Gesture Recognition model for KSL

Introduction

This document outlines the design approach for a Gesture Recognition machine learning model aimed at recognizing Kenyan Sign Language (KSL) gestures in real-time. The model leverages **MediaPipe** for hand landmark detection and LSTM for classifying gestures based on hand and finger movements captured through video input.

1.0 Problem Statement

The goal is to create a reliable gesture recognition machine learning model for KSL, capable of identifying intricate hand movements and finger positions. This model will need to accurately translate these gestures into meaningful phrases or words in KSL, overcoming variability in sign execution, hand sizes, and environmental conditions.

2.0 Requirements

2.1 Functional Requirements:

- Real-time video capture using a webcam.
- Hand gesture extraction using MediaPipe to detect keypoints.
- LSTM-based model to classify the gestures.
- Gesture recognition output in text.
- User feedback in the form of on-screen text.

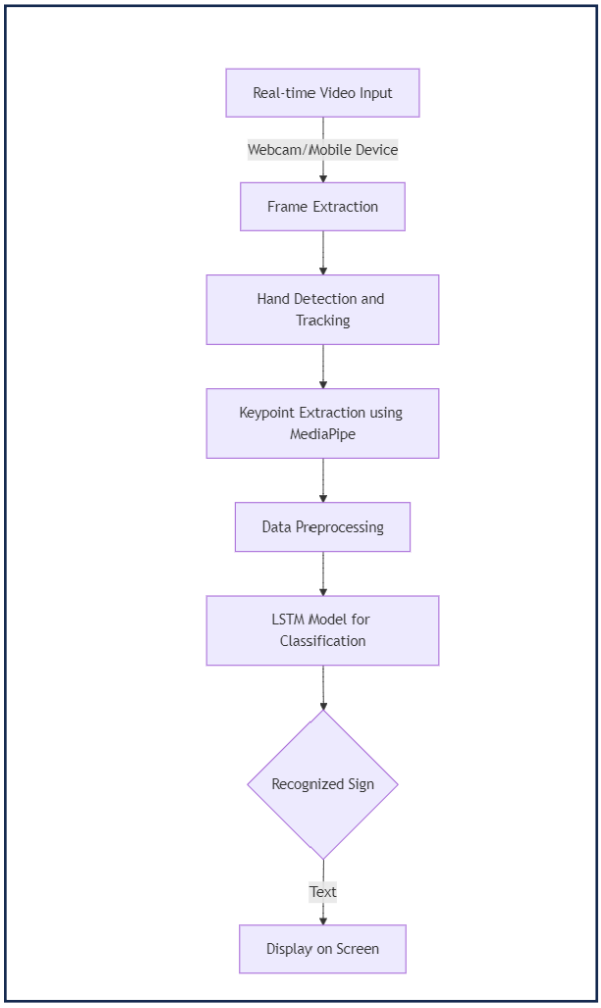
2.2 Non-Functional Requirements:

- Real-time performance.
- Accuracy for gesture classification.
- Low-latency response for real-time interaction.
- Scalability.

3.0 System Architecture

3.1 Overview:

- Input: Real-time video stream from a webcam or mobile device.
- Processing: Keypoint extraction using MediaPipe, preprocessing data, and passing it to the LSTM model for classification.
- Output: Display the recognized sign in text on-screen or as speech.



3.2 Technology Stack:

- **Programming Language**: Python.
- **Frameworks**: TensorFlow (for LSTM), OpenCV, MediaPipe

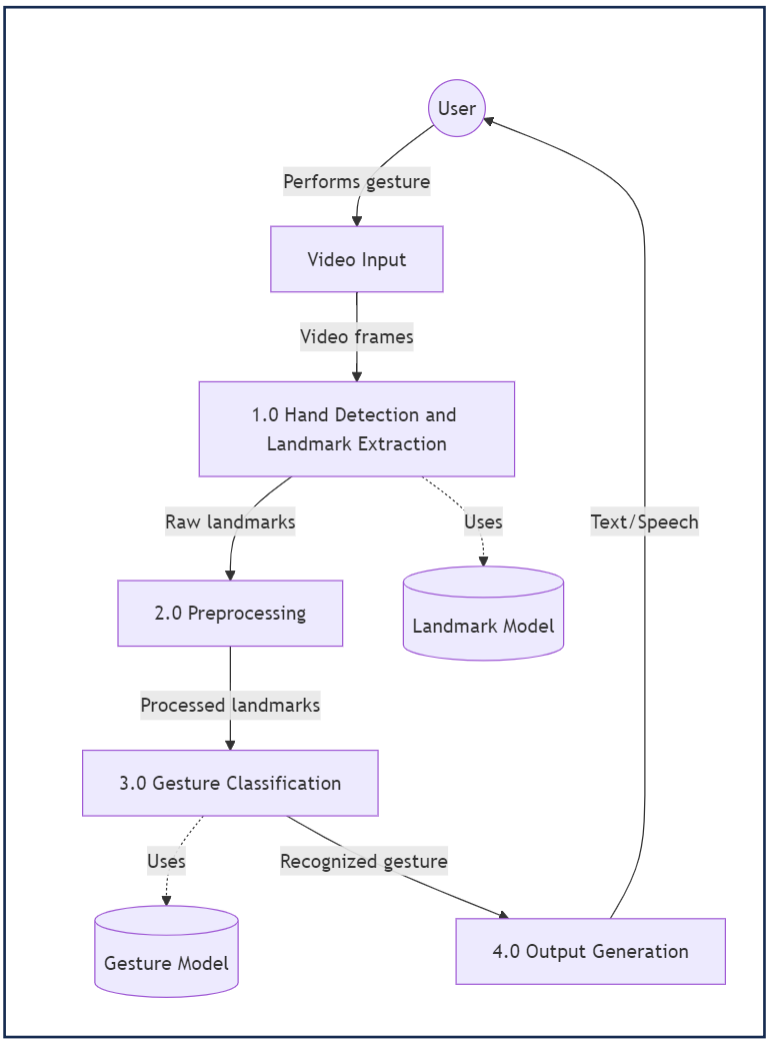
4.0 Logical Breakdown

4.1 Components

i. **MediaPipe Hand Detection:** Extracts 21 keypoints (coordinates of hand landmarks).
ii. **Data Preprocessing Module:** Normalizes, scales, and reshapes the keypoint data.
iii. **LSTM Model:** The core model responsible for gesture classification, trained on labeled KSL data.
iv. **Gesture Classification Engine:** Classifies gestures and returns predictions with a confidence score.
v. **Output Module:** Displays recognized gestures as text.

4.2 Data Models:

- **Gesture Data:** Keypoints of each hand gesture (21 points for each hand).
- **Model Weights:** Trained LSTM weights for gesture classification



5.0 Technology Stack

- **Programming Languages:** Python.
- **Frameworks**: TensorFlow, MediaPipe, OpenCV.

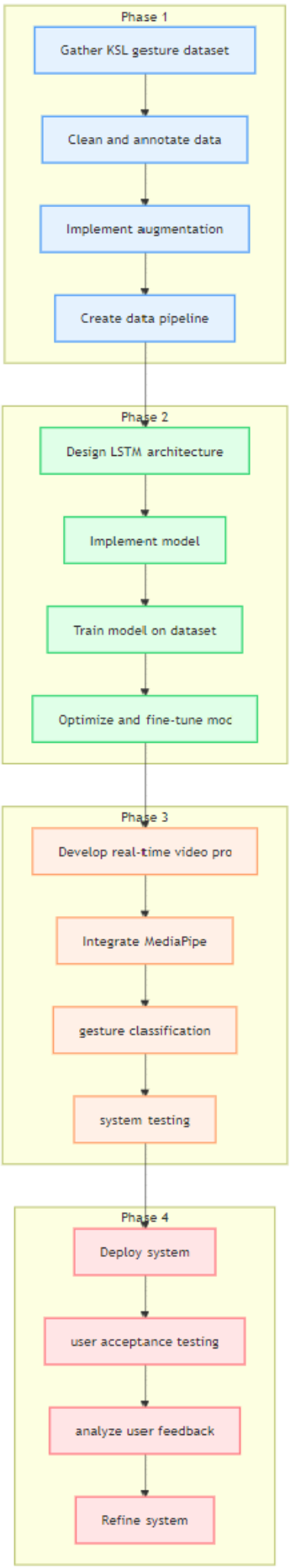
6.0 Implementation Roadmap

- **Phase 1**: Data Collection and Preprocessing.
- **Phase 2**: Model Development and Training.

- **Phase 3**: System Integration and Testing.
- **Phase 4**: Deployment and User Testing.

7.0 Next steps

- Gather more data
- Build on the model to implementation


Phase 1
Gather KSL gesture dataset
Clean and annotate data
Implement augmentation
Create data pipeline
Phase 2
Design LSTM architecture
Implement model
Train model on dataset
Optimize and fine-tune moc
Phase 3
Develop real-time video pro
Integrate MediaPipe
gesture classification
system testing
Phase 4
Deploy system
user acceptance testing
analyze user feedback
Refine system