Yoga Pose Detection System Report

By

Peethala V Siva Sampath Likhit,

MTech CSE,

2024028407,

vpeethal@student.gitam.edu

1. Introduction

This report provides an overview of the **Yoga Pose Detection System**, its components, and the flow of data through the system. The goal of the system is to detect and classify yoga poses from user-uploaded images using machine learning (ML) techniques. The core of the system utilizes **Mediapipe** for pose detection and a **Deep Neural Network (DNN)** model trained on keypoints extracted from the images.

2. System Architecture

The system architecture consists of multiple components, each responsible for a specific task. The architecture follows a step-by-step pipeline, where each step processes the image to ultimately predict the yoga pose.

2.1 High-Level Flow of the System

The system can be broken down into five main components:

1. **User Interface (UI)**
2. **Pose Detection**
3. **Model (DNN) for Classification**
4. **Display Prediction**
5. **Backend Model Storage**

Each of these components has specific functionalities, as outlined below.

2.2 Detailed System Architecture

2.2.1 Client Side (Streamlit Web App UI)

- **Streamlit Web App (UI)**: The user interacts with the system through a simple web interface built using **Streamlit**. The main components of the UI include:

    - **Image Upload Section**: Users can upload images of themselves performing yoga poses.
    - **Prediction Display**: Once an image is uploaded, the system will predict the pose and display the results to the user.
    - **Pose Detection (Mediapipe)**: The uploaded image is passed through Mediapipe for detecting body landmarks.
- **Action**: This interface provides options for the user to upload an image and view the prediction result.

2.2.2 Pose Detection (Mediapipe)

- **Pose Detection**: This component uses **Mediapipe**, a framework for building pipelines to process video and images. Mediapipe is used here to detect the body landmarks in the image.
- **Detect Body Landmarks**: Mediapipe processes the image and returns the keypoints (landmarks) of the human body (such as wrists, elbows, shoulders, etc.). These landmarks are critical for determining the pose.

2.2.3 Model (DNN)

- **Deep Neural Network (DNN)**: The extracted keypoints are fed into a **Deep Neural Network (DNN)** for classification. The model has been trained using keypoints extracted from various yoga poses.
    - **Classify Yoga Pose**: The DNN classifies the yoga pose based on the input keypoints.
    - **Output Prediction**: The DNN outputs the predicted yoga pose label (e.g., "downdog", "plank", etc.).

2.2.4 Display Prediction

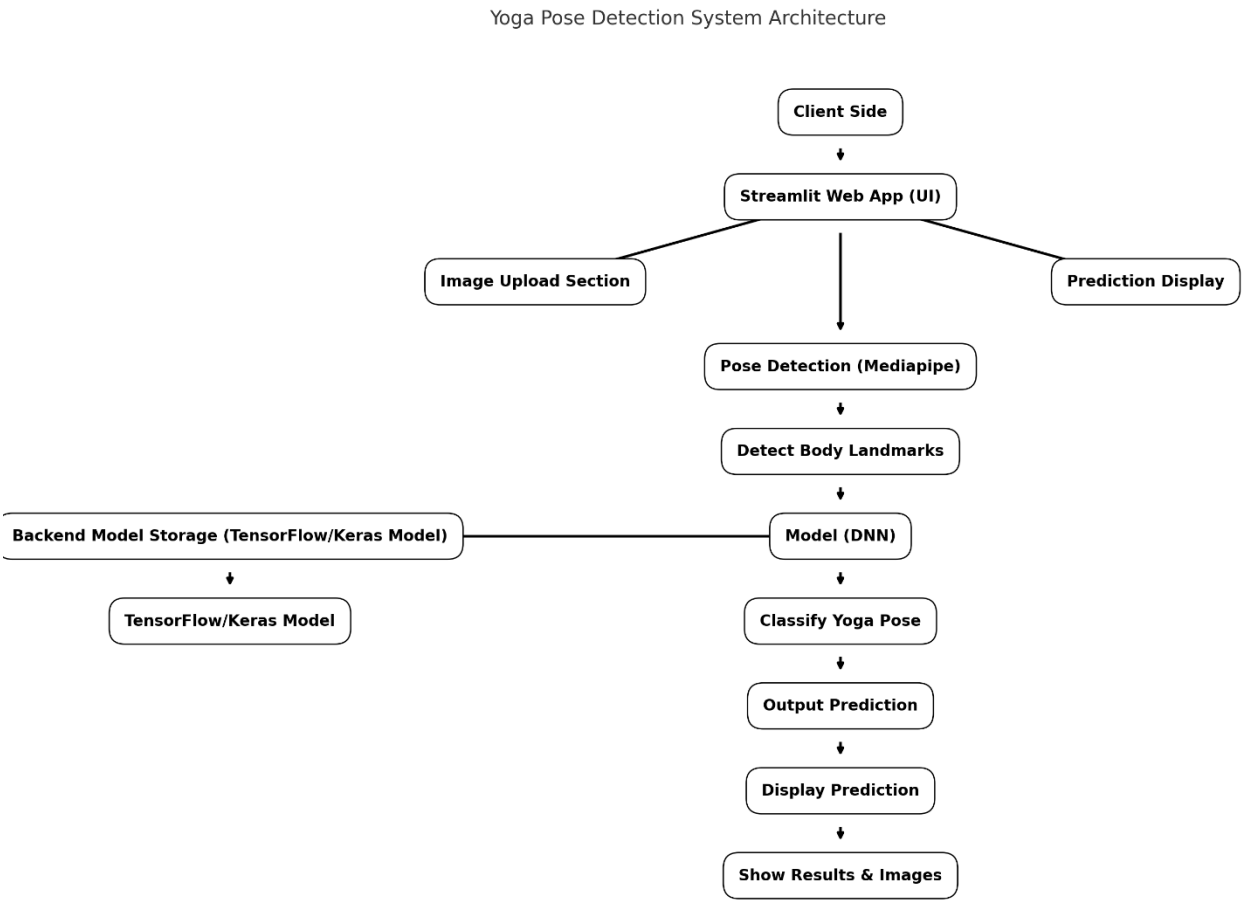
- **Show Results & Images**: After classification, the system will display the predicted yoga pose and relevant reference images for the selected pose.

2.2.5 Backend Model Storage

- **TensorFlow/Keras Model**: The model is stored in the backend as a **TensorFlow/Keras model**, which was trained on a dataset of labeled yoga poses. This model is used to make predictions for uploaded images.

2.3 System Architecture Diagram

The system architecture is visualized as a tree diagram. Each component of the system is represented as a node, with arrows denoting the flow of data between them.



3. Workflow

3.1 Image Upload

The user uploads an image of a yoga pose through the **Streamlit Web App**. The uploaded image is passed to the next component for processing.

3.2 Pose Detection

Mediapipe processes the uploaded image and detects keypoints of the human body. These keypoints are the coordinates (x, y, z) of various body landmarks, which serve as the features for the model.

3.3 Model Prediction

The extracted keypoints are input into the **Deep Neural Network (DNN)**. The DNN is responsible for classifying the pose by recognizing patterns in the body landmarks. The output is a prediction of the yoga pose.

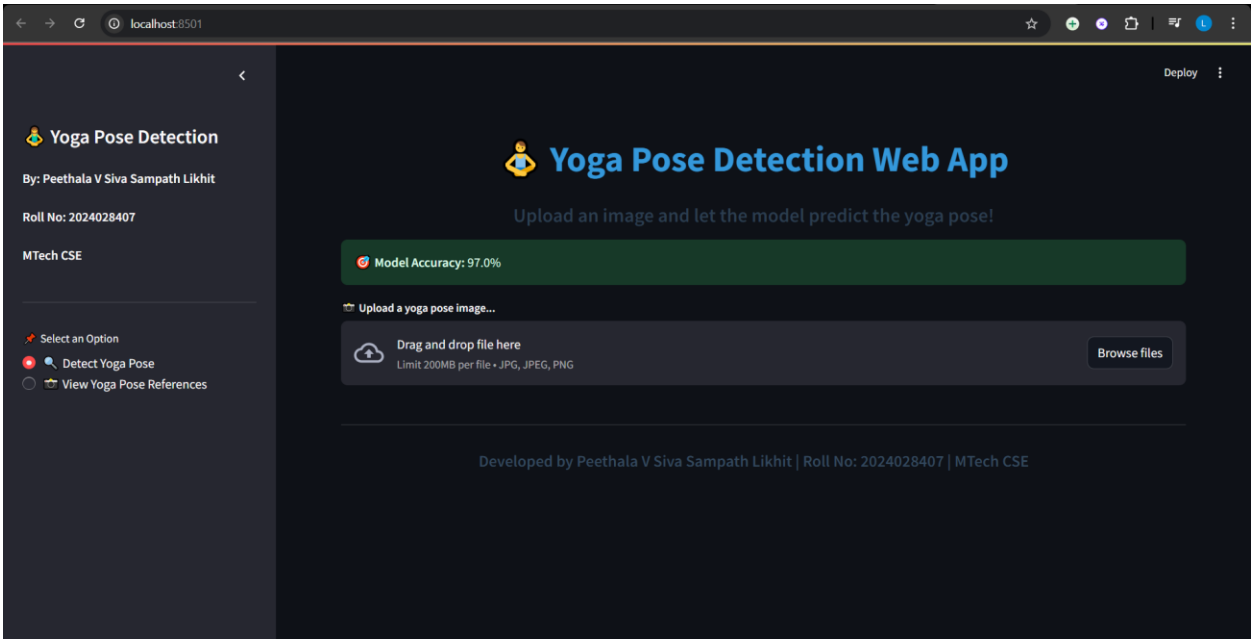
3.4 Display Prediction

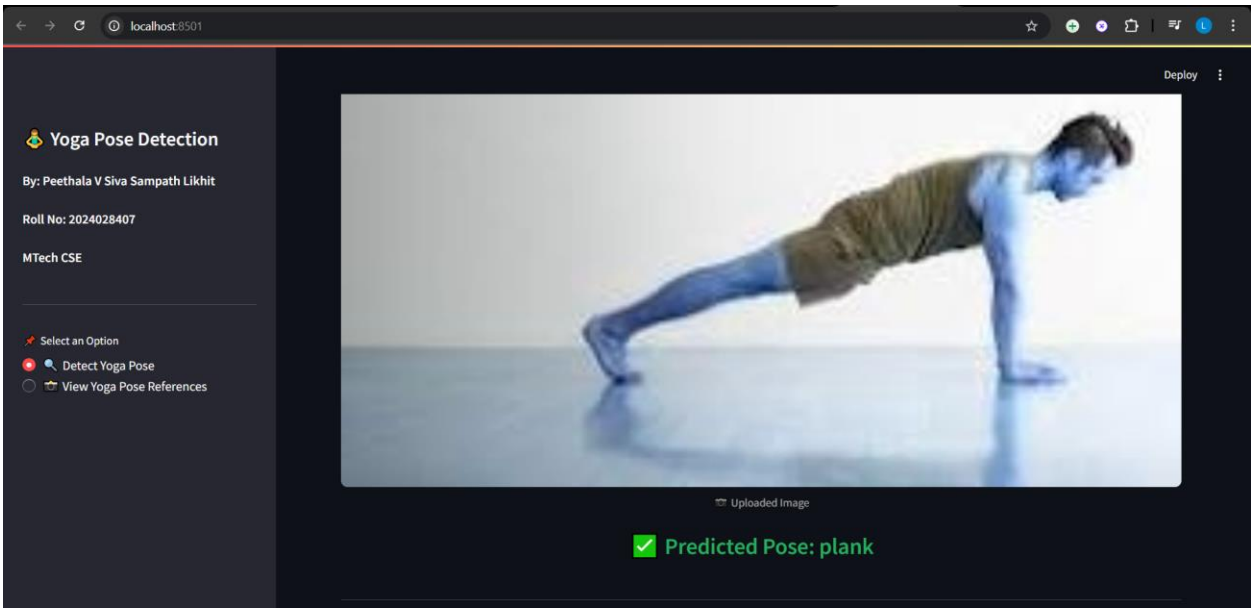
The predicted pose is displayed on the **Streamlit Web App**, along with a message indicating the result. Additionally, reference images of the detected yoga pose are shown to the user.

3.5 Backend Model Storage

The DNN model, stored as a **TensorFlow/Keras model**, is used in the backend to process inputs and generate predictions.

4. Results





5. Conclusion

The Yoga Pose Detection system is an efficient and user-friendly tool designed to classify various yoga poses from images. By utilizing **Mediapipe** for pose detection and a **Deep Neural Network (DNN)** for classification, the system provides a robust and scalable solution for detecting and displaying yoga poses.

With the integration of **Streamlit** for the web interface and **TensorFlow/Keras** for backend processing, this system can be deployed easily for real-time yoga pose detection. Future improvements can focus on increasing the dataset, improving model accuracy, and adding more poses for classification.