**DR. B. R. AMBEDKAR NATIONAL INSTITUTE OF TECHNOLOGY**



# FINAL YEAR PROJECT REPORT

# YOGA TRAINER USING DEEP LEARNING

***Degree of Bachelor in Technology***

***In Department of Computer Science and Engineering***

Session: 2019-2023

­­­

Mentored by: Presented by:

Dr. K. P. Sharma KHUSHI (19103055)

Assistant Professor Manan Jain(19103062)

Department of CSE Amanpreet kaur (19103011)

**Abstract**

Yoga must be one of the best choices for most of us when it comes to staying fit and healthy. But, the fact is that not all of us get good results and this might be because we are not following the postures in the right manner which is in turn because there is no proper guidance.

It is important to practice the right yoga poses when we don’t prefer an instructor to guide our postures. When at home, it is easier to watch youtube videos and practice yoga at home. But we cannot be sure if the poses that we practice is the correct one.

Pose estimation is one of the real-world problems, where developers try to develop some kind of machine instructors which can monitor the user’s postures and provides live results about the posture status if it is right or wrong.

An attempt to solve this problem has been made with the help of cutting edge technologies available. Most of the pose estimation techniques require users to have costly hardware and software components to run the program. In this project we tried to implement all the computations and ML model training on the web .

**Keywords:** Machine learning , PoseNet, Pose Estimation, TenserFlow, backend, pipelining, frontend.

**Description**

**PoseNet**is a computer vision and Machine learning model which helps in tracking full-size human poses with 17 key points as the output. Key points are nothing but the basic human skeleton joints on the body which on connecting resembles the human skeleton structure. The model was introduced by TensorFlow community and was trained using full-size human images (from eyes to ankles) and therefore the model works well when the user provides full-size human input.

Let us say, there is a trained PoseNet model which on implementing provides skeletal outputs referencing the key points in it. Now, in transfer Learning, the output from the PoseNet model ( i.e. the key points) is introduced as the inputs to another model and this model is trained for required poses that the user wants to estimate. The features from the PoseNet model allows you to ‘retrain’ or ‘reuse’ the model for a new custom task.

**Ml5.js** is a java script library that makes Machine Learning on Web with so much ease. With the help of Ml5.js, the following tasks are achieved .Data Collection- Where json files are created depending on how many types of data(pose) has to be classified. Training and classifying- Construct necessary neural network to train the data recorded in the json files.

Deploying the model- The neural network is trained and now, it receives input from the users and provides required output as per the data(pose) trained. To deploy the application on the website we have used frontend technologies like HTML,CSS,JAVASCRIPT,BOOTSTRAP and for backend server we have used Node js ,express js ,mongo db in our project.

**Previous Works**

**POSENET: Real-time Human Pose Estimation:**

POSENET is a machine learning model which allows for real-time human pose estimation in the browser. This is done by detecting where the key joints are for a person in the image/video. The model returns x, y coordinates for a total of 17 points on a person’s body along with a confidence score ranging from 0 to 1. It makes use of a convolutional neural network and single-pose or multi-pose decoding algorithm to decode poses, confidence scores, key point positions and key point confidence scores.

**Yoga Pose Estimation using POSENET vision model - Arjana Pradeep:**

In this implementation, 2 JavaScript libraries - P5.js and Ml5.js - were used to detect the yoga pose a person is performing in an image. The main basis here is the PoseNet model discussed above. It makes use of transfer learning, takes the output from PoseNet and uses another machine learning model to predict yoga poses based on that output.

**Scope and Contributions**

Pose detection has very wide and far reaching applications. It can be used to detect bad posture and help correct it. Here it is going to help people perfect their yoga poses without needing a trainer or expert all the time. This can be applied and scaled to countless other sports and disciplines. It can help cricketers perfect shots, help footballers perfect shooting techniques and so on. In Olympic disciplines like, synchronised swimming, synchronised diving, diving etc. it can help make judging more objective, helping if not replacing the current practices. These are just sports related benefits.

Further applications can be body modelling to create 3D models of humans, images and videos. This can be used in animation, video game development, graphic rendering etc. Furthermore it can be used in AR modelling to try on clothes in real time using a smartphone or embedded devices like smart mirrors.

It can also be used in advanced video surveillance, to detect suspicious or criminal behaviour, alert emergency services if a person is detected as being injured/unconscious etc. It can also help in the progress of Advanced Driver Assistance Systems (ADAS).

What our contribution to this is that we will use pose detection to detect yoga poses. A person can see what a yoga pose a person is performing in an image or video. Or select a pose that they want to learn and then do it perfectly themselves.

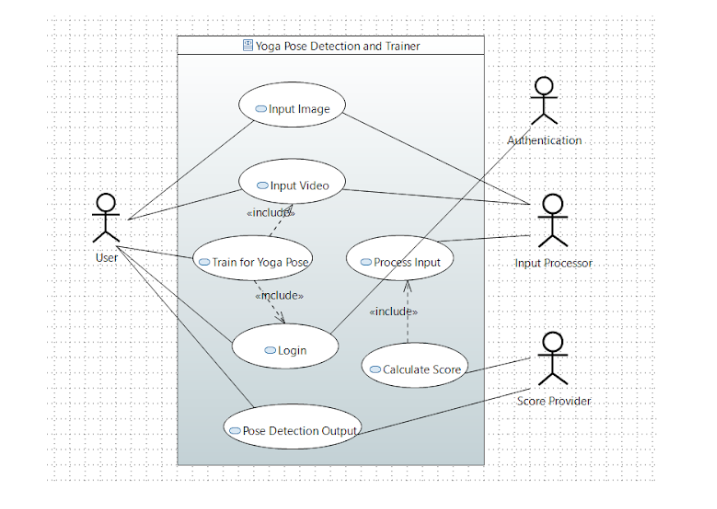
**Technical Feasibility**

Technologies that will be used in this project are: Machine learning, Deep Learning, Frontend and Backend Web Development and deploying a machine learning model on the web. All these technologies are fairly developed at the current point in time and it is possible to create a well working platform for our ambition.

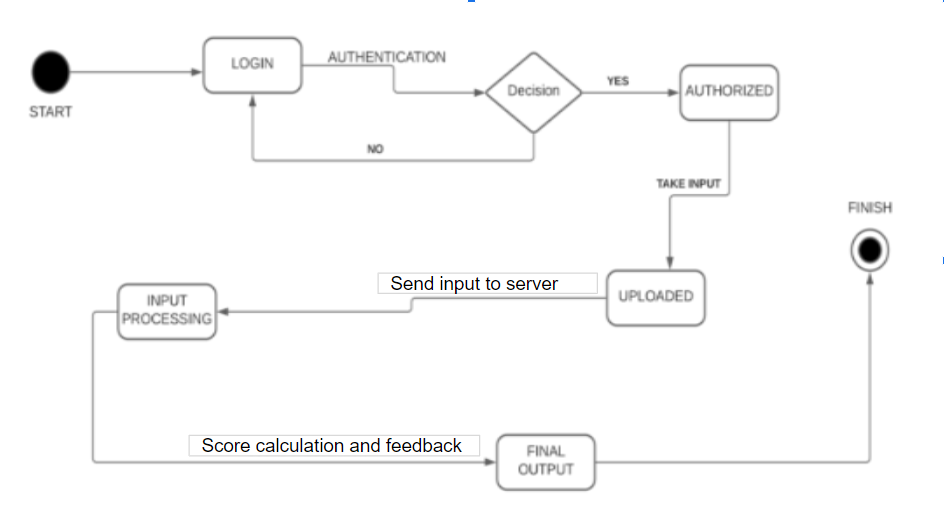
All team members have had exposure to the above listed technologies and are confident that they can deliver great results.

There are some constraints relating to time and communication due to the current scenario but it’s nothing that can be dealt with. So, it is feasible to create and deliver this project within the decided time frame.

**Use Case Diagram :**



**State Diagram :**

****

**Risk analysis**

Since this project will be real time application taking input and putting out outputs the most important factor will be managing the time complexity in our machine learning model we will train and it will have to judge movements continuously. So there will be a risk on making our app user friendly. We will use the earlier developed models as a backup If our own models fail to give the time complexity we really want.

**Work Plan/Schedule**

* Data Collection - Where json files are created depending on how many types of data(pose) has to be classified.
* Training and classifying - Construct necessary neural network to train the data recorded in the json files.
* Frontend of the site - Working upon the user Interface and the appearance of the website.
* Pipelining - Integration of the model with frontend using pickle.
* Deploying the model - The neural network is trained and now, it receives input from the users and provides required output as per the data(pose) trained.

**WORK FLOW**

Data Collection

TRAINING ML MODEL

Frontend of the site

Pipelining

Deployment

**Final Review of the Website**

1 week

4 weeks

2 weeks

1 week

4 days

**Suggested deliverables**

In this Project, we will deliver a free of cost, handy, easy to use, user-friendly web App which will assist the users to practice yoga without the help of a yoga trainer or consultant.

**References**

1. **Author:** Anjana Pradeep

**Title:** Pose Estimation yoga poses <https://github.com/Anjanapradeep/POSE_ESTIMATION-YOGA-POSES>

1. **Author:** Dan oved, **Irene Alvarado**, Alexis Gallo

**Title:** PoseNet

<https://blog.tensorflow.org/2018/05/real-time-human-pose-estimation-in.html>

1. **Author:** Mark Otto, Jacob Thornton

**Title:** Bootstrap

<https://getbootstrap.com/>

1. **Author:** Ryan Dahl

**Title:** Nodejs

<https://nodejs.org/en/>