SRM
UNIVERSITY *AP*
Andhra Pradesh

SIGN LANGUAGE INTERPRETER

Project Report

Lakshaya K- AP21110010265

Charishma K- AP21110010268

Charishma A- AP21110010290

Sravani K - AP21110010293

Deekshitha CH- AP21110010304

TABLE OF CONTENTS

ABSTRACT

This project focuses on developing an American Sign Language (ASL) Interpreter using a combination of Digital Image Processing (DIP) techniques and a deep learning-based MobileNetV2 model. The goal is to bridge the communication gap between individuals with hearing impairments and the rest of the world by translating ASL gestures into corresponding alphabets. The dataset includes static images of ASL gestures, preprocessed using DIP techniques such as grayscale conversion, Gaussian blur, adaptive thresholding, edge detection, and morphological operations. These techniques highlight the structural features of gestures, which are fed into the MobileNetV2 model for classification. The model achieved high accuracy, demonstrating its effectiveness in recognizing ASL gestures.

INTRODUCTION

American Sign Language (ASL) is widely used by the hearing-impaired community to communicate, but its adoption among non-users is limited. The lack of interpreters creates a significant communication barrier. This project aims to develop an automated system to recognize ASL gestures and convert them into readable text.

Using a combination of Digital Image Processing (DIP) techniques and a deep learning model (MobileNetV2), the system processes images of ASL gestures to extract key features. The project addresses challenges such as noise, lighting variations, and the similarity between certain gestures. By focusing on static ASL gestures, the model serves as a foundation for future systems capable of real-time gesture recognition.

LITERATURE REVIEW

Several approaches have been explored for ASL recognition. Traditional methods relied on handcrafted feature extraction techniques such as HOG (Histogram of Oriented Gradients) and SIFT (Scale-Invariant Feature Transform), which struggled with variations in lighting and background noise.

Deep learning models, such as Convolutional Neural Networks (CNNs), have shown great promise, especially when combined with transfer learning. However, these approaches often rely on large datasets and computational resources.

This project stands out by integrating Digital Image Processing techniques with a lightweight and efficient model (MobileNetV2). This combination enables accurate recognition with relatively smaller datasets, making the system both robust and computationally feasible.

METHODOLOGY

DATASET

The dataset used contains static images of ASL gestures organized into train, validation, and test sets. Each class represents an alphabet (excluding 'j' and 'z'), with images in RGB format.

PREPROCESSING (DIP TECHNIQUES)

- **Library-Based Pipeline:** Used highly optimized OpenCV functions for grayscale conversion, Gaussian blur, adaptive thresholding, Sobel edge detection, and morphological operations.
- **Manual Implementation Pipeline:** Each technique was implemented from scratch, ensuring full control over the operations. The pipeline included:
    - ***Grayscale Conversion:*** Simplified image data while retaining intensity.
    - ***Gaussian Blur:*** Reduced noise using a manually computed Gaussian kernel.
    - ***Adaptive Thresholding:*** Converted images to binary using a local mean-based thresholding technique.
    - ***Sobel Edge Detection:*** Detected edges using horizontal and vertical gradient kernels.

- ***Dilation and Erosion:*** Refined edges to emphasize gesture features and remove noise. Both pipelines were tested with the same dataset to evaluate their impact on model performance.

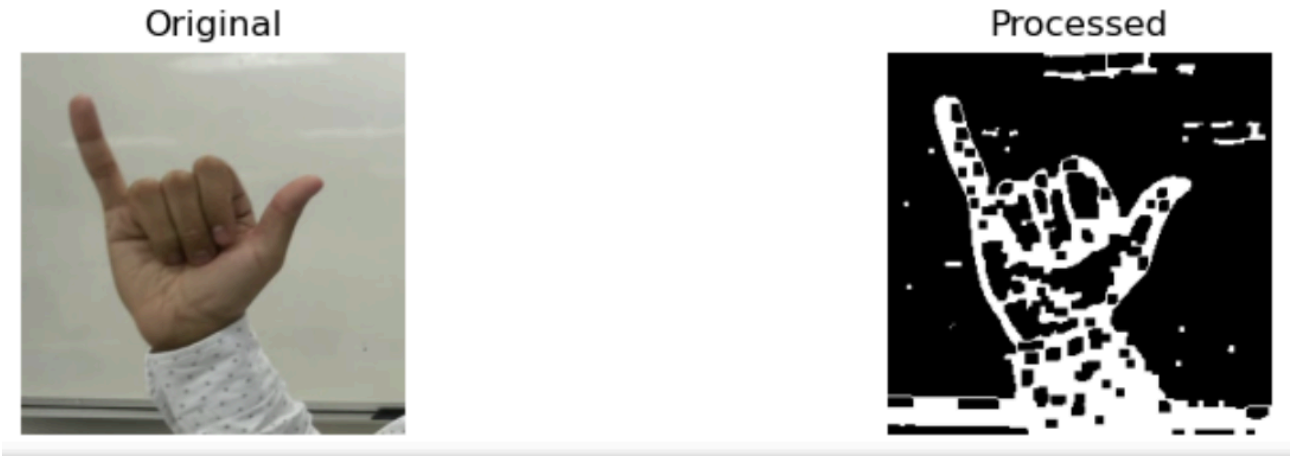

Figure X: Example of the preprocessing pipeline. The original image (left) is converted to a processed image (right) using techniques such as grayscale conversion, Gaussian blur, adaptive thresholding, and edge detection.

MODEL ARCHITECTURE

The MobileNetV2 model was used for classification:

- **Base Model:** Pre-trained on ImageNet and fine-tuned for ASL gestures.
- **Custom Layers:** Added Dense, Dropout, and output layers for classification.

The preprocessing pipeline ensures the model focuses on gesture features while ignoring irrelevant background information.

EXPLORATION OF ALTERNATIVE APPROACHES

Before finalizing the current preprocessing pipeline and MobileNetV2 architecture, various approaches were tested to evaluate their effectiveness in recognizing ASL gestures. The findings are summarized below:

1. **Classic CNN:**
    a. A traditional Convolutional Neural Network (CNN) was implemented from scratch.
    b. ***Accuracy:*** 65%
    c. Observation: While the CNN performed better than random guessing, it lacked the feature extraction power of modern architectures and struggled with classifying gestures with similar shapes.

2. **ResNet**
    a. A pre-trained ResNet architecture was fine-tuned for the ASL dataset.
    b. ***Accuracy:*** 7%
    c. Observation: ResNet drastically underperformed, likely due to the small dataset size, making it difficult to train the deeper layers of the model effectively.
3. **MobileNetV2**
    a. A lightweight, pre-trained MobileNetV2 was tested.
    b. ***Accuracy:*** 72%
    c. Observation: MobileNetV2 provided the best results among the tested architectures due to its efficient feature extraction capabilities, making it the ideal choice for the base model.
4. **MobileNetV2 with Mediapipe Hands**
    a. The MediaPipe Hands library was integrated to detect and segment hand regions before passing them to the model.
    b. ***Accuracy:*** 78%
    c. Observation: While the model performed well, MediaPipe struggled in cases where hands overlapped with complex backgrounds or when gestures were partially visible.
5. **MobileNetV2 with Rotation Augmentation:**
    a. Data augmentation using random rotations was applied to make the model more robust to variations in hand orientation.
    b. ***Accuracy:*** Similar to MediaPipe Hands (78%).
    c. Observation: The model benefited from the augmentation but failed to surpass MediaPipe in accuracy.
6. **MobileNetV2 with Segmentation using Otsu's Thresholding:**
    a. Otsu's thresholding was used for image segmentation, attempting to separate the hand region from the background.
    b. ***Accuracy:*** 62%
    c. Observation: The technique failed to accurately capture inner finger positions, leading to loss of critical gesture features and reduced accuracy.

IMPLEMENTATION

The project was implemented using Python, leveraging libraries such as TensorFlow/Keras for the deep learning model and OpenCV for Digital Image Processing. The preprocessing pipeline processes each image before feeding it into the model. The training process involved data augmentation to improve model generalization, using techniques like rotation, zoom, and flipping. The MobileNetV2 model was initialized with pre-trained weights, and the final layers were trained for 50 epochs. Early stopping and model checkpoints were used to optimize training and prevent overfitting. The final trained model, final v1.keras, achieved significant accuracy on the test set.

RESULT AND EVALUATION

The project evaluated the performance of two approaches: one using optimized library-based DIP techniques and the other using manually implemented DIP techniques. The library-based approach achieved a test accuracy of 82%, while the manual implementation achieved 79%. The difference of 3% highlights the trade-offs between the two methods.

COMPARISON OF APPROACHES

The dataset used contains static images of ASL gestures organized into train, validation, and test sets. Each class represents an alphabet (excluding ‘j’ and ‘z’), with images in RGB format.

1. **Library-Based DIP Techniques:**
    a. These are highly optimized, computationally efficient, and handle edge cases better.
    b. Achieved slightly higher accuracy due to precise implementations of operations like adaptive thresholding and Sobel filtering.

2. **Manual Implementation of DIP Techniques:**
    a. Provided flexibility and allowed for deeper understanding and customization.
    b. Although slightly less accurate, it demonstrated effective preprocessing, with only a minor drop in performance.

The results show that both approaches are viable, with the library-based techniques being better suited for deployment and real-world applications, while manual implementations are excellent for customization and educational purposes.

CHALLENGES AND IMPROVEMENTS

One of the main challenges faced during the project was achieving the same level of accuracy with manually implemented DIP techniques as with library-based functions. The manually implemented functions lacked the fine optimizations found in libraries like OpenCV. This gap resulted in a 3% lower accuracy compared to the library-based approach.

POTENTIAL IMPROVEMENTS

1. **Optimization of Manual Functions:**
    a. Enhance operations such as Gaussian blur and Sobel filtering with more efficient convolution techniques.
    b. Experiment with kernel size and parameters to better adapt to edge cases.
2. **Hybrid Approach:**
    a. Combine manual implementations for certain tasks (e.g., adaptive thresholding) with library-based functions for others.
    b. This could balance precision with flexibility.
3. **Post-Processing Enhancements:**
    a. Apply additional normalization or contrast enhancement to further refine the processed images before feeding them to the model.

To address these issues, more advanced DIP techniques could be employed, such as background subtraction or contrast enhancement. Additionally, increasing the size and diversity of the dataset could improve model performance. Fine-tuning the deeper layers of the MobileNetV2 model may also enhance feature extraction for complex gestures.

CONCLUSION

This project successfully implemented and compared two approaches for preprocessing images in an American Sign Language Interpreter system. Using library-based DIP techniques resulted in a slightly higher accuracy (82%) compared to manually implemented methods (79%). The small difference demonstrates the effectiveness of the manual techniques despite their lack of computational optimization.

KEY LEARNINGS

- ***Library-Based Methods:*** Better for real-world applications where precision and efficiency are critical.
- ***Manual Implementations:*** Ideal for learning and experimenting with customized preprocessing pipelines.

FUTURE WORK

Future enhancements to this project include:

1. Incorporating dynamic gestures for word- and sentence-level recognition.
2. Extending the system to recognize gestures in video sequences.
3. Optimizing the model for real-time performance on mobile devices.
4. Leveraging 3D image data to capture finer details of gestures.

These improvements would significantly expand the applicability and effectiveness of the ASL Interpreter.

REFERENCES

- TensorFlow Documentation: https://www.tensorflow.org/
- OpenCV Documentation: https://docs.opencv.org/
- MobileNetV2 Paper: “Inverted Residuals and Linear Bottlenecks” by Sandler et al.
- Dataset Source: [Mention your dataset source]