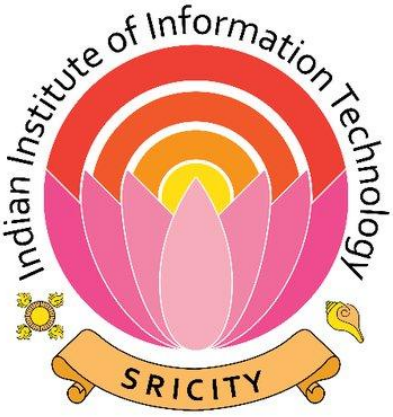
GESTURE DETECTION DURING GAMES: YOGA

A BTP REPORT

by

Prahitha Movva (S20180010108)
Hemanth Pasupuleti (S20180010131)
M.Sumanth Chowdary (S20180010096)



INDIAN INSTITUTE OF INFORMATION TECHNOLOGY SRICITY

15th May 2021

First Semester Report

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY SRICITY

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the BTP entitled "**GESTURE DETECTION DURING GAMES: YOGA**" in the partial fulfillment of the requirements for the award of the degree of B. Tech and submitted in the Indian Institute of Information Technology SriCity, is an authentic record of my own work carried out during the time period from January 2021 to May 2021 under the supervision of Prof. Himangshu Sarma, Indian Institute of Information Technology SriCity, India.

The matter presented in this report has not been submitted by me for the award of any other degree of this or any other institute.

Prahitha 15-05-2021 | 15-5-21 | M. Sumanth 15/5/21

(PRAHITHA MOVVA) **(HEMANTH PASUPULETI)** **(SUMANTH CHOWDARY)**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

(Dr. Himangshu Sarma)

ABSTRACT

In the light of recent events with the pandemic, Yoga has gained a lot of popularity across the globe due to its physical and mental health benefits. The objective of our project is to build a classification model that would recognize various Yoga postures or asanas from a small set of images using a combination of PoseNet and Deep Learning. This will involve a virtual trainer who will take us through a series of postures, demonstrating each pose and then instructing us to do it together. The poses should be typically held for 45 seconds to a minute, depending on the pose. There will be different levels and categories to choose from ranging from beginner to advanced and classic to modern yoga forms. At the end of each session, users get feedback on their posture and possible improvements, by the pose estimation model. This will be a minimal investment model and will help the users practice yoga from the comfort of their homes just by using their camera.

Currently, a variety of approaches are still being researched with respect to yoga pose recognition and development of self-training systems. We will also be including the aspect of exergames, also known as serious games, to our project through the feedback system. The overall system achieved an accuracy of 93% in classifying 4 different yoga postures from images.

CONTENTS

INTRODUCTION

Sports and physical fitness activities have always been an attraction to people around the world. With the advancements in computer vision and deep learning techniques, the thirst for intelligent fitness training systems has increased as it is one of the most flexible options for people involved in a rapidly-moving modern lifestyle. In this project, we focus on an approach that would help in building yoga self training systems for people to learn and practice yoga.

In the proposed work, a classification system was modelled to learn and automatically recognize yoga postures from images. The proposed system is capable of classifying up to 4 yoga postures, including: (1) Mountain (Tadasana), (2) Warrior1 (Virabhadrasana I), (3) Goddess (Utkata Konasana) and (4) Triangle (Trikonasana). Although the names of these postures originated from Sanskrit language, we will use the associated English posture names to refer them throughout this report.

1.1 MOTIVATION

Modern life is stressful. Long working hours, poor diets, inactivity, and increasing social isolation in the digital age have all contributed to rising rates of anxiety and depression. The year 2020 has presented a series of challenges, which made all of this even worse. With the current situation forcing us to spend more time indoors, learning any new skill can be frustrating and stressful, especially one like yoga that requires balance and physical coordination. Learning something like this all on our own can be very overwhelming. Our project is about designing an application to help people learn and practice correct yoga forms in an easy, stress-free manner.

Researches have shown that 15 minutes of yoga practice everyday changes the brain's chemistry and boosts mood. It also brings balance to body, mind and emotions, boosts metabolism, eases stress and anxiety levels and improves flexibility and strength.

1.2 WHAT ARE SERIOUS GAMES?

Serious games are games whose primary objective is not fun or entertainment, rather learning or practicing a skill, in our case -- YOGA. In recent years serious games have proven that it is possible to learn while we play. This teaching method is known as game-based learning.

1.3 ELEMENTS OF SERIOUS GAMES

There are 5 key elements to serious games:

1. Story

   The more sophisticated the argument (and the characters), the easier it will be to motivate the players and for them to get into the game

2. Gamification

   Namely, game dynamics which include things like rankings, rewards, badges or points system. This tends to animate and motivate the players building a healthy competition which encourages them to try harder

3. Immediate and individualised feedback

   The player interacts directly with the game and instantly receives a reward or punishment. Users can know where they have gone wrong and can work on doing better next time

4. Simulation

   Serious games, in most cases, imitate or reproduce real-life situations. These simulators make it possible for users to interact with a new reality and to practice the skills and concepts they have acquired through the game

5. Finally, the goal: To learn!

   The key element of serious games is to teach something. All the elements mentioned above can be found in endless commercial video games, but that does not necessarily make them serious games. Serious games have a purpose that is not recreational and that almost always has to do with certain educational or training aspects.

LITERATURE SURVEY

2.1 ExerCube vs. Personal Trainer: Evaluating a Holistic, Immersive, and Adaptive Fitness Game Setup.[1]

This paper gives us insights on how to make exergames better and give the experience as that of a personal trainer. Most existing commercially available exergames for consoles such as Wii Sports, Wii Fit, and Kinect Sports have been criticized for disregarding performance aspects that are key to successful workouts, like accuracy, precision and intensity. By validating inaccurate movements with successful game performance, these games lack feedback information regarding movement mistakes. Game controllers are also criticized for instrumentalizing the body too much and we can overcome this by using the laptop camera.

2.2 Sense of Familiarity: Improving Older Adults' Adaptation to Exergames. [2]

In this paper they show how exergames can be made inclusive by making them adaptable to older adults. They give the relationship between 5 factors that maximize the effectiveness of exergames and familiarity. This correlation is calculated by Spearman's correlation coefficient test and then we can see that prior experience is very significant in the overall familiarity.

Fig 1: Spearman's correlations between the proposed five factors and familiarity[2]

	Prior experience	Positive emotion	Repeated time	Level of processing	Retention rate
Familiarity	0.694	0.614	0.588	0.652	0.511
Sig.	<0.01	<0.01	<0.01	<0.01	<0.01

• Spearman's correlation coefficient test

Spearman's correlation coefficient is a statistical measure of the strength of a monotonic relationship between paired data- ranges between -1 and 1, the closer the value is to 1 the stronger is the monotonic relationship.

2.3 AI-Based Yoga Pose Estimation for Android Application. [3]

This paper surveys the different options that can be used for pose estimation such as OpenPose, Posenet, DeepPose, and concludes PoseNet is the best for deploying on an android application

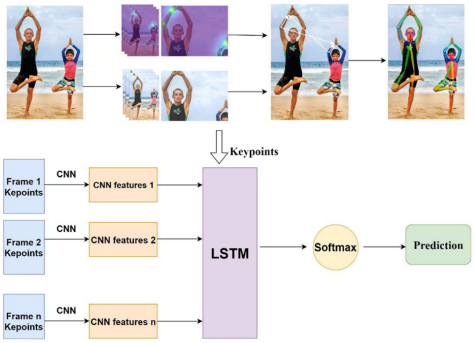
for doing yoga. DeepPose model used regression for XY locations for certain regions. This added complexity and weakened generalization hence performing poorly.

2.4 Real-time Yoga recognition using deep learning.[4]

This paper proposes a hybrid deep learning model using CNN and LSTM for Yoga recognition on real-time videos. The CNN layer is used to extract features from keypoints of each frame obtained from OpenPose and is followed by LSTM to give temporal predictions. Finally, the probability of each Yoga pose in a frame is given by the Softmax layer. ReLU activation is applied to keypoints of each frame for feature extraction. Batch normalization is applied to the CNN output for faster convergence. This is followed by a dropout layer which randomly drops a fraction of the weights preventing overfitting. The predictions were made with the help of openpose. A set of keypoints of 45 frames were extracted and sent into LSTM to make predictions.

Computations were done on a system with NVIDIA TITAN X GPU and Intel Xeon processor with 32 GB RAM. They have also tested the system in real time for a different set of 12 persons (five males and seven females) and achieved 98.92% accuracy.

Fig 2: System architecture: OpenPose followed by CNN and LSTM model[4]



2.5 Three-dimensional CNN-inspired deep learning architecture for Yoga pose recognition in the real-world environment.[5]

The approach they used in this paper was that instead of using only 2D conv filters, they have used 3D conv filters that use the spatial and temporal dimensions. This overcomes the need for a fixed number of inputs for the LSTM networks. During inference on a system with Nvidia K80

GPU, the proposed model takes 0.6s for the initial 16 frames clip of 4K resolution, which gets reduced to 45 ms for the subsequent clips.

2.6 NeuroPose: geriatric rehabilitation in the home using a webcam and pose estimation.[6]

This paper gives emphasis on real-time feedback and suggests that it is best with PoseNet. The implementation is done using PoseNet provided by TensorFlow.js with WebGL to access the local GPU resources when available. The scoring algorithm uses the joint location data to calculate the key joint angles.

2.7 BlazePose: On-device Real-time Body Pose tracking.[7]

The main conclusion that we derive from this paper is that, although BlazePose shows slightly worse performance than the OpenPose model on the AR dataset, BlazePose Full outperforms OpenPose on yoga/fitness use cases.

2.8 VNect: real-time 3D human pose estimation with a single RGB camera.[8]

This paper provides an impressive approach for 3D human pose estimation, but it comes with a catch, that is, it uses RGB-D cameras. We know that RGB-D cameras provide valuable depth data which greatly simplifies monocular pose reconstruction. However, RGB-D cameras often fail in general outdoor scenes (due to sunlight interference), are bulkier, have higher power consumption, have lower resolution and limited range, and are not as widely and cheaply available as color cameras. Computations were performed on a 6-core Xeon CPU, 3.8 GHz and a single Titan X (Pascal architecture) GPU. The CNN computation takes ≈18ms, the skeletal fitting ≈7-10ms, and preprocessing and filtering 5ms.

METHODOLOGY

3.1 DATASET PREPARATION

The main idea was to make sure the user performs the right given pose. For that we need different types of yoga poses. We use PoseNet to get the pose key points from the given image and then we use these key points to further compare and analyze with the user's key points. Initially, we have used cosine similarity to compare the two poses since the key points can be represented as a vector. But the results are not quite satisfactory, so we have moved on further to train a simple neural network. So, we have collected a few yoga pose images from the web and from Kaggle[9] and A. Marchenkova[10][11] dataset. These images are further sent into PoseNet to predict the poses. The PoseNet gives 17 key points with their x and y coordinates. These key points are further flattened to get a vector of length 34. Thus, having N images will give a feature matrix of dimensions Nx34.

From the below image, we want to show that we took into account different genders and environments, so as to make our model work in various conditions and for different people.

Fig 3: Images we used to train our model taken from A. Marchenkova[10][11] dataset for the mountain pose

3.2 DATASET DESCRIPTION

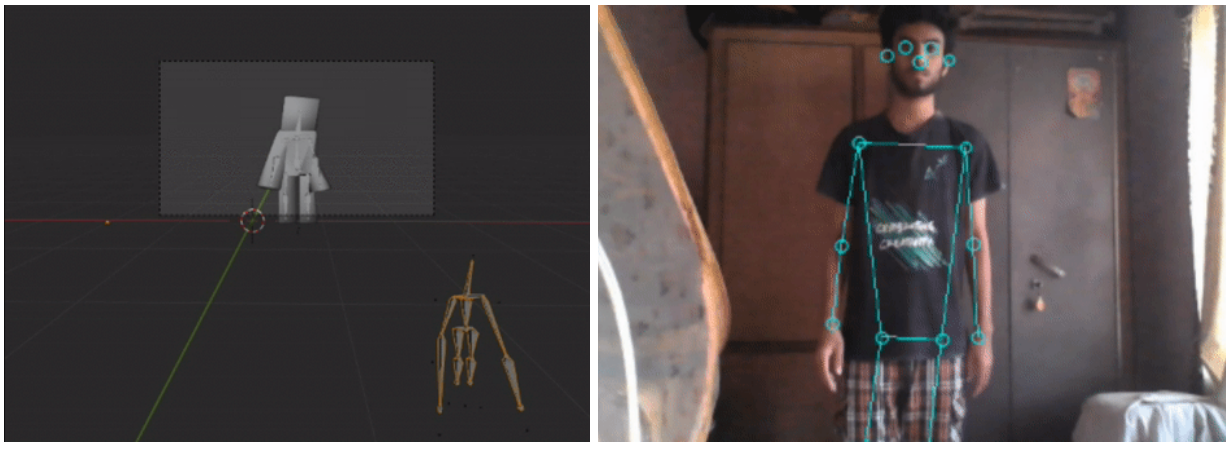
The yoga poses dataset was taken from Kaggle[9] and A. Marchenkova[10][11]. Together, both the datasets contain images of various poses -- Mountain, Goddess, Triangle, Warrior -- that we used to train our model. Each pose consists of approximately 200 images. We also performed image flipping for data augmentation.

Note: The training data used for the model with the moving SVG consists of 222 Training images, 110 Validation and 80 test images.

3.3 AN ATTEMPT WITH BLENDER

We have tried to create a model in blender from scratch, so we have used the PoseNet pytorch implementation and tried to integrate it with the blender resulting in a 3D real time moving model. MobileNet V1 has been used as the backend model for PoseNet with 450x315 as the input resolution. We have used these settings for higher frame rate but this comes at an expense of accuracy. To increase the accuracy, ResNet 50 can be used with higher input resolution and output stride, but this requires some decent GPU inorder to get a good frame rate. The key points provided by the PoseNet are normalized by the height and width of the pose and are further normalized to match with the coordinates of the blender model. To get the animation, we simply start the program and the key points will get recorded, these key points are now representing the coordinates of the blender model. So, when we are finished with our act and we pause the video we can get the recording of what we did.

Fig 4: An attempt with Blender - moving the arms of a 3D model from real-time coordinates taken from camera feed



3.4 USING P5.JS AND ML5.JS

Here, we have a small yet efficient neural network which is trained using Tensorflow. We then saved the model and the metadata in a json file and the weights in a binary file. These are then loaded using the ml5 neural network function - which is the core of the program. We then calculate the trainer angles using the keypoints from the pose with the most accuracy. We also calculate the angles from the video feed in a similar way. This is done using the arctan function which returns the angle between two points. Then we compare the angles between the trainer and the input. If this difference is greater than 30 then we append the corresponding keypoint into a list and show it in red color. These keypoints are also printed in the console. By alerting of the wrong posture, we will make it easier for the users to feel comfortable while exercising. The poses are classified using the neuralNetwork.classify function for which we set a limit of 75%, so if the pose is classified with a 75% confidence it is taken as correct.

Fig 5: Workflow of pose estimation

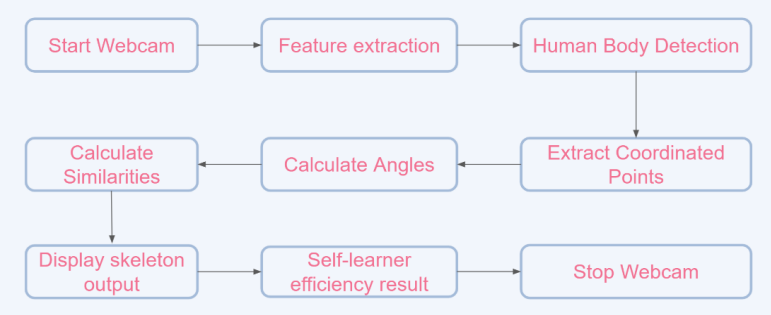
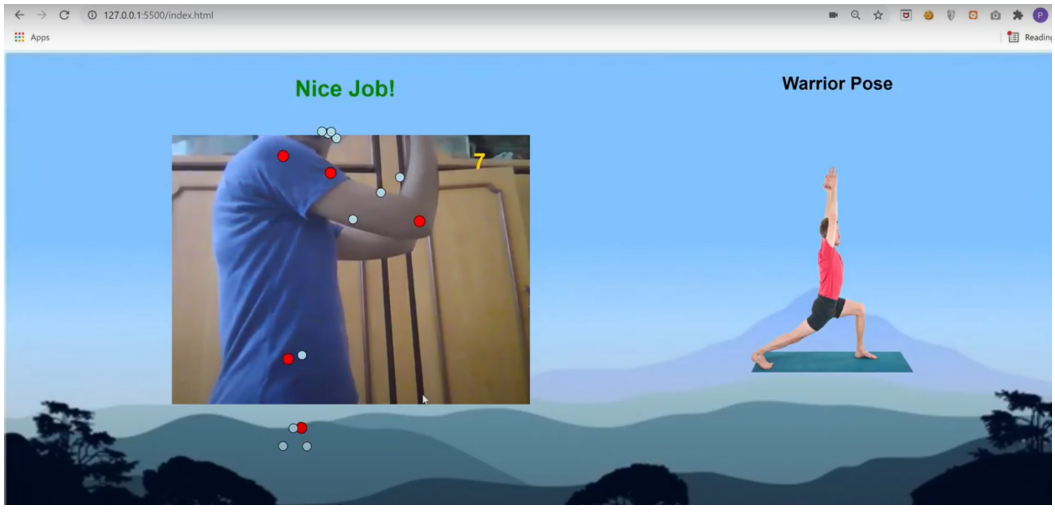


Fig 6: A screenshot from our application using p5.js and ml5.js with wrong keypoints highlighted in red. The timer also counts down from 10 to 0 before moving on to the next pose



3.5 USING TF.JS

TensorFlow.js provides an amazing open source animator that uses PoseNet and FaceMesh to move the model in real time. This model is a 2D SVG based model, so we do not have any 3D model to move around as of now. The key idea here is to use the outputs provided by the PoseNet and send them into a simple neural network to predict the pose. If the model is able to predict the pose i.e. the user continues to be in the same pose, then there will be a countdown so that the person has to hold on to his pose and if the person moves i.e. if no pose is detected then the countdown can be stopped. The following is the final result:

Fig 7: Moving SVG demo

RESULTS

4.1 Evaluation criteria

The performance of a model is based on how accurately the trained model recognizes a yoga posture from a new set of data. We also know that, the higher the accuracy, the better the model is, in terms of performance. In addition to the accuracy, we also consider the test loss which determines how well the model has learnt to perform this image recognition task.

4.2 Results from the model

The training was initially done in TensorFlow and then the model was exported and converted into TensorFlow.js. We use a 2 layer neural network with 16 units in the hidden layer. The neural network accepts the flattened human pose key points provided by the initial PoseNet. These can be used to classify the given pose. Here, as we have mentioned earlier we can see that if we have N images then our feature matrix will be of dimensions Nx34 where the 34 features we have are the coordinates of various key points given by the PoseNet. Now that we have the feature matrix, the labels can be one-hot encoded based on the number of classes to get the target dimensions of NxC where C is the number of classes we have.

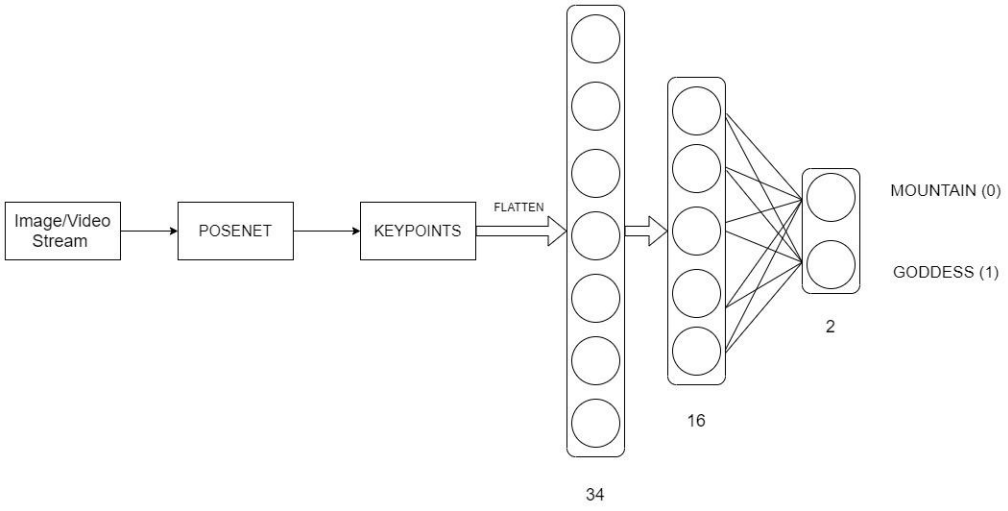


Fig 8: Model Architecture

Note: This model has been trained separately twice, once for the ml5.js and then for the moving SVG. We are listing the stats here for the moving SVG only.

The dataset contained 412 images of 2 classes namely Mountain and Goddess. This data is further divided into 222 train, 110 validation and 80 test images respectively. The model has been trained for 200 epochs with Stochastic Gradient Descent and CategoricalCrossEntropy loss as the loss function. The model is able to achieve **93.75%** of accuracy on the test dataset.

The learning curves can be visualized as follows:

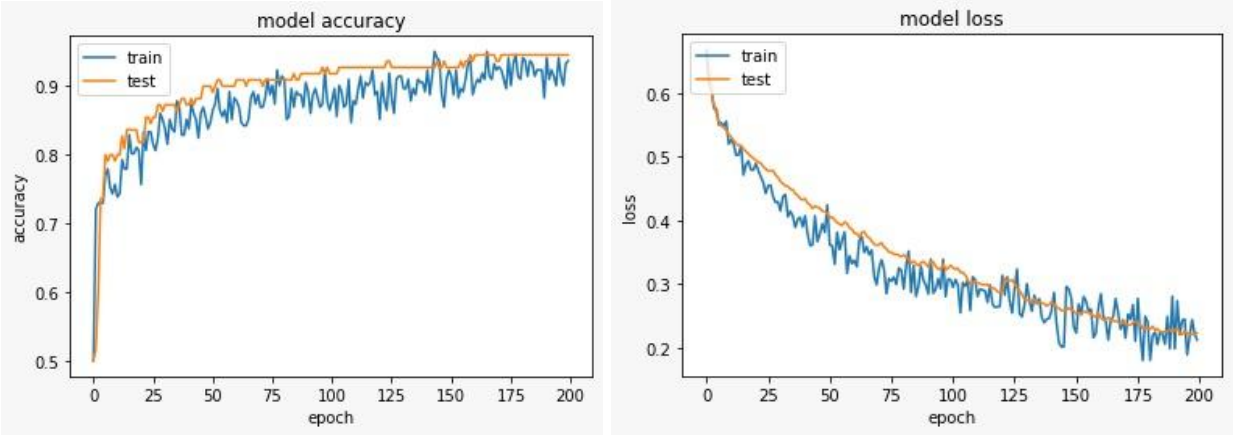
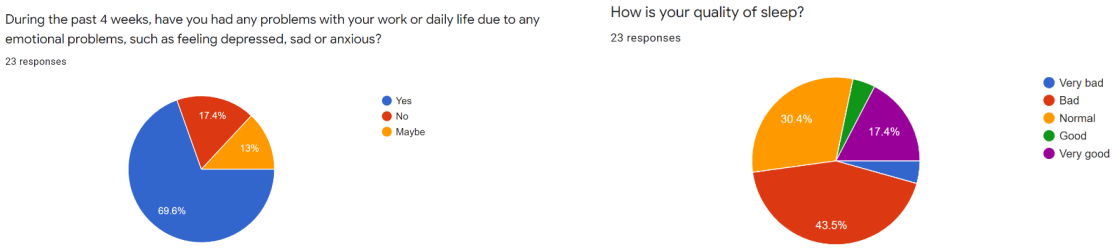


Fig 9: Model accuracy and Model loss (learning curves)

4.3 Results from the survey

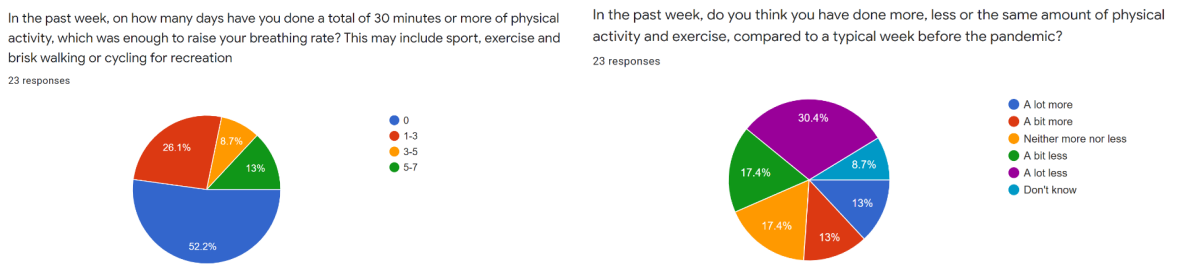
We sent out a survey a couple days before our evaluation and here, we document our observations from the responses that we have got. 62% of the respondents are female and around 80% of the total responses were from the 18-25 years age group.

Fig 10: Responses for questions on emotional problems and quality of sleep



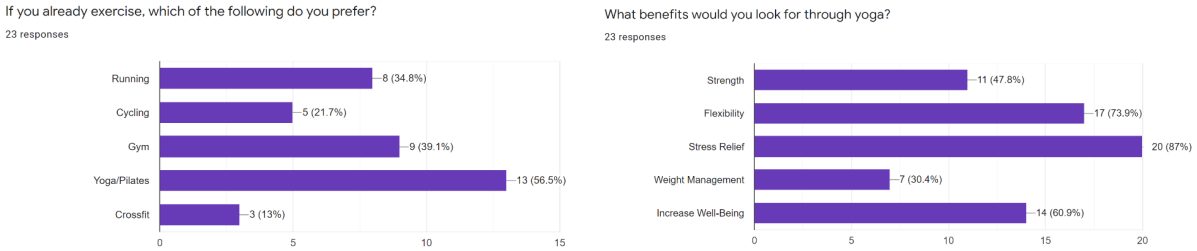
Here we can see that a lot of the respondents struggled with emotional problems and the lack of quality sleep.

Fig 11: Responses for questions on frequency and amount of physical exercise



We can also observe that more than half of them did not have any kind of physical activity in the past week. Here, we can see that most of them responded that they observed a reduction in the amount of time they spent on physical activity and exercise during the pandemic.

Fig 12: Responses for questions on which exercise they chose and what they expect from yoga



The respondents also preferred yoga over other exercises and primarily looked for stress relief and flexibility through it. Hence, the results that we got through our survey support our idea behind choosing yoga for our project and strengthen the need for the same.

CONCLUSION

5.1 Challenges

- Limited computational resources. Real-time pose estimation needs high computational capabilities which are not supported by our personal laptops currently.
- Limited resources on how to move an avatar in a game environment without special equipment, that is only using a laptop camera.

5.2 Future work

For having a fully-functional intelligent fitness-training system, we have to further have a plot or story around which the game revolves. We also should include a 3D trainer for the poses instead of having them shown as 2D images, however, this can be a bit challenging as we have to perform rigging and the animation by ourselves since they are not publicly available. We also plan on including some music for a more wholesome experience. Further, including a progress chart and reward system will be beneficial for the users as they increase accountability and a sense of achievement. Finally, a possible area of investigation would be to try a different set of yoga postures and see how the model can learn the different arrangement of features
from the images and its prediction rates.

LIST OF FIGURES

LIST OF ABBREVIATIONS

CNN - Convolutional Neural Network

LSTM - Long Short Term Memory

ReLU - Rectified Linear Unit

GPU - Graphic Processing Unit

CPU - Central Processing Unit

WebGL - Web Graphics Library

AR - Augmented Reality

RGB-D camera - Red Green Blue - Depth cameras

SVG - Scalable Vector Graphics

REFERENCES

[1] Anna Lisa Martin-Niedecken, Katja Rogers, Laia Turmo Vidal, Elisa D. Mekler, and Elena Márquez Segura. ExerCube vs. Personal Trainer: Evaluating a Holistic, Immersive, and Adaptive Fitness Game Setup. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI 2019)

[2] Hao Zhang. Sense of Familiarity: Improving Older Adults' Adaptation to Exergames. In Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems (CHI EA 2019)

[3] Chiddarwar, Girija & Ranjane, Abhishek & Chindhe, Mugdha & Deodhar, Rachana & Gangamwar, Palash. AI-Based Yoga Pose Estimation for Android Application. International Journal of Innovative Science and Research Technology (IJISRT 2020)

[4] Yadav, Santosh & Singh, Amitojdeep & Gupta, Abhishek & Raheja, Jagdish. Real-time Yoga recognition using deep learning. Neural Computing and Applications (2019)

[5] Jain, S., Rustagi, A., Saurav, S. et al. Three-dimensional CNN-inspired deep learning architecture for Yoga pose recognition in the real-world environment (Neural Computing & Applications 2020)

[6] Steven R. Rick, Shubha Bhaskaran, Yajie Sun, Sarah McEwen, and Nadir Weibel. NeuroPose: geriatric rehabilitation in the home using a webcam and pose estimation. In Proceedings of the 24th International Conference on Intelligent User Interfaces: Companion (IUI 2019)

[7] Valentin Bazarevsky and Ivan Grishchenko and Karthik Raveendran and Tyler Zhu and Fan Zhang and Matthias Grundmann. BlazePose: On-device Real-time Body Pose tracking (Google AI Blog, arXiv 2020)

[8] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. VNect: real-time 3D human pose estimation with a single RGB camera. ACM Transactions on Graphics (July 2017)

[9] https://www.kaggle.com/niharika41298/yoga-poses-dataset

[10] Train: https://drive.google.com/drive/folders/1vsBgnEf2NcQNvlCnIZr0P1m2_XXLhLlQ

[11] Test: https://drive.google.com/drive/folders/1WW7n4C01nBgBGNsQBLiRrsTB7h0WFIOB

[12] https://neurohive.io/en/news/yoga-82-new-dataset-with-complex-yoga-poses/