



MCT-241: Embedded Systems II

**Real-Time Sign Language Detection using
Convolutional Neural Network (CNN) and Media Pipe
for Hand Landmark Detection**

Submitted By:

Muhammad Haider Ali

Registration No:

2022-MC-45

**Department of Mechatronics & Control Engineering, University of
Engineering & Technology, Lahore**

December 25, 2024

Overview:

This project focuses on implementing a **Real-Time Sign Language Detection System**. The goal is to translate sign language gestures into corresponding alphabets using a combination of **Convolutional Neural Networks (CNNs)** and **MediaPipe** for hand landmark detection. The system captures live video input, detects and processes hand gestures, and classifies them into corresponding sign language alphabets in real time.

Project Implementation:

1. System Components

The project is organized into the following components:

CNNModel.py: Contains the CNN architecture used for classifying hand gestures.

handLandMarks.py: Implements the MediaPipe hand landmark detection logic and processes the detected landmarks for input to the CNN.

mediapipeHandDetection.py: Integrates MediaPipe for real-time hand tracking from webcam input.

realTime.py: Combines hand detection and gesture classification for real-time operation.

training.py: Trains the CNN model on a dataset of labeled hand gestures.

testCNN.py: Evaluates the accuracy of the trained CNN model on a test dataset.

CNN_model_alphabet_SIBI.pth: Pre-trained model weights for gesture classification.

Steps to Run the Project:

Install Required Dependencies: Ensure Python 3.x is installed. Install project dependencies with:

```
pip install -r requirements.txt
```

Alternatively, install packages individually:

```
pip install opencv-python mediapipe torch numpy pandas
```

Run the Real-Time Detection Activate the real-time system by executing:

```
python realTime.py
```

The script will start the webcam feed and display real-time hand gesture classifications.

Train the CNN Model (Optional) Train the CNN model from scratch using a custom dataset by running:

```
python training.py
```

Test the CNN Model (Optional) Evaluate the trained model's performance with:

```
python testCNN.py
```

How the System Works

1. Hand Landmark Detection

- MediaPipe captures hand landmarks from the live webcam feed.
- The detected landmarks are normalized and processed into a format suitable for the CNN model.

2. Feature Extraction

- Key features from hand landmarks are extracted, focusing on joint positions and relative distances.

3. Gesture Classification

- The processed features are fed into the CNN model.
- The model outputs the predicted class, corresponding to a sign language alphabet (A-Z).

4. Real-Time Feedback

- The classified gesture is displayed in real-time on the screen, enabling immediate user feedback.

System Requirements

Software

- Python 3.x
- OpenCV
- MediaPipe
- PyTorch
- Pandas

Hardware

- A computer with a webcam for capturing hand gestures.
- GPU support is recommended for faster model training and testing.

Contributions and Future Scope

Contributions

Developers and researchers are encouraged to contribute by:

- Adding support for additional gestures or full words.
- Improving the CNN model's architecture for higher accuracy.
- Optimizing performance for low-latency detection.

Future Enhancements

- Integration with speech synthesis for spoken output of classified gestures.
- Support for dynamic gestures and full sentences.
- Mobile and cross-platform compatibility.

Contact Information

For inquiries, suggestions, or contributions, please contact:

- Email: [vegeta.khan2000@gmail.com]
- GitHub: [<https://github.com/haider-sama>]

This project represents a significant step towards bridging communication gaps and promoting accessibility for individuals relying on sign language.