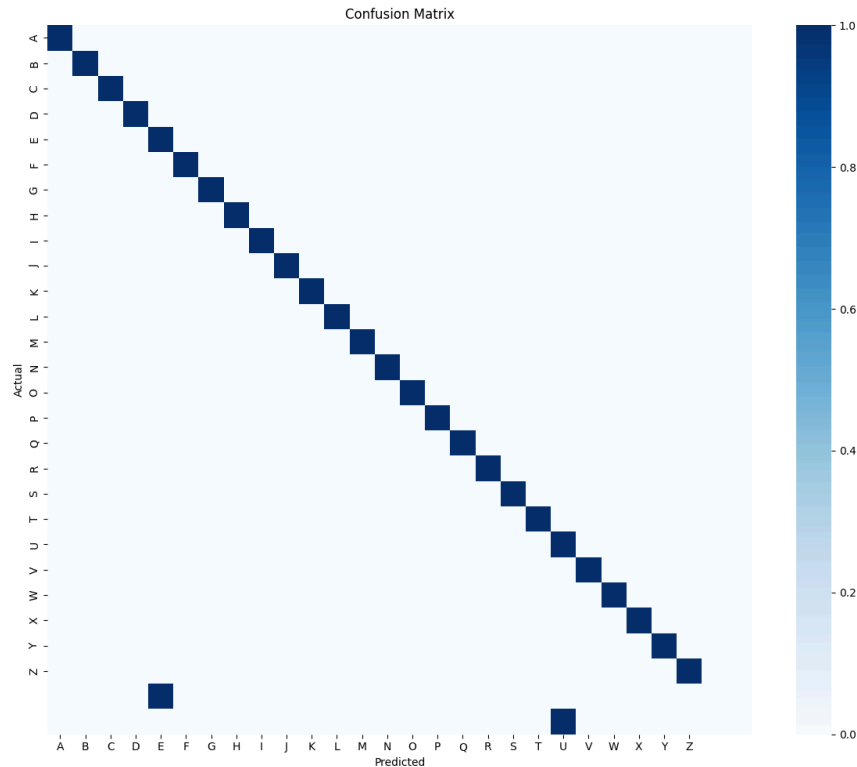
Figure 4 presents the training and validation accuracy and loss metrics for the model trained on the ASL Alphabet dataset. As depicted by the accuracy curves, the model displays consistent improvement in performance for both the training and validation sets as training progresses through successive epochs. Notably, the training accuracy experiences a rapid ascent in the early stages, exceeding 90% by the fifth epoch and eventually stabilizing between 95% and 98% toward the later epochs. This trend reflects the model's successful acquisition of the distinctive patterns characteristic of ASL hand gestures.

Similarly, the validation accuracy exhibits a parallel trajectory, with a steady increase during the initial epochs and eventual stabilization at approximately 85% to 90%. The presence of a modest gap between training and validation accuracy hints at a mild degree of overfitting, a common occurrence in image classification tasks,

particularly when the dataset lacks extensive diversity. Importantly, the absence of a marked divergence between these curves suggests that the model retains a commendable capacity for generalization, without succumbing to severe overfitting.

The loss curves provide complementary insight into the model's learning dynamics. The training loss steadily diminishes, converging toward minimal values as the model assimilates the training data. The validation loss, by contrast, demonstrates a pronounced decrease in the early epochs, indicative of rapid learning, but exhibits minor oscillations in later epochs. These fluctuations may be attributed to inherent variability within the validation set, such as differences in lighting, hand orientation, and background composition, which challenge the model's generalization ability. Despite these perturbations, the validation loss remains relatively stable overall, suggesting that the model is developing robust feature representations rather than simply memorizing the training data.

Collectively, these findings indicate that the model exhibits robust learning behavior and maintains a strong capacity for generalization. Nonetheless, the observed instances of mild overfitting highlight the potential value of integrating additional regularization strategies, such as dropout layers, enhanced data augmentation, or early stopping, to further bolster model robustness. Moreover, the exploration of learning rate scheduling or optimizer tuning may help attenuate fluctuations in validation loss and promote more stable convergence during training.



**Figure 5. Confusion Matrix**

The confusion matrix provides a detailed and quantitative assessment of the model's classification performance across the 26 letters of the English alphabet in American Sign Language (ASL). The matrix is characterized by a strong dominance of values along its main diagonal, which signifies that the model correctly identifies the majority of hand signs with a high degree of accuracy. This pattern reflects the model's robust ability to learn and distinguish the distinct visual patterns, shapes, and orientations that define each letter, resulting in accurate predictions for most classes.

However, a closer examination of the matrix reveals a few off-diagonal entries, indicating specific areas of misclassification. Notably, the model exhibits confusion

between the letters 'E' and 'F' and between 'V' and 'U'. These particular hand signs share visual similarities—such as finger positioning and orientation—that can make them challenging to distinguish, both for the model and even for human annotators. For instance, the letters 'E' and 'F' both involve a partially closed hand shape, while 'V' and 'U' differ mainly in the spread of two fingers, leading to potential ambiguity. These types of errors are commonly observed in fine-grained gesture classification tasks, as subtle differences in finger alignment, occlusion, or camera angle can introduce additional complexity.

Beyond these specific confusions, the absence of widespread errors or systematic biases in the matrix suggests that the model does not overfit to particular classes and maintains a balanced generalization capability across the entire ASL alphabet. This indicates that the model architecture, training strategy, and dataset composition were effective in facilitating comprehensive learning of ASL signs. However, the observed confusions also point to potential avenues for refinement.

Overall, the confusion matrix analysis demonstrates that the model is highly capable of recognizing and classifying ASL hand signs with strong accuracy. The few instances of misclassification underscore the inherent challenges in fine-grained visual classification and provide valuable insights for targeted improvements in future iterations of the model. These findings emphasize the importance of continuous model evaluation, error analysis, and domain-specific adaptations in building robust sign language recognition systems.

## SAMPLE PREDICTIONS

To evaluate the efficacy of the model, we will conduct a test using three input images. This approach allows us to examine how well the model can distinguish between visually distinct hand signs and accurately assign them to their respective categories.

### Sample 1: Image from Test Set: Letter L


Upload a hand gesture image

Drag and drop file here  
Limit 200MB per file • PNG, JPG, JPEG

Browse files

L\_test.jpg 11.4KB

×



Uploaded Image

Prediction

Predicted Letter: L

Confidence Score  
100.00%

Top 5 Predictions

L: 100.00%

D: 0.00%

K: 0.00%

J: 0.00%

T: 0.00%

The model correctly identified the sign as the letter "L." This demonstrates that the model is performing well on unseen data drawn from the same distribution as the training data, indicating effective generalization within the dataset.

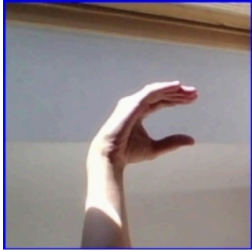
## Sample 2: Image from Train Set: Letter C

Upload a hand gesture image

Drag and drop file here  
Limit 200MB per file • PNG, JPG, JPEG

Browse files

C93.jpg 12.2KB



Uploaded Image

### Prediction

Predicted Letter: C

Confidence Score  
100.00%

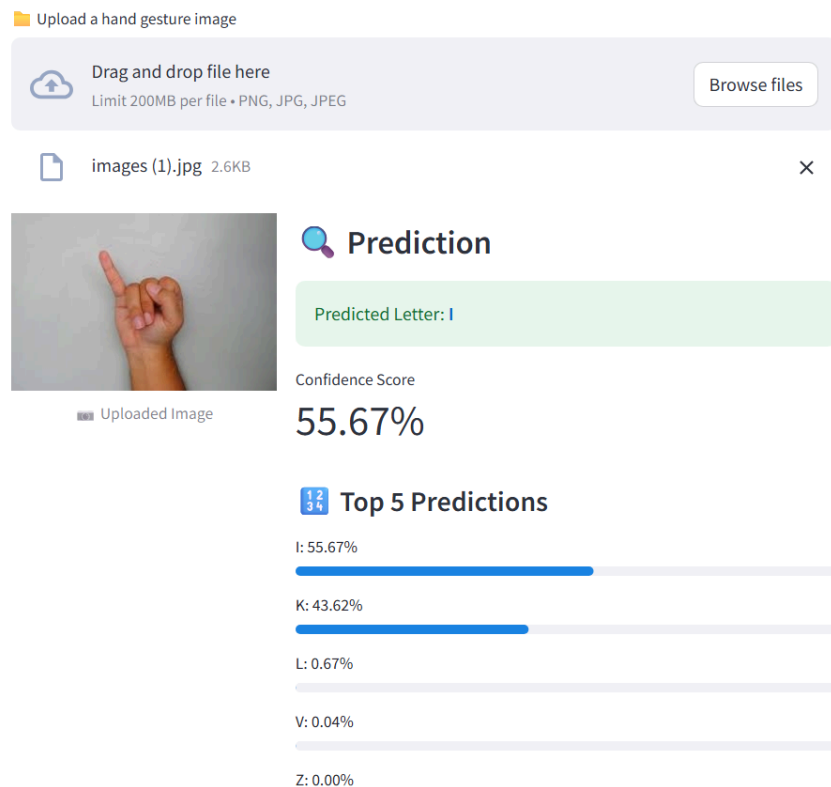
### Top 5 Predictions

C: 100.00%	
O: 0.00%	
G: 0.00%	
P: 0.00%	
D: 0.00%	

The model also correctly classified this image as the letter "C." This suggests that the model has successfully learned from the training data and can accurately recognize patterns it has encountered during training. However, consistent success on training data may also indicate potential overfitting, which should be monitored in further evaluation.

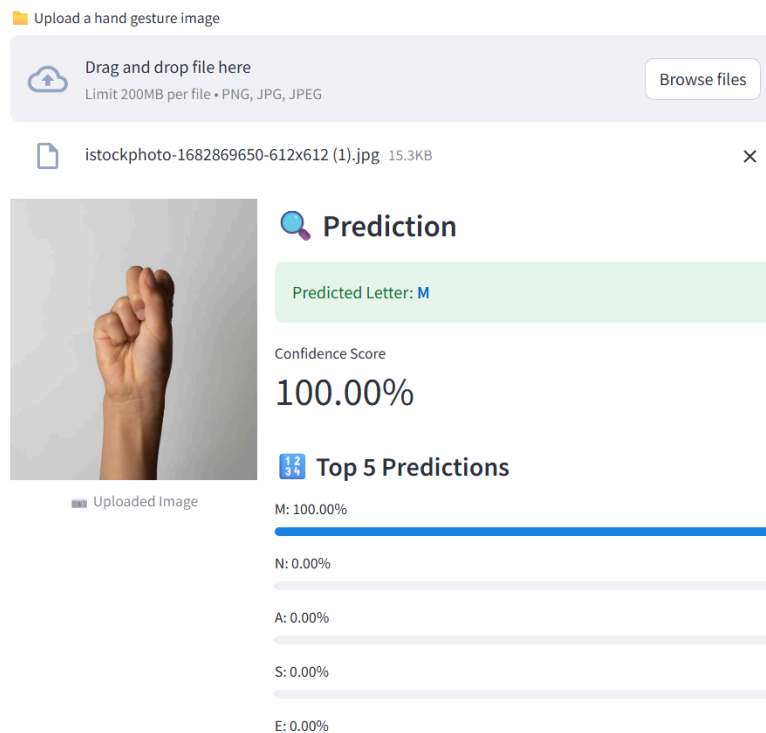


### Sample 3: Internet Image: Letter I



The model correctly classified the letter “I” with a confidence score of 55.67%, demonstrating its ability to accurately recognize signs even when presented with new images. This result highlights the model’s considerable generalization capabilities within the domain and its robustness to variations such as lighting, background, and hand shapes. It suggests that the current training data and augmentation strategies are effective in helping the model handle diverse inputs, though further improvements can still enhance performance on more varied real-world data.

#### Sample 4: Internet Image: Letter S



The model misclassified this input, predicting a different letter, “M”, when the input was actually “S.” This result highlights a key limitation: the model struggles to generalize to out-of-distribution data, especially images with different lighting, backgrounds, or hand variations. This underscores the need for domain adaptation strategies, such as training with webcam-captured data or expanding the diversity of the dataset through augmentation or transfer learning.

Based on the model evaluation on the held-out test set, the current model achieved an accuracy of **93%**. This indicates strong overall performance and suggests that the model has learned meaningful features to distinguish between different signs

effectively. However, while 93% accuracy is promising, it also means there is a 7% error rate where the model misclassifies signs. Depending on the application, especially in sensitive communication contexts like ASL recognition, even small error rates can impact usability. In summary, there is definitely room for further improvements in data quality, augmentation, class balancing, and model architecture to reduce errors and improve robustness in real-world settings.

## LIMITATIONS AND IMPROVEMENTS

One of the primary challenges faced by the execution of this project is the model's **lack of generalizability** to real-time webcam inputs under real-world conditions. While the model works exceptionally well on high-quality, curated dataset images, it considerably struggles when applied to more variable, real-world scenarios. This limitation is mostly caused by the domain mismatch between the dataset used for training, which often consists of high-resolution, uniformly-lit, and well-posed images, and the inherently noisier, lower-quality webcam captures. These issues underscore the importance of addressing the domain shift to ensure robust model performance beyond controlled environments.

A second major challenge is the model's **tendency to misclassify** certain hand signs that exhibit strong visual similarities. For instance, signs 'M' and 'N' have very similar hand configurations, and it is hard to tell them apart, particularly when they are imaged from slightly different angles or in low-light conditions. This problem highlights the inherent difficulty of recognizing subtle intra-class variations in ASL gestures. The model's performance on such cases suggests that while it can learn broad gesture patterns, it requires additional refinement to handle fine-grained distinctions effectively.

To address these limitations and make the model more useful in practice, various improvements can be suggested for future versions of this project. First, **developing a custom ASL dataset captured from webcams** is essential to bridge the domain gap.

This dataset should reflect diverse real-world conditions, such as varying lighting, backgrounds, skin tones, and hand shapes, ensuring the model can handle the variability it will encounter during deployment.

Second, **enhancing the data augmentation pipeline** with advanced techniques, such as brightness and contrast shifts, Gaussian noise, perspective distortions, and motion blur, will improve the model's resilience to common visual artifacts in webcam feeds.

Third, **leveraging transfer learning** by fine-tuning pretrained models like MobileNetV2, EfficientNet, or lightweight vision transformers (ViTs) can significantly improve performance. These architectures are well-suited for real-time inference on low-resource devices, and initializing with pretrained weights (e.g., from ImageNet) allows for faster convergence and stronger feature representations.

**Lastly**, advancing from static sign recognition to dynamic sign language recognition is a critical next step. Many ASL signs involve motion and temporal context, which static classifiers cannot capture. Incorporating temporal deep learning architectures, such as Long Short-Term Memory (LSTM) networks, Gated Recurrent Units (GRUs), or 3D Convolutional Neural Networks (3D CNNs), would enable the system to model temporal dynamics and recognize continuous sign language sequences more effectively.

All in all, while the current model is a promising foundation, addressing domain discrepancies through webcam-specific datasets, expanding data augmentation, employing transfer learning, and integrating temporal modeling are key to building a robust and practical ASL recognition system. These improvements will not only enhance accuracy and generalization but also contribute to the development of more inclusive and accessible communication technologies for Deaf and DHH communities.

## REFERENCES

- [1] B. Alsharif, E. Alalwany, A. Ibrahim, I. Mahgoub, and M. Ilyas, "Real-Time American Sign Language Interpretation Using Deep Learning and Keypoint Tracking," *National Library of Medicine*, 2025. <https://pubmed.ncbi.nlm.nih.gov/40218651/>
- [2] Y. Zhang and X. Jiang, "Recent Advances on Deep Learning for Sign Language Recognition," *Tech Science Press*, 2024. <https://www.techscience.com/CMES/v139n3/55626>
- [3] F. El-Qoraychy and Y. Mualla, "American Sign Language Recognition Using Convolutional Neural Networks," *The Sixteenth International Conference on Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services*, 2023. [https://personales.upv.es/thinkmind/dl/conferences/centric/centric\\_2023/centric\\_2023\\_2\\_60\\_38007.pdf](https://personales.upv.es/thinkmind/dl/conferences/centric/centric_2023/centric_2023_2_60_38007.pdf)
- [4] A. Karapbasi, A. Elbushra, O. Al-Hardanee, and A. Yilmaz, "DeepASLR: A CNN based Human Computer Interface for American Sign Language Recognition for Hearing-Impaired Individuals," *Computer Methods and Programs in Biomedicine-Update*, 2, Article 100048, 2022. <https://doi.org/10.1016/j.cmpbup.2021.100048>
- [5] A. Nagaraj, "ASL Alphabet," *Kaggle*, 2018. <https://www.kaggle.com/dsv/29550>