IMPLEMENTATION OF YOGA POSE ESTIMATION AND FEEDBACK MECHANISM USING POSE DETECTION FOR SELF LEARNING

An Interim Project Report

Submitted by

DHANUSH VARMA CH [CB.EN.U4CSE18112]
VENKATARAMAN R [CB.EN.U4CSE18162]
HARSHAVARDHAN D [CB.EN.U4CSE18174]
GUHAN K [CB.EN.U4CSE18511]

Under the guidance of
Dr. Raghesh Krishnan K
(Assistant Professor, Department of Computer Science & Engineering)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

AMRITA VISHWA VIDYAPEETHAM

श्रद्धावान् लभते ज्ञानम्

Amrita Nagar PO, Coimbatore - 641 112, Tamilnadu

Review 1 – September 2020

ABSTRACT

Yoga which was developed in ancient India was initially considered as an old age exercise. But because of its many spiritual, physical and mental benefits, Yoga became popular across all age categories. One of the major problems that Yoga and other exercises face is the correct posture in which they must be performed. Even a slight error in the posture may not only nullify the benefits of the exercise but also may result in injuries or lead to structural deformities. So, this puts forward the requirement of an instructor who guides on how the exercise must be done in order to get the maximum from the exercise. But not everyone has the accessibility to an instructor who can supervise and correct the posture whenever the person decides to exercise. Thus, this demands the requirement of a model that will take an asana as input from the user and give an output if the person is doing it correctly or a feedback in case a posture correction is required. This project focuses on the idea of solving the problem of improper posture and allows people to learn and practice exercises correctly by themselves.

Keywords: Yoga, MediaPipe, Pose Assessment, Body Angle, Self-learning, Evaluation

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

CNN	Convolutional Neural Network
CSV	Comma separated values
KNN	K-Nearest Neighbors
LSTM	Long Short Term Memory
ML	Machine Learning
SVM	Support Vector Machine

Chapter 1

INTRODUCTION

Human posture assessment is a difficult issue in the control of Personal Computer vision. It manages confinement of human joints in a picture or video to shape a skeletal portrayal. Practice and wellness is one use of posture assessment that has attracted the attention of many experts in this subject. Yoga, a deeply entrenched practise that originated in India but is now widely acclaimed for its multiple profound physical and mental benefits.

The problem with yoga is that it is critical to do it correctly, as any incorrect position during a yoga session can be useless and potentially inconvenient. This necessitates the presence of an instructor to supervise and correct the individual's stance. Not every client has access to a yoga instructor, a computerised reasoning-based program might be used to recognise yoga poses and provide personalised feedback to help people improve their structure.

1.1 Problem Definition

Our project aims to design a Machine Learning (ML) based application based on the MediaPipe Pose library that monitors the pose performed by the practitioner and gives real time feedback for improvement of the yoga posture.

Generally , Yoga poses are classified by applying ML algorithms with the difference between joints as in Figure 1.1 . The diversity in the structure of Humans across the world and no feedback for improvement to the practitioner are a major drawback of using length calculated between joints . Using angle between adjacent joints as in Figure 1.2 would overcome these drawbacks and can give feedback to improve the pose of the practitioner.

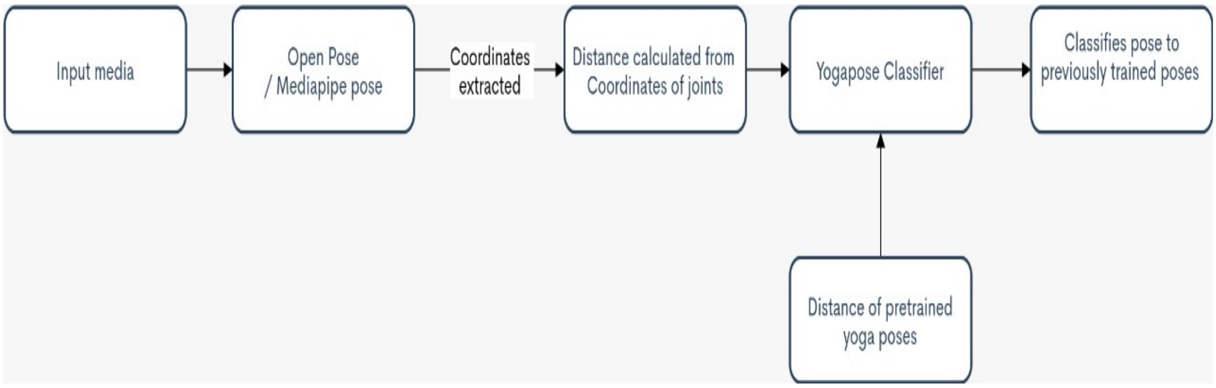

Input media
Open Pose / Mediapipe pose
Coordinates extracted
Distance calculated from Coordinates of joints
Yogapose Classifier
Classifies pose to previously trained poses
Distance of pretrained yoga poses

Figure 1.1: Classification Model based on distance between joints

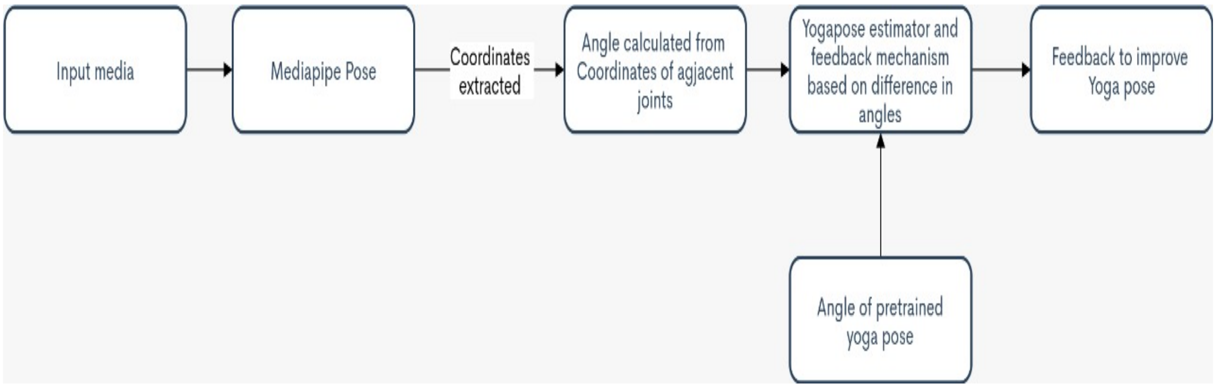

Input media
Mediapipe Pose
Coordinates extracted
Angle calculated from Coordinates of agjacent joints
Yogapose estimator and feedback mechanism based on difference in angles
Feedback to improve Yoga pose
Angle of pretrained yoga pose

Figure 1.2: Feedback Model based on angles between adjacent points

Chapter 2

LITERATURE SURVEY

ML has profound advancements in real time image processing with pose prediction , autonomous systems . Machine Learning Applications which could help humans in various ways are being developed . Various pose estimation based models that predict and classify human poses are discussed below.

2.1 Classification Models

Abhishek Sharma Agrawal et al. (2020) discusses classification of yoga poses using Machine Learning Algorithms like Logistic Regression, Random Forest, Support Vector Machine (SVM), Decision Tree, Naive Bayes and K-Nearest Neighbors (KNN) based on the YOGI dataset using similarities between the joint coordinates extracted by tf-pose-estimation algorithm.

Table 2.1: Benchmark on YOGI Dataset

Classifier	Accuracy
Logistic Regression	0.8215
Random Forest	0.9926
SVM	0.8791
Decision Tree	0.9752
Naive Bayes	0.7475
KNN	0.9725

Muhammad Usama Islam Islam et al. (2017) proposed a model that estimates the accuracy of the pose through data acquisition with the help of a Kinect device (a Kinect device can acquire different types of data in data acquisition steps such as color, depth and skeleton information at the frame rate of 30fps with a resolution of 480*320) then preprocess the data that they acquire and find the acquisitions of the three different weight persons and compare the accuracies of the three persons by three different poses.

Deepak Kumar Kumar and Sinha (2020) classifies the poses using SVM, Convolutional Neural Network (CNN) and CNN+Long Short Term Memory (LSTM) algorithms using the features extracted from the yoga practitioner by OpenPose library based on 12 different joints. CNN+LSTM gave the highest accuracy, compared to the SVM and CNN models. The major drawback with the SVM model was the misclassification between vrikshasana and tadasana since both of them require a standing position and furthermore the underlying posture arrangement is comparative.

2.2 Comparison between 2D and 3D Models

Sankara Narayanan Narayanan et al. (2021) extracts 17 different body points and compares the accuracy of classification of 2D and 3D versions of OpenPose models using the features generated for 2D and 3D Comma separated values (CSV) files and finally states that the 3D model of OpenPose Predicts more accurate form results compared to 2D model.

2.3 Summary

The research gap is these models don't provide valuable feedback on how yoga practitioners practice a pose , they are not trained over a diverse dataset and some use very expensive devices which require high maintainability since they are easily damaged. Classification models don't give valuable information for the learner to improve /practise yoga. Thus a system that assesses yoga pose of a learner by detecting the human body first using the webcam, extracting the coordinated points of various joints in the body, calculating the difference of body angles between the trained pose images and that of the user accurately and finally provide real-time feedback on improvisation of the incorrect parts between learner and the trained images is necessary.

2.4 Data Set

Source of data set :

- https://www.kaggle.com/niharika41298/yoga-poses-dataset
- https://sites.google.com/view/yoga-82/home
- The combined dataset contains- 900 + diverse images (Children , Women and Men) for five different Yoga Asanas.
- Images in different Lightings/orientation for each yoga pose.

2.5 Software/Tools Requirements

- Flask (Python library) - Front End
- OpenCV - Used for feature extraction ,object detection in this case the person and altering the orientation and brightness of images.
- Media Pipe (coordinates extraction) - Media Pipe offers open source cross-platform, customizable ML solutions for live and streaming media.
- Visual Studio Code - code editor redefined and optimized for building and debugging modern web and cloud applications.
- TensorFlow - provides a collection of workflows to develop and train models using Python or JavaScript, and to easily deploy in the cloud, on-prem, in the browser or on-device

Chapter 3

MODULARIZATION & PLAN

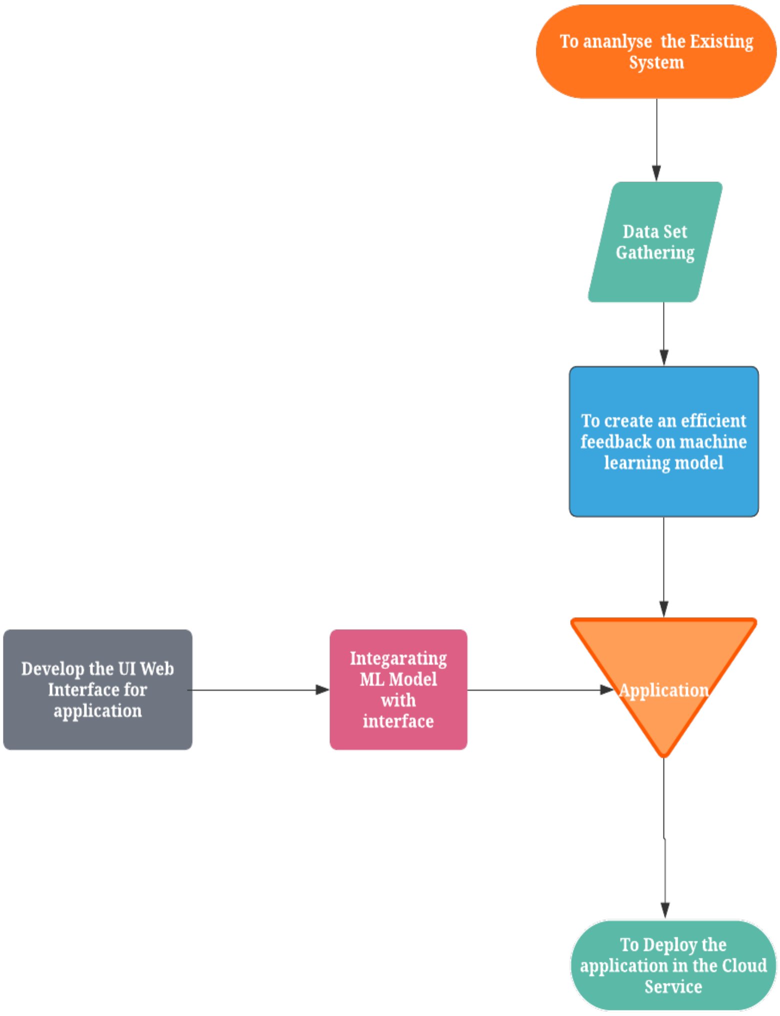

To ananlyse the Existing System
Data Set Gathering
To create an efficient feedback on machine learning model
Develop the UI Web Interface for application
Integarating ML Model with interface
Application
To Deploy the application in the Cloud Service

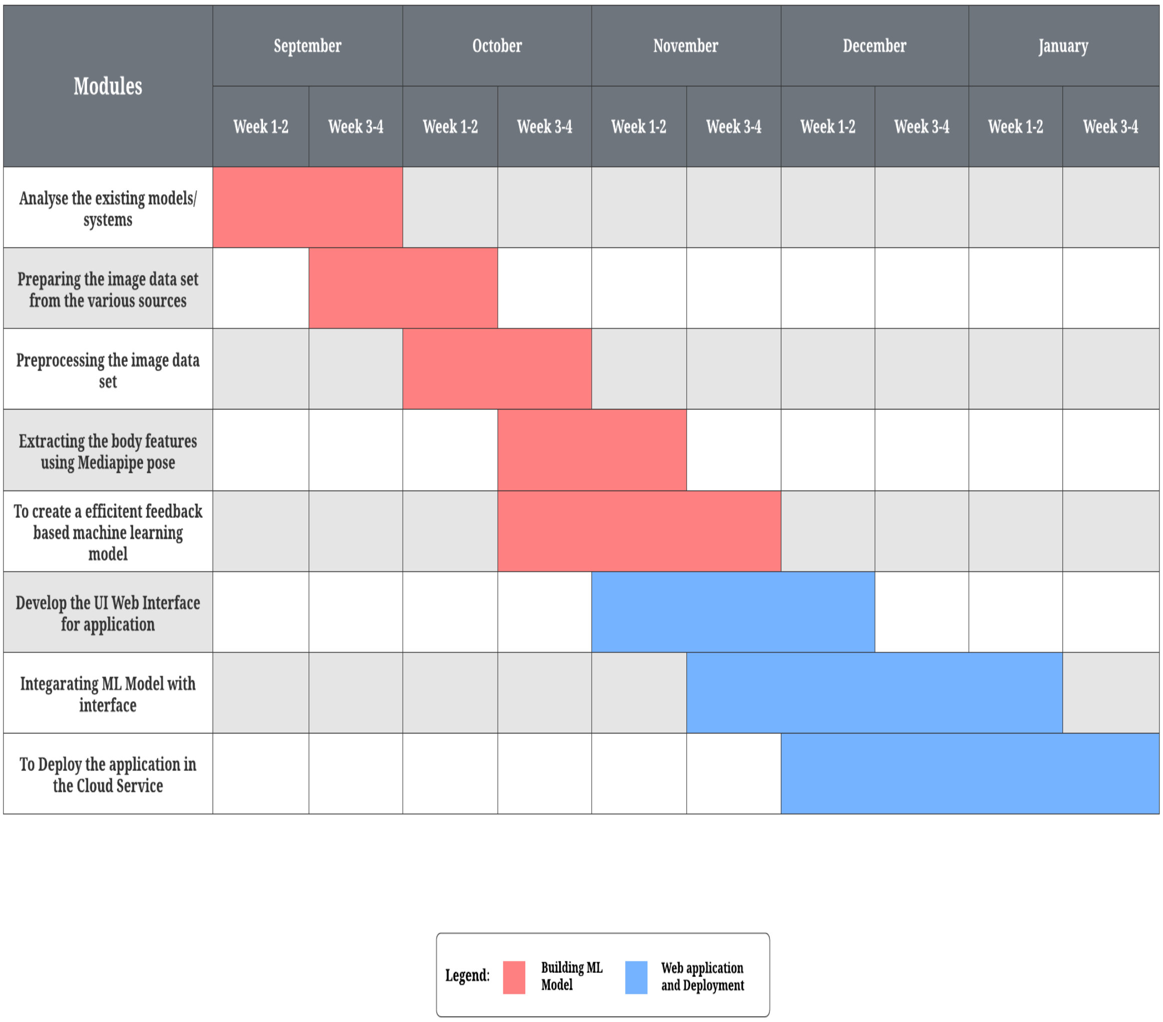
Figure 3.1: Modularization flowchart

3.1 Responsibilities of Team Members

Table 3.1: Modules and Responsibilities

Modules	**Group Member**
Analyze the existing models/ systems	Ch.Dhanush Varma D. Harshavardhan K. Guhan R. Venkataraman
Preparing the image data set from the various data sources	R. Ventakaraman
Preprocessing the image data set	D. Harshavardhan
Extracting the body features using Mediapipe pose	K. Guhan
To create an efficient feedback based machine learning models	R. Venkataraman K. Guhan
Develop the UI Web Interface for application	Ch. Dhanush Varma D. Harsha Vardhan
Integrating ML Model with interface	R. Venkataraman K. Guhan
To Deploy the application in the Cloud Service	D.Harshavardhan

3.2 Month Wise Plan



Modules	September		October		November		December		January	
	Week 1-2	Week 3-4	Week 1-2	Week 3-4	Week 1-2	Week 3-4	Week 1-2	Week 3-4	Week 1-2	Week 3-4
Analyse the existing models/ systems	🟥	🟥								
Preparing the image data set from the various sources		🟥	🟥							
Preprocessing the image data set			🟥	🟥						
Extracting the body features using Mediapipe pose				🟥	🟥					
To create a efficitent feedback based machine learning model				🟥	🟥	🟥				
Develop the UI Web Interface for application					🟦	🟦	🟦			
Integarating ML Model with interface						🟦	🟦	🟦	🟦	
To Deploy the application in the Cloud Service							🟦	🟦	🟦	🟦

Legend: 🟥 Building ML Model 🟦 Web application and Deployment

Figure 3.2: Month wise plan

REFERENCES

1. Agrawal, Y., Shah, Y., and Sharma, A. (2020). “Implementation of machine learning technique for identification of yoga poses.” *2020 IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT)*, IEEE. 40–43. https://ieeexplore.ieee.org/document/9115758.

2. Islam, M. U., Mahmud, H., Ashraf, F. B., Hossain, I., and Hasan, M. K. (2017). “Yoga posture recognition by detecting human joint points in real time using microsoft kinect.” *2017 IEEE Region 10 humanitarian technology conference (R10-HTC)*, IEEE. 668–673. https://ieeexplore.ieee.org/document/8289047.

3. Kumar, D. and Sinha, A. (2020). *Yoga Pose Detection and Classification Using Deep Learning*. LAP LAMBERT Academic Publishing. https://ijsrcseit.com/CSEIT206623.

4. Narayanan, S. S., Misra, D. K., Arora, K., and Rai, H. (2021). “Yoga pose detection using deep learning techniques.” *Available at SSRN 3842656*. https://papers.ssrn.com/sol3/papers.cfm?abstract$_{i}d = 3842656$.