MADRAS INSTITUTE OF TECHNOLOGY
ANNA UNIVERSITY, CHENNAI-600 044

DEPARTMENT OF INFORMATION TECHNOLOGY

IT 5039 - DEEP LEARNING

PROJECT REPORT

REAL-TIME SIGN LANGUAGE TRANSLATION

Submitted by,

Meenakshi C (2022506097)

Nethra V S (2022506115)

ABSTRACT:

Sign language is a vital communication tool for the hearing-impaired community. This project implements a real-time sign language translation system that recognizes hand gestures corresponding to numbers, words, emotions, and actions, and translates them into readable text.

The system uses Media Pipe to extract accurate hand landmarks and a hybrid deep learning model combining CNN (for spatial feature extraction) and LSTM (for temporal sequence learning) to classify gestures. The system supports dynamic gestures, providing real-time translation with high accuracy, making it suitable for assistive technology, educational tools, and interactive applications.

INTRODUCTION:

- Sign language is essential for communication among hearing-impaired individuals.
- Automatic translation allows instant conversion of gestures to text, making communication accessible to everyone.
- Previous systems using simple models or template matching are limited to static gestures.
- CNN can extract spatial patterns from hand images or landmark grids.
- LSTM can learn temporal sequences from consecutive frames, enabling dynamic gesture recognition.
- Media Pipe provides hand landmarks in 3D (x, y, z) as input features for the CNN-LSTM model.

OBJECTIVES:

1. **Collect Sequential Gesture Data:**

- Capture hand gestures representing sign language symbols using a webcam.
- Extract 21 hand landmarks per frame with MediaPipe.
- Store gestures as sequences of consecutive frames to capture dynamic motion.

2. **Preprocess Landmarks for Temporal Learning:**

- Normalize landmark coordinates for consistency across users and hand sizes.
- Format the data into sequence arrays suitable for temporal analysis by an LSTM network.
- Encode gesture labels using one-hot encoding for multi-class classification.

3. **Train CNN-LSTM Hybrid Model:**

- Use CNN layers to extract spatial features from hand landmarks at each frame.
- Use LSTM layers to learn temporal patterns across sequences of frames.

- Train the model to classify gestures accurately and save it for real-time inference.

4. Implement Real-Time Translation:

- Capture live webcam feed and detect hand landmarks in real-time.
- Process sliding window sequences of frames to predict gestures dynamically.
- Display the recognized gestures as text output on screen for immediate translation.

5. Support a Wide Range of Gestures:

- Recognize numbers (0–9), common words, emotions, and daily actions.
- Ensure robust performance across different users, lighting conditions, and hand orientations.

TOOLS AND TECHNOLOGIES USED:

1. Programming Language

- **Python:** Chosen for its extensive libraries for computer vision, deep learning, and real-time data processing. Python provides easy integration with frameworks like TensorFlow and OpenCV, making it ideal for gesture recognition systems.

2. Libraries

1. OpenCV (Open Source Computer Vision Library):

- Used for webcam capture and video stream processing.
- Provides GUI functionalities to display real-time gesture recognition output.
- Handles frame manipulation, flipping, and drawing overlays for landmarks.

2. Media Pipe:

- Developed by Google for real-time hand and body tracking.
- Detects hand landmarks per frame (coordinates) for each hand.
- Provides reliable hand detection even under moderate variations in lighting and hand orientation.

3.TensorFlow/Keras:

- Deep learning framework used to build the CNN-LSTM hybrid model.
- CNN layers extract spatial features from hand landmarks.
- LSTM layers capture temporal patterns for dynamic gesture recognition.
- Provides tools for model training, evaluation, and saving (gesture_cnn_lstm.h5).

4.NumPy:

- Used for efficient numerical operations and data handling.
- Handles sequences of landmarks, normalization, and array manipulation.

3. Development Environment / IDE

1. VS Code

- Provides an integrated environment for writing Python code, debugging, and managing packages.
- Facilitates real-time testing and visualization of gesture recognition.

MODELS USED:

1. Convolutional Neural Network (CNN)

- Extracts spatial features from hand landmarks in each frame.
- Learns hand shape, finger positions, and relative distances between landmarks.
- Processes each frame individually using Time Distributed CNN layers before feeding into LSTM.
- Helps distinguish similar gestures based on spatial patterns.

2. Long Short-Term Memory Network (LSTM)

- Captures temporal patterns in sequences of frames.
- Learns motion dynamics, such as waving or sliding gestures.
- Preserves temporal dependencies across consecutive frames.
- Outputs a feature representation summarizing the motion of the gesture.

3. CNN-LSTM Hybrid Model

- Combines spatial learning (CNN) and temporal learning (LSTM).
- Works effectively for both static and dynamic gestures.
- Steps in the model:

    1. **Input:** Sequence of frames, each containing hand landmarks and coordinates.
    2. **CNN layers:** Extract spatial features for each frame.
    3. **Pooling layers:** Reduce dimensionality of spatial features.
    4. **LSTM layers:** Learn temporal dependencies across the sequence of frames.

5.Dense + Softmax layer: Classify the gesture among all possible classes (numbers, words, emotions, actions).

- Trained model is saved as gesture_cnn_lstm.h5.

4. Advantages of CNN-LSTM

- Handles dynamic gestures that change over time.
- Improves accuracy compared to models that only use spatial (CNN) or temporal (LSTM) features.
- Lightweight enough to perform real-time prediction using hand landmarks instead of full images.

SYSTEM ARCHITECTURE:

The system is designed to translate sign language gestures into text in real time. It is divided into three main modules: **Data Collection, Model Training, and Real-Time Translation.**

1.Data Collection:

Purpose: Build a comprehensive dataset of dynamic sign language gestures.

Key Steps:

1. Open a webcam feed using OpenCV.
2. Detect hands using MediaPipe and extract 21 3D landmarks (x, y, z) per frame.
3. Capture sequences of frames (e.g., 20 consecutive frames) for each gesture to preserve motion information.
4. Label each sequence with the corresponding gesture name.
5. Save the dataset in .npy or similar formats for efficient access during model training.

Outcome: A structured dataset of temporal gesture sequences for CNN-LSTM input.

2.Model Training

Purpose: Train a CNN-LSTM hybrid model to classify gestures accurately.

Key Steps:

1. **Load Data:** Import the sequential gesture dataset.
2. **Preprocess:**
    - Normalize landmark coordinates for uniform scale.
    - Format sequences into arrays suitable for CNN-LSTM input.

- Encode gesture labels using one-hot encoding.

3. **CNN Layers:** Extract spatial features from landmarks in each frame (hand shape and position).
4. **LSTM Layers:** Learn temporal dependencies across frames (dynamic motion of gestures).
5. **Dense + Softmax Layer:** Output gesture class probabilities for multiclass classification.
6. Compile and train the model using Adam optimizer and categorical cross entropy loss.
7. Save the trained model as gesture_cnn_lstm.h5.

Outcome: A robust CNN-LSTM model capable of recognizing both static and dynamic gestures.

3. Real-Time Translation

Purpose: Translate sign language gestures into readable text live.

Key Steps:

1. Capture live webcam feed.
2. Detect hand landmarks in each frame using MediaPipe.
3. Collect sliding window sequences of consecutive frames for temporal
4. prediction.
5. Input sequences into the CNN-LSTM model to predict gestures.
6. Display the recognized gesture as text on-screen, updating continuously.
7. Provide real-time translation even for dynamic gestures like waving or moving hands.

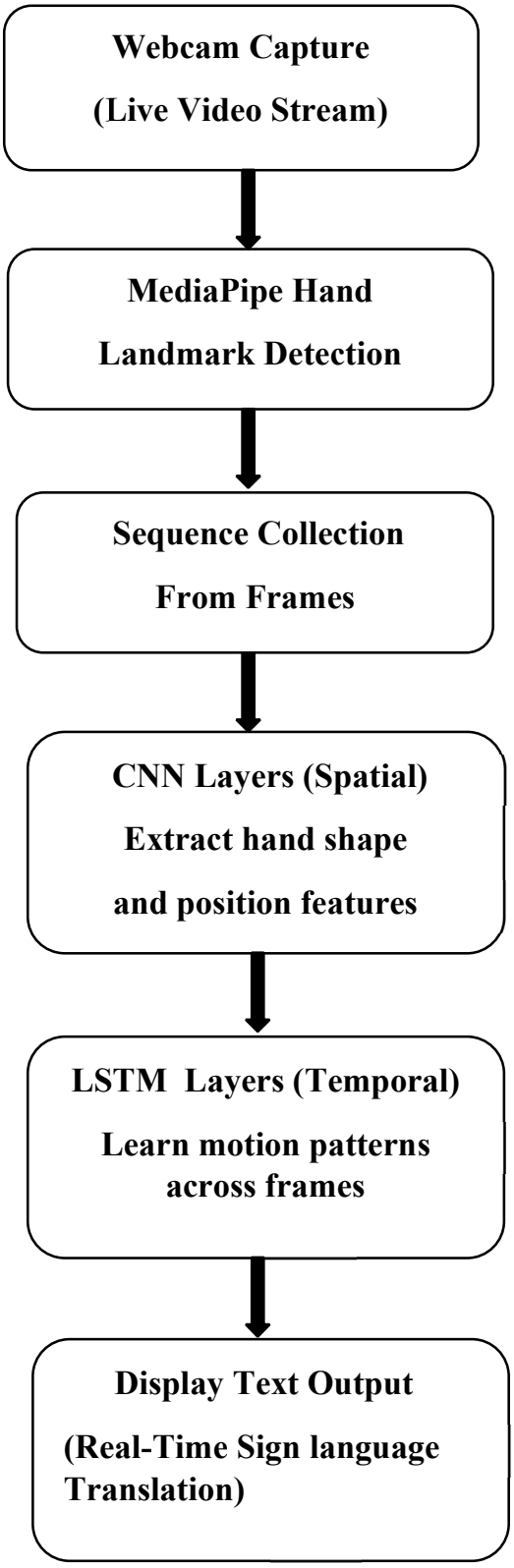
Outcome: Users can communicate via sign language with instant textual translation visible on the screen.

ARCHITECTURE DIAGRAM EXPLANATION:

Explanation:

1. Webcam feed provides live input.
2. MediaPipe detects hands and extracts landmarks.
3. Landmarks are collected into sequences to capture motion.
4. CNN processes each frame for spatial patterns.
5. LSTM processes sequences for temporal patterns.
6. Softmax layer outputs predicted gesture class.
7. Text output is displayed live on screen.

ARCHITECTURE DIAGRAM:



APPLICATIONS:

1.Communication Aid for the Hearing and Speech Impaired

- Helps bridge the communication gap between sign language users and people who don't understand sign language.
- Converts gestures into readable text in real time, enabling smoother daily interactions.

2.Educational Tools and Learning Platforms

- Can be integrated into sign language learning apps for beginners.
- Provides instant feedback on gesture accuracy, improving learning efficiency.

3.Customer Service and Public Assistance

- Useful in banks, hospitals, airports, and government offices to assist differently-abled individuals.
- Allows staff to understand sign-based queries without a translator.

4.Smart Devices and IoT Integration

- Can be extended to gesture-controlled systems for home automation.
- Enables control of devices like lights, fans, and media through sign-based commands.

5.Virtual and Augmented Reality Applications

- Enhances interaction in VR/AR environments through gesture-based inputs.
- Can be adapted for virtual meetings or immersive sign-based communication.

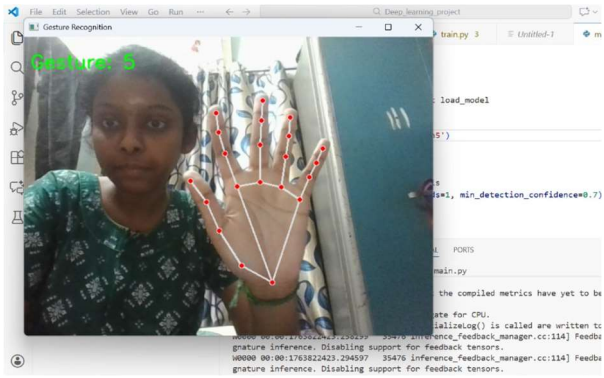
6.Human–Computer Interaction (HCI)

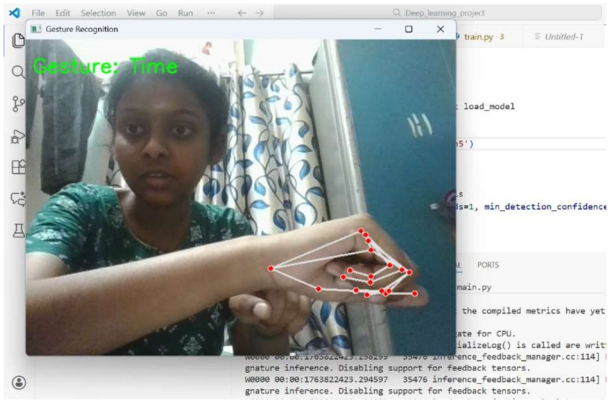
- Improves accessibility by allowing computers to understand hand gestures as input commands.
- Can be integrated into software interfaces, games, or assistive systems.

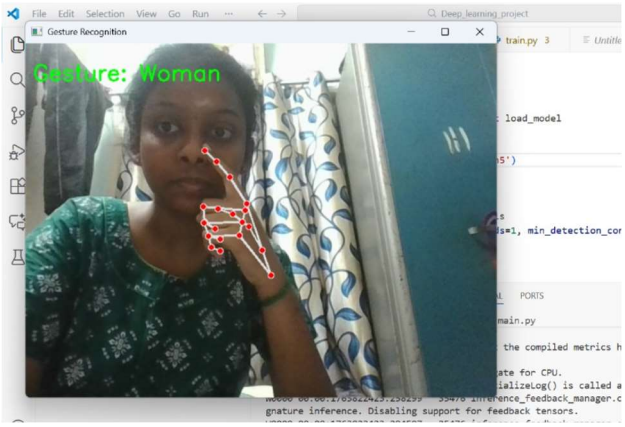
7.Translation and Accessibility Services

- Acts as a real-time interpreter during conferences, classrooms, or online meetings.
- Promotes inclusivity and equal access to information for all users.

PROJECT DEMONSTRATION OUTPUTS:







CONCLUSION:

The real-time sign language translation system effectively recognizes both static and dynamic hand gestures. Using Media Pipe and a hybrid CNN–LSTM model, it accurately converts gestures into readable text. The system improves communication accessibility for hearing-impaired users. Its real-time performance makes it suitable for assistive and interactive applications. With further enhancements, it can become a more comprehensive communication tool.