UPES

UNIVERSITY WITH A PURPOSE

Report for Major I

Submitted by

500069734	R134218123	Pratham Pandey
500068183	R134218125	Pulkit Mittal
500067543	R134218152	Shashwat Chitransh
500068730	R134218206	Karmanya Dadhich

Under the mentorship of

Ms. Shahina Anwarul

Department of Systemics

Yoga Pose Detection

INDEX

Abstract

The project aims at designing a system that can detect a yoga performer's pose in real time and predict whether he/she is doing it correctly or not. We plan on accurately predicting five popular yoga poses namely, downward dog, plank pose, tree pose, goddess pose and warrior-2 pose.

This would involve training a ML model to complete the required task. OpenCV would be used for handling the images and pose detection will be performed using MediaPipe.

A web application will be developed providing the user a platform to easily perform his task with ease.

Keywords: *OpenCV, MediaPipe, Pose detection*

Introduction

Yoga is a spiritual science which involves physical, mental and spiritual practices aimed at stilling the mind. It has renowned health benefits and is known to improve one's approach towards life. But to reap maximum benefits, one needs to ensure that it is done properly.

Human pose estimation from video plays a critical role in various applications such as quantifying physical exercises, sign language recognition, and full-body gesture control.

The purpose of this project is to design a system that predicts the yoga pose a person is performing and ensuring that he/she is doing it correctly. It is aimed towards people who’re new to yoga and are aiming to perform the poses with high accuracy.

This might be useful for trainers who can’t individually pay attention to each new trainee, allowing them to carefully monitor performance of multiple trainees without paying close attention to each of them.

Related Works

[1] DeepPose was the first major paper, published in CVPR 2014 that applied Deep Learning to Human pose estimation. It achieved SOTA performance and beat existing models back in the year 2014.

[2] Efficient Object Localization Using Convolutional Networks uses heatmaps by sliding through multiple subsets of the image (windows) in parallel to simultaneously capture features at a variety of scales. A heatmap predicts the probability of one of the essential points.

[3] Convolutional Pose Machines propose the serial usage of stages to predict. A CPM consists of an image feature computation module followed by a prediction module. The multiple stages can be trained end to end.

[4] The HRNet (High-Resolution Network) model maintains a high-resolution representation throughout the whole process, instead of high → low → high-resolution representation, and this works very well. The architecture starts from a high-resolution subnetwork as the first stage, and gradually adds high-to-low resolution subnetworks one by one to form more stages and connect the multi-resolution subnetworks in parallel.

[5] OpenPose offers realtime Multi-Person 2D Pose Estimation using Part Affinity Fields proposes the detection of multiple people in an image. It uses an approach called non-parametric representation, also known as Part Affinity Fields (PAFs).

Problem Statement

Yoga, if done properly has vast health benefits. But if performed incorrectly it can cause a lot of problems like ankle sprain, stiff neck, muscle pulls, etc. Thus, at least in initial stages yoga must be performed under an experienced instructor. This might not always be possible, owing to expensive yoga classes and social distancing restrictions being imposed around the world.

Objectives

The project aims at designing a system that can detect a yoga performer's pose in real time and predict whether he/she is doing it correctly or not. We plan on accurately predicting five popular yoga poses namely, downward dog, plank pose, tree pose, goddess pose and warrior-2 pose.

This would allow anyone to learn yoga and improve his/her lifestyle, without even interacting with a third person allowing them to reap max benefit and ensure proper social distancing at hard times like this.

Sub-objectives

- Comparative analysis of different classifiers for Yoga pose classification.
- Collection and cleaning of dataset of five poses for yoga pose classification.
- Design a classification model for predicting yoga pose accurately
- Evaluation of the proposed model
- Building an interactive web application for the user.

Methodology

The project involved all the steps involved in a typical ML project. They are:

- Loading the dataset
- Exploratory data analysis
- Data pre-processing
- Building the model
- Compiling the model
- Training the model
- Using our model to make predictions

We built a dataset with 5 poses using a webscraper. Once the dataset was cleaned and processed, we loaded it and converted the images into required format. We used MediaPipe to find a number of landmarks in the human body and on the basis of the position of those landmarks we determine the pose being performed and use it to train our model. Once the model is trained we'll use it to predict yoga pose in real time.

We used multiple algorithms like:

- Logistic Regression, a statistical analysis (also known as logit model) is often used for predictive analytics and modeling and extends to applications in ML. In this analytics approach, the dependent variable is finite or categorical: either A or B (binary regression) or a range of finite options A, B, C or D (multinomial regression). It is used in statistical software to understand the relationship between the dependent variable and one or more independent variables by estimating probabilities using a logistic regression equation.
- Random Forest Classifier consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction.

- Gradient Boosting Classifier is a special type of Ensemble Learning technique that works by combining several weak learners (predictors with poor accuracy) into a strong learner (a model with strong accuracy). This works by each model paying attention to its predecessor's mistakes.
- KNN Classifier is a supervised ML algorithm which assumes that similar things exist in close proximity. It considers K Nearest Neighbours (Data points) to predict the class or continuous value for the new Datapoint.
- Ridge Classifier, based on Ridge regression method, converts the label data into [-1, 1] and solves the problem with regression method. The highest value in prediction is accepted as a target class and for multiclass data multi-output regression is applied.

The algorithm giving the highest accuracy was used for creating the final model.

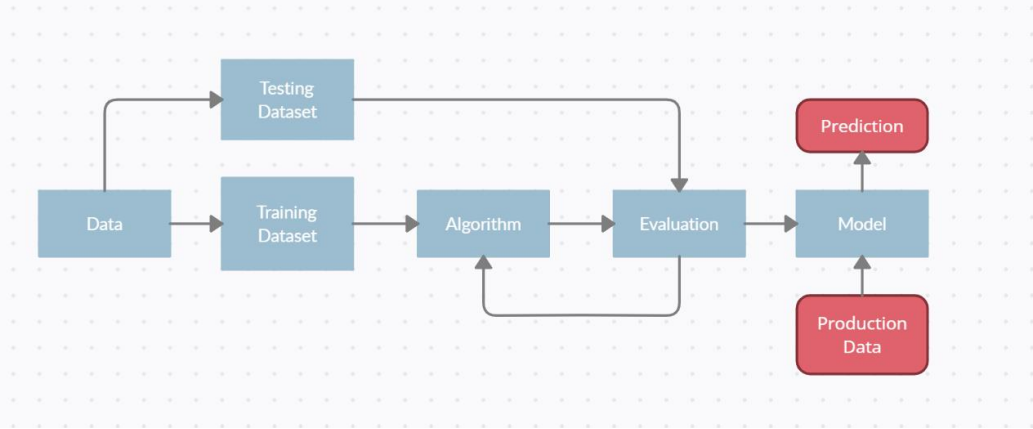


Fig 1. Workflow

Requirements

Hardware: RAM -> 4GB , ROM-> 2GB

Recommended Processor -> intel Core i3

Software: Python, Web Browser

Technical Diagrams

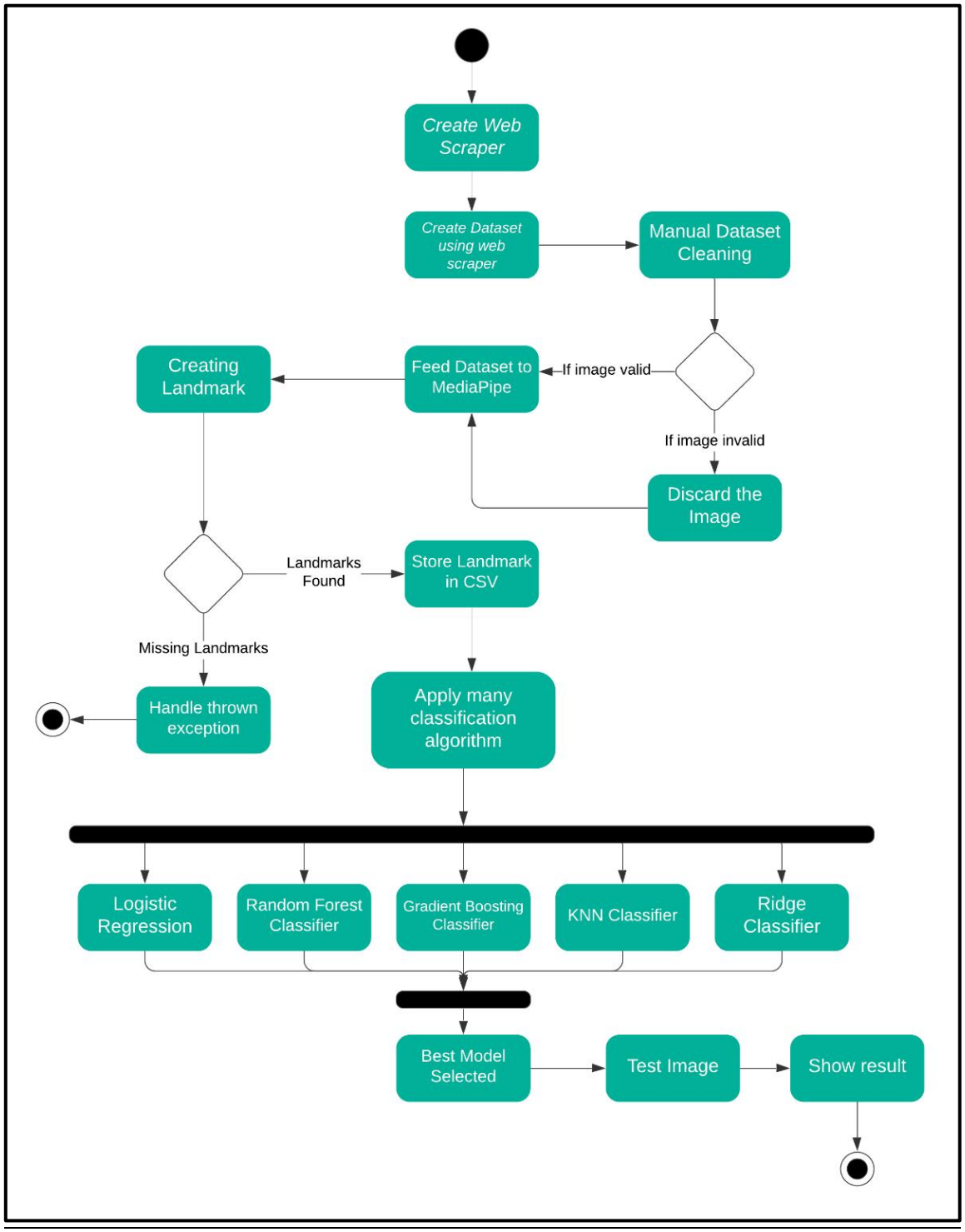


Fig 2. Activity Diagram

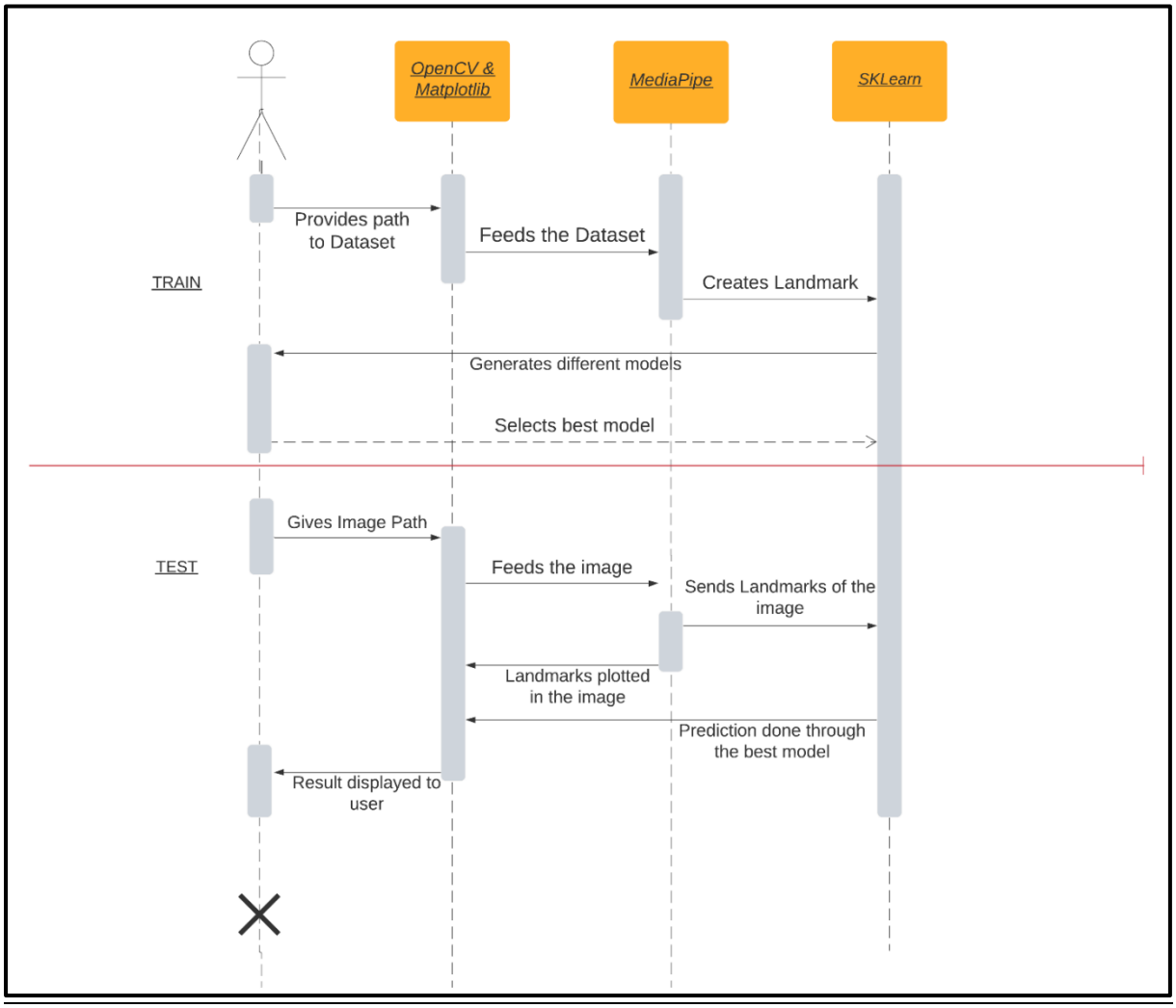


Fig 3. Sequence Diagram

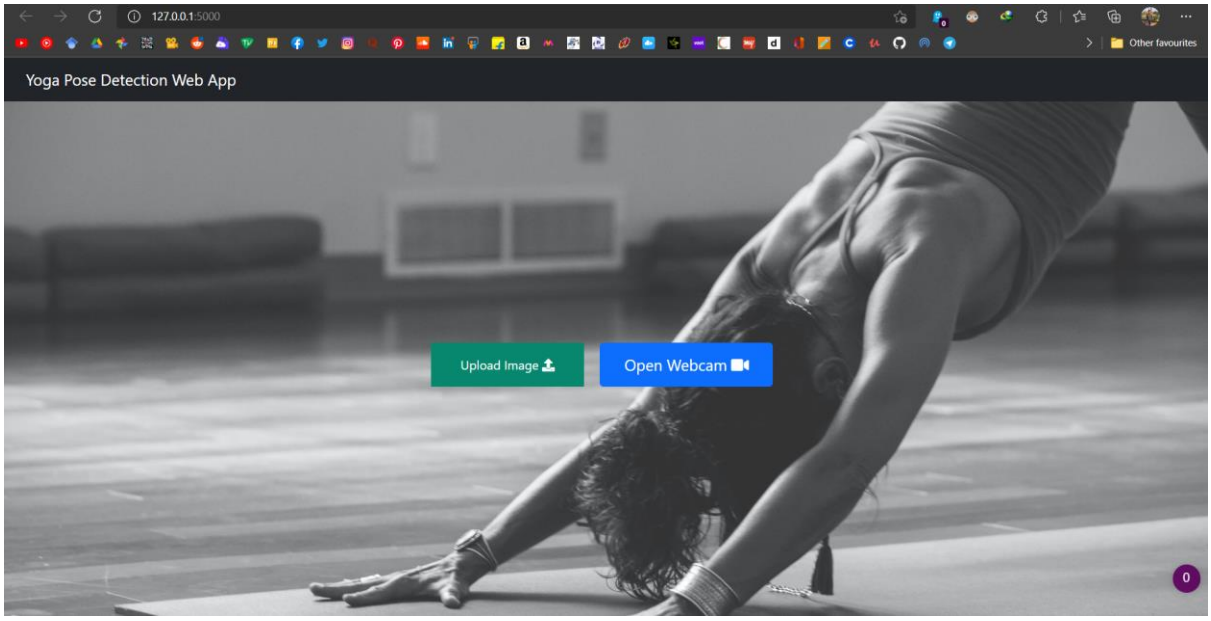
Design & Architecture

The project is comprised of multiple components which have been integrated into one to work effectively. The components are as follows:

- A webscraper was designed to created to obtain a dataset of images and separate it into five folders as per the requirement of the project.
- A piece of code is used to train a model to perform yoga pose detection. It includes loading the mediapipe model to obtain landmarks on the images, taking training dataset as input, data preprocessing, dividing the dataset, creating the model using multiple algorithms and testing the model using certain metrics. This is done with the help of sklearn, matplotlib and opencv.
- Now that we've trained the model, we need to use the model to perform predictions. This part is done using python in app.py, which includes different functions to take image as input via upload or run a live feed using webcam. The images are handled using opencv. Flask is used to work with the frontend, which is then integrated with HTML pages, styled using CSS and Javascript. Different components of functions have different endpoints which render a certain component according to functionality needed.
- The HTML component includes three files namely base.html, index.html and webcam.html. base.html contains a basic template for the frontend which is inherited by index.html and webcam.html. Index.html is rendered upon starting the server and opening the page for the project in a web browser. Webcam.html is rendered if webcam function is called from app.py.
- CSS is used to beautify the webpage by adding design elements and a beautiful background.
- The image request and response are handled using javascript.

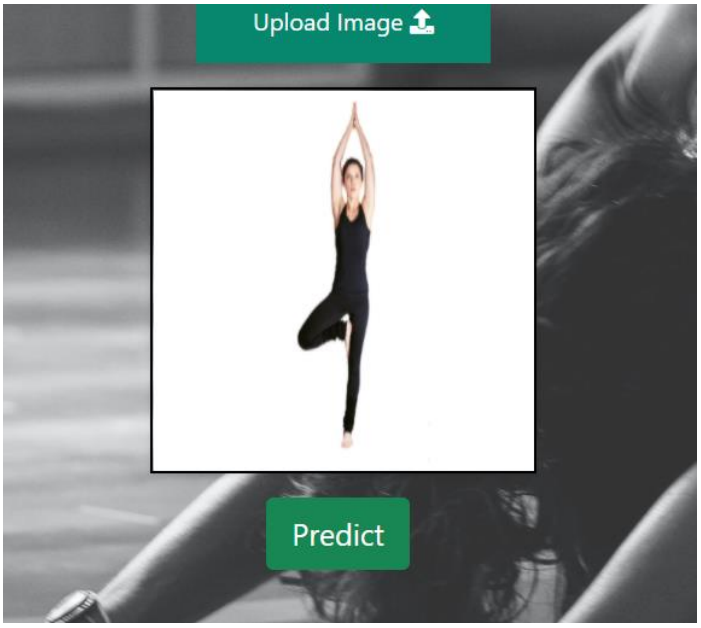
Result

The web application:



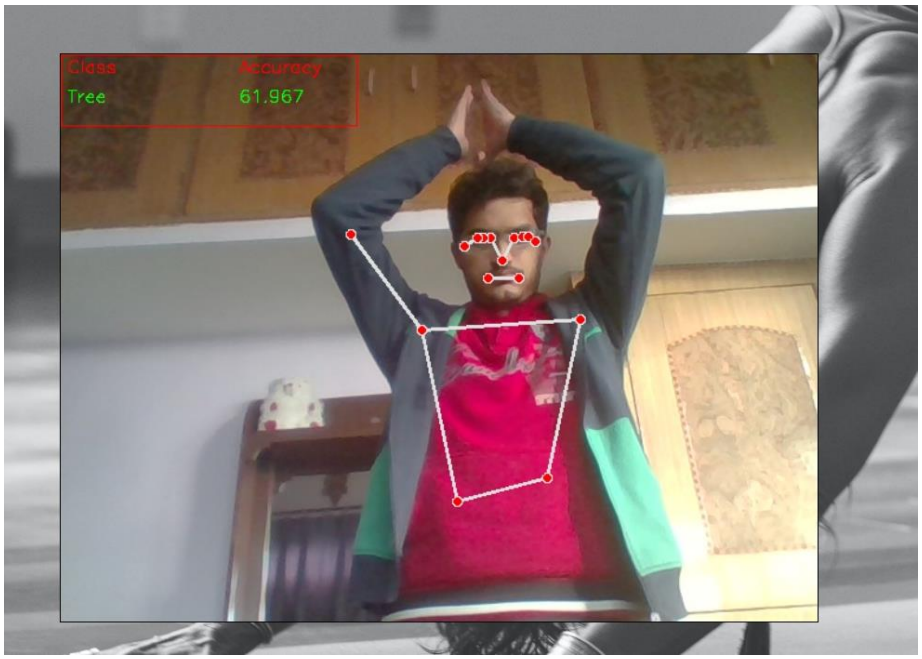
The GUI was built using Flask, JavaScript, HTML and CSS.
The welcome window consists of two options which include webcam and image upload functionality.

The upload image option opens a file explorer window, allowing the user to find the image he/she wants to upload.



Once the user has selected the picture, he/she can click on predict and wait for results.

Upon clicking the open webcam option the webcam starts and a live feed is started where yoga pose prediction takes place in real time.



Conclusion

The world has seen a tremendous rise in fitness community over the past decade. People from all walks of life are becoming much more conscious about their general health. Yoga has turned out to be a go-to choice for many but with the digital age looming on us, people find it difficult to properly learn and perform yoga with correct posture. Performing any pose with incorrect body posture can not only waste ones time & efforts but it can also prove to be quite counterproductive for your body as it increases risk of several muscle injuries. This problem can be avoided by creating a system that intelligently detects your current form and helps in rectifying it, thus ensuring that an individual gets maximum benefits out of their hardwork.

In our project we use MediaPipe to determine the accuracy of a yoga pose. MediaPipe analysis all available landmarks in human body & their position relative to each other in order to quantify the correctness of body posture. Our work will have a positive impact on all fitness-enthusiasts who will be able to easily judge the accuracy of their favourite Yoga-Pose and live with a healthy and sustainable lifestyle.

Future Scope

With the blazing speed at which computer vision is evolving, new pose estimation techniques and models will soon replace the tried and true methods of today. Using better models and techniques in the future for improvement of the model is something we'd certainly be interested in.

Extending our work with a number of other popular yoga poses is also on the cards.

Improving the model we develop with new, improved and thorough datasets with a higher usability could also be possible.

Adding a whole new design language to the project by introducing UI elements is also possible. Developing a web application or a native app will also help in the distribution of this idea.

References

[1] Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C.L. and Grundmann, M., 2020. Mediapipe hands: On-device real-time hand tracking. arXiv preprint arXiv:2006.10214.

[2] Henry Hollis, "Using computer vision to rate yoga poses" medium.com https://medium.com/@hollhc17/using-computer-vision-to-rate-yoga-poses-90a5dcb9aabe (accessed on 25th Sep)

[3] Kleinbaum, D.G., Dietz, K., Gail, M., Klein, M. and Klein, M., 2002. Logistic regression (p. 536). New York: Springer-Verlag.

[4] Svetnik, V., Liaw, A., Tong, C., Culberson, J.C., Sheridan, R.P. and Feuston, B.P., 2003. Random forest: a classification and regression tool for compound classification and QSAR modeling. Journal of chemical information and computer sciences, 43(6), pp.1947-1958.

[5] Tony Yiu, "Gradient Boosting Classification explained through Python" medium.com https://towardsdatascience.com/gradient-boosting-classification-explained-through-python-60cc980eeb3d

[6] Zhang, Min-Ling, and Zhi-Hua Zhou. "ML-KNN: A lazy learning approach to multi-label learning." Pattern recognition 40, no. 7 (2007): 2038-2048

[7] DataTechNotes: Classification Example with Ridge Classifier in Python

[8] Chiddarwar, G.G., Ranjane, A., Chindhe, M., Deodhar, R. and Gangamwar, P., 2020. AI-based yoga pose estimation for android application. Int J Inn Scien Res Tech, 5, pp.1070-1073.

[9] KUTÁLEK, J., DETECTION OF A YOGA POSES IN IMAGE.

[10] Jose, J. and Shailesh, S., 2021, March. Yoga Asana Identification: A Deep Learning Approach. In IOP Conference Series: Materials Science and Engineering (Vol. 1110, No. 1, p. 012002). IOP Publishing.