**A PROJECT REPORT**

**on**

**“Yoga Trainer using poseNet”**

**Submitted to**

**KIIT Deemed to be University**

**In Partial Fulfilment of the Requirement for the Award of**

**BACHELOR’S DEGREE IN**

**COMPUTER SCIENCE AND ENGINEERING**

**BY**

**PRATYUSH AANAND**

**ASHUTOSH MISHRA**

1905189

1905600

**UNDER THE GUIDANCE OF**

**SANKALP NAYAK**



**SCHOOL OF COMPUTER ENGINEERING**

**KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY**

**BHUBANESWAR, ODISHA - 751024**

**May 2023**

A PROJECT REPORT

on

“Yoga Trainer using poseNet”

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR’S DEGREE IN COMPUTER SCIENCE & ENGINEERING

BY

PRATYUSH AANAND

ASHUTOSH MISHRA

1905189

1905600

UNDER THE GUIDANCE OF

SANKALP NAYAK



SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAE, ODISHA -751024

May 2023

KIIT Deemed to be University

School of Computer Engineering

Bhubaneswar, ODISHA 751024



CERTIFICATE

This is certify that the project entitled

“Yoga Trainer using poseNet“

submitted by

Pratyush Aanand

Ashutosh Mishra

1905189

1905600

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Sci-ence & Engineering) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2022-2023, under our guidance.

Date: 8th May 2023

Sankalp Nayak

Project Guide

**Acknowledgements**

We are profoundly grateful to **SANKALP NAYAK** of **School of Computer Engineering, KIIT, BBSR** for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion. .....................

PRATYUSH AANAND

ASHUTOSH MISHRA

**ABSTRACT**

With the globalization in 1990’s, India opened itself to the world and finally became the giver of the ancient civilization it has nurtured for 3-4 millennium. One such discipline was yoga, which is making strife worldwide with the benefits it giving a world who is at the breaking point of its stress and mental burden. Even though it’s making strife in the west the reach has still not reached the level it deserves. One reason for this could be the lack of commercialization done in this field, we lack proper trainer who can while training also make the one performs it understand it’s significance.

With the goal of making yoga more accessible and user friendly we took a deep dive into our field of Machine Learning, in order to come up with a solution. With the various strives made by many experiences folk working in the software industry of Silicon Valley, we were able create a yoga ai, using many open-source platform such as PoseNet, TensorFlow, ML5, P5, Google teachable machine. We created a web page hosted on GitHub that will train user 6 yoga poses with an interactive user interface.

**Keywords:** PoseNet, ML, TensorFlow, ML5, P5.

Contents

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | Introduction | | | 8-9 |
| 2 | Basic Concepts/ Literature Review | | | 10-12 |
| 3 | Problem Statement / Requirement Specifications | | | 13-15 |
|  | 3.1 | Project Planning........................... | | 13 |
|  | 3.2 | Project Analysis (SRS)................. | | 13-14 |
|  | 3.3 | System Design ………………….. | | 14-15 |
|  |  | 3.3.1 | Design Constraints …… | 14 |
|  |  | 3.3.2 | System Architecture (UML) / Block Diagram … | 15 |
| 4 | Implementation | | | 16-28 |
|  | 4.1 | Methodology / Proposal ........................... | | 16-20 |
|  | 4.2 | Testing / Verification Plan ……………. | | 20-23 |
|  | 4.3 | Result Analysis / Screenshots …………. | | 23-25 |
| 5 | Standard Adopted | | | 26-28 |
|  | 5.1 | Design Standards . . . . . . . . . . . . . . . | | 26 |
|  | 5.2 | Coding Standards . . . . . . . . . . . . . . | | 27 |
|  | 5.3 | Testing Standards . . . . . . . . . . . . . . . | | 27-28 |
| 6 | Conclusion and Future Scope | | | 29-30 |
|  | 6.1 | Conclusion ……………………….. | | 29 |
|  | 6.2 | Future Scope ………………………. | | 29-30 |
| References | | | | 31 |
| Individual Contribution | | | | 32-33 |
| Plagiarism Report | | | | 34-38 |

List of Figures

|  |  |  |
| --- | --- | --- |
| 3.1 | Program Logic Flow | 16 |
| 3.2 | Flow Diagram of Project. | 16 |
| 4,1 | Entry screen | 23 |
| 4.2 | Tadasana | 24 |
| 4.3 | Vrksasana | 24 |
| 4.4 | Adho Mukha Shavasana | 24 |
| 4.5 | Virabhadrasana | 25 |
| 4.6 | Virabhadrasana 2 | 25 |
| 4.7 | Utkatasana | 25 |

*YOGA TRAINER USING POSENET*

Chapter 1

Introduction

Anxiety, stress and pressure are condition that everyone experience in some phase of their life. But with digitalization and world becoming small, people experiencing these thing on scale never seen before in history of mankind. According to [American psychological association](https://www.apa.org/news/press/releases/2000/12/anxiety#:~:text=In%20fact%2C%20the%20studies%20find,patients%20did%20during%20the%201950's.) studies “normal kids today exhibit anxiety at a severe scale not even seen in 1950’s psychiatric patients”. So, it exaggeration to say that it’s “[The Health Epidemic of the 21st Century](https://hcahealthcaretoday.com/2019/04/30/stress-the-health-epidemic-of-the-21st-century/)”. There is multiple thing that are prompting this. Feeling of lots of pressure, the digital reality in which most of us live has shown that no matter how good you do there is always someone out there doing better, which leads to a downward spiral of self-doubt and inadequacy among people prompting them to set unrealistic target and falling under pressure of not achieving those. Another could be the situation when something drastic changes in life and people are unable to cope with situation. A sense of feeling that, thing are too much out of your control. Which leads into the feeling of nihilism, meaning nothing they do can do anything in the grand scheme of things. Or it could be toxic environment. Too much workload, long working hours, unclear expectation of your work, insecurity about promotion in workplace and economic crash (recession) that can lead to sudden termination rendering them unemployed and with an uncertain future. The next step of this is panic attack, nervous breakdown etc. Panic attack is sudden feeling of fear, anxiety and helplessness that leads to problem of being unable to breath. This leads to one thinking of ways, prevent situation like these, it may surprise you but the answer to the devil of 21st century comes from 3 millennium old practice, we Indians called ‘Yoga’.

Yoga is one of the 6 spiritual disciplines of Hindu philosophy toward the enlighten and attainment of higher consciousness, with the aim of bringing union between body and mind. It’s an art and cultural heritage of India. Derived from the ‘YUJ’ which in Sanskrit means ‘to join’, ‘to yoke’, to unite’. It’s mentioned in ancient Hindu text ‘Rig Veda’ making it almost 3500 years old. This 3-millennium old practice doesn’t show its age and it’s going strong among the masses. As the text mentions with the goal of achieving perfect harmony between man and nature, show how our ancestors in their chalcolithic phase were able to figure these basic things even before the industrialized counterpart ever could. Modern scientist has coined that everything we called the universe is derived from quantum firmament, which further proof the philosophy that to achieve enlightment one has to become part of the something bigger, the universal consciousness. And the one who achieve this union is called the ‘yogi’. This state is also called Mukti, nibbana or moksha (state of being liberated).

Often associated with sage Patanjali, yoga is associated with realizing one’s true self, which can make one aware of his self-worth. Along with this realization of self being, it also has many health benefits like improving flexibility enriching strength and balance along with providing a good body posture. Relive from back pain, associated with the youth due to the promotion of sedentary lifestyle. Provide ease to the one suffering from arthritis. Improve cardiovascular activity. Relaxes both body and mind, by giving a breather from the constant bombardment of information. Enriches one with energy and a good mood, enough to take any difficult task head on. And off course managing stress and helping with mental health, quality of life as quoted by National Institute of Health. And if one wants, they can join yoga groups, surrounding one with likeminded people constantly pushing each other and striving better while supporting each other.

But everyone is aware of these things, but the main problem today keeping one from becoming a perfect yogi is lack of time, lack of access. As any other exercise, yogic poses require a perfect symphony of body according to the pose the yogi is performing, otherwise one can’t reap the benefit or even risk injuries. Another for being an all-encompassing discipline there aren’t any large commercialization done to this art, and hence there is a lack of proper trainer, making it quite restricted. But that were a data engineer comes, everything and every problem is being solved using multiple lines of codes to the extent it can be. So, using AI and deep learning we are going create a machine learning model which will be capable of learning yogic poses and give real time feedback to the one performing, and hence saving them their precious time, making yoga once again all accessible artform.

*School of Computer Engineering, KIIT, BBSR*

*YOGA TRAINER USING POSENET*

Chapter 2

Basic Concepts/ Literature Review

The project has multiple stage but beginning start with detecting user movement in real time, for that purpose we took to Google’s TensorFlow’s Pose NET. Pose net is a vision technique which identifies joints in human body, it doesn’t exactly identify a person, just their joints. It’s having many interactive uses in augmented reality (wherein something can be added to you face, body like a snapchat filter) and fitness (like the one we are going to use). It runs on Tensorflow.js, which makes it’s fairly accessible and allow simple functions in JavaScript just to run it, the only requirement being a decent webcam and browser. Another advantage of it being run on browser is privacy, as no pose data use for the simulation leaves the user computer. There are two types of pose Net, either for single pose detection or multiple pose detection. In our program we are going to use single pose detection as it’s faster and easier.

Pose Net uses a convolutional neural network, whose input is an RGB image and after processing the pose is decoded and the output is pose (it’s an object that contains a list whose content includes key points and confidence score for each person) pose confidence score (how much the model is sure about the pose it’s detecting, just like probability it ranges between 0-1), key point position (coordinated of the predicted joints in 2 dimension that is in (x,y), with the capabilities it has it’s able to detect 17 joints for a single person), key point confidence score (how sure is it about the joints being their, similar to pose confidence score this also ranges between 0 and 1).

Coming back to single pose estimation, out cnn is capable enough to identify 17 joint in a single person. That being the following

1. Right Eye
2. Left Eye
3. Right Ear
4. Left Ear
5. Nose
6. Right Shoulder
7. Left Shoulder
8. Right Elbow
9. Left Elbow
10. Right Wrist
11. Left Wrist
12. Right Hip
13. Left Hip
14. Right Knee
15. Left Knee
16. Right Ankle
17. Left Ankle

Being some of the most important joint of human body. But being the faster one of the two algorithm, it wouldn’t be correct to not mention it’s shortcoming. Once another person enters the frame the poseNet get’s confuses and start associate some point to one person and some to the other one e.g., Right Eye of person 1 and Left Eye of person 2. Some of the input condition for this to work as mentioned on the [TensorFlow](https://blog.tensorflow.org/2018/05/real-time-human-pose-estimation-in.html) includes

* Input image element: the input image fed through an html element, it can be either video tag or image tag. Another fact that the element being fed into the input should be necessarily square.
* Image Scale Factor: between 0.2 to 1. The default condition provided by TensorFlow is 0.5. If you put a smaller number in hopes of scaling down the image, it will result in loss of accuracy or at it’s cost.
* Flip Horizontal: It has Boolean input, if false the image will appear flipped or you will see mirror image. But when using the webcam which already flips the image, there it’s value should be true
* Output stride: multiple of 8 e.g., 8,16,32. By default 16 is the value. It’s the no of layers our CNN going to use. If you put a high value the output will faster but at the cost of accuracy, conversely if you put smaller value the output would be the trade off and accuracy will increase.

Now lets shift our focus on the output side. As mentioned earlier we will have a pose object which will have pose confidence score and an array of 17 key points. And each of these key points will have key point position and key point confidence score in it. Then again, the key point position will have x (ordinated) and y (abscissa) position. These are the basic things one need to understand and deploy pose net.

According to TensorFlow Blogpost, Pose Net is trained on both ResNet and MobileNet. But due to the large size and slow loading of time of ResNet we are only going to use MobileNet. Much to our relief the our model i.e. PoseNet has a tendency of being scale invariant. Even if the image is downscaled it will predict the position using same scale as the original image. Output stride not only decided size of layer of the CNN and output produced by it but it is also the variable which determines downscaling of output in comparison of input. Hence accuracy of our PoseNet can be increased by setting output stride to a higher value.

In order to achieve a large output resolution, input striding in the layers of model needs to reduced, which can do through setting our output stride as 8 or 16. This value also allows us to implement Atrous convolution to the layers, which otherwise couldn’t be applied if the output stride would’ve been 32. Also according to TensorFlow Blog, unlike TensorFlow TensorFlow.js doesn’t support Atrous convolution so the team their have added PR to make it accessible.

During image processing in TensorFlow the return value happens to be a heatmap and offset vector. Using these two one can, find the high confidence area of key points off course after they are decoded. The heatmap and offset vector are both a 3D tensor along with a height and width.

*School of Computer Engineering, KIIT, BBSR*

*YOGA TRAINER USING POSENET*

Chapter 3

Problem Statement / Requirement Specifications

TensorFlow.js PoseNet which enable us to detect 17 key point in human body, taking that output we are going to build a classification model which will capable/trained enough to differentiate between different poses, these poses will be yoga poses. Using all this data we are going to create a yoga AI and deploy it. So the visitor can learn how to perform Yoga.

3.1 Project Planning

Our first step to get a grasp on how to work with PoseNet, which will provide necessary tools that are required to build our project. Once a familiarity is build our focus will be to learn the necessary library required to bring this idea into fruition, in our case these are ‘ML5’ and ‘P5’, former will help us bring machine learning capabilities to JavaScript and latter will help to visualize these information. The project is streamline one but it has multiple moving pieces. Our next target will be to create a basic html file so that we can begin our journey. Once stated we have to setup webcam in order to collect real-time data on which we are going to work. These are the basic function required by any PoseNet program. Moving on the ML side of things we have to first create our classifier model which help us classify the poses. Next will be the implementation and coding for the actual website we are going to design. Our front end material will show visitor’s footage, in which the PoseNet will work, and another division will contain instruction to be followed which will be the pose, the time for which they have to hold the pose. We have to develop a logic flow which help visitor to navigate through all the poses and mechanisms for counting how the user is performing and time out. Fusing the html and JavaScript file deploying will be the next thing on our plate. As our program hardly has any back end so deploying will be easier on GitHub itself. With which are project will be accessible to anyone who are willing to become a yogi.

3.2 Project Analysis

This project is not hard but contain many moving element which have to done accordingly in order to achieve our desired goal. It’s mixture of multiple discipline from web development to machine and deep learning. Before moving on a basic plan on what we need is essential. Starting with a html file which will provide us a canvas on which we can work and begin the project. A style sheet although not playing a major role it will be there to format how different html element comes together. It can be emphasized how much role the P5, ML5 will play in this project, not only making the code easily usable but easily understandable. Their implementation on html can be done in 2 ways either will link them directly through the web or attach a physical version. Seeing that these libraries are at constant updating by their respective authors, our choice would be use a locally available version to avoid situation like some features not working. Then we need some image file of the asana that we are going teach the machine, it will help visitor of the site to visualize how to perform them, just like libraries it’s better to have a local version of them so that they are not subject to change. And off course our project cannot complete until all these element do not come in harmony and perform the task they are supposed to do that, would be the task for our JavaScript file, which will pave the way for us.

3.3 System Design

3.3.1 Design Constraints

The whole project is done on Microsoft’s Visual Studio Codes, with necessary package being downloaded.

1. Live Server v5.7.9: this is an VS code extension which creates a virtual port through which you can access your website this will give the emulation of how the website will work after it’s been deployed.
2. The virtual emulation of the project was run on Google Chrome (111.0.5563,147 (Official Build) (64-bit) (cohort: Stable)) which was on JavaScript V8 11.1.277.17 at the time of project development.
3. Ml5.min.js - @license Copyright 2019 Google LLC.
4. P5.js v1.1.0 February 29 2020
5. Google Tag Manager: allows quick and easy update of tags.

PC requirement

1. Intel® CoreTM i5-8250U CPU @ 1.60 GHZ 1.80 GHz
2. 16.0 GB Ram
3. 64-bit operating system, x64-based processor
4. Windows 11 Version 22H2

3.3.2 System Architecture **OR** Block Diagram

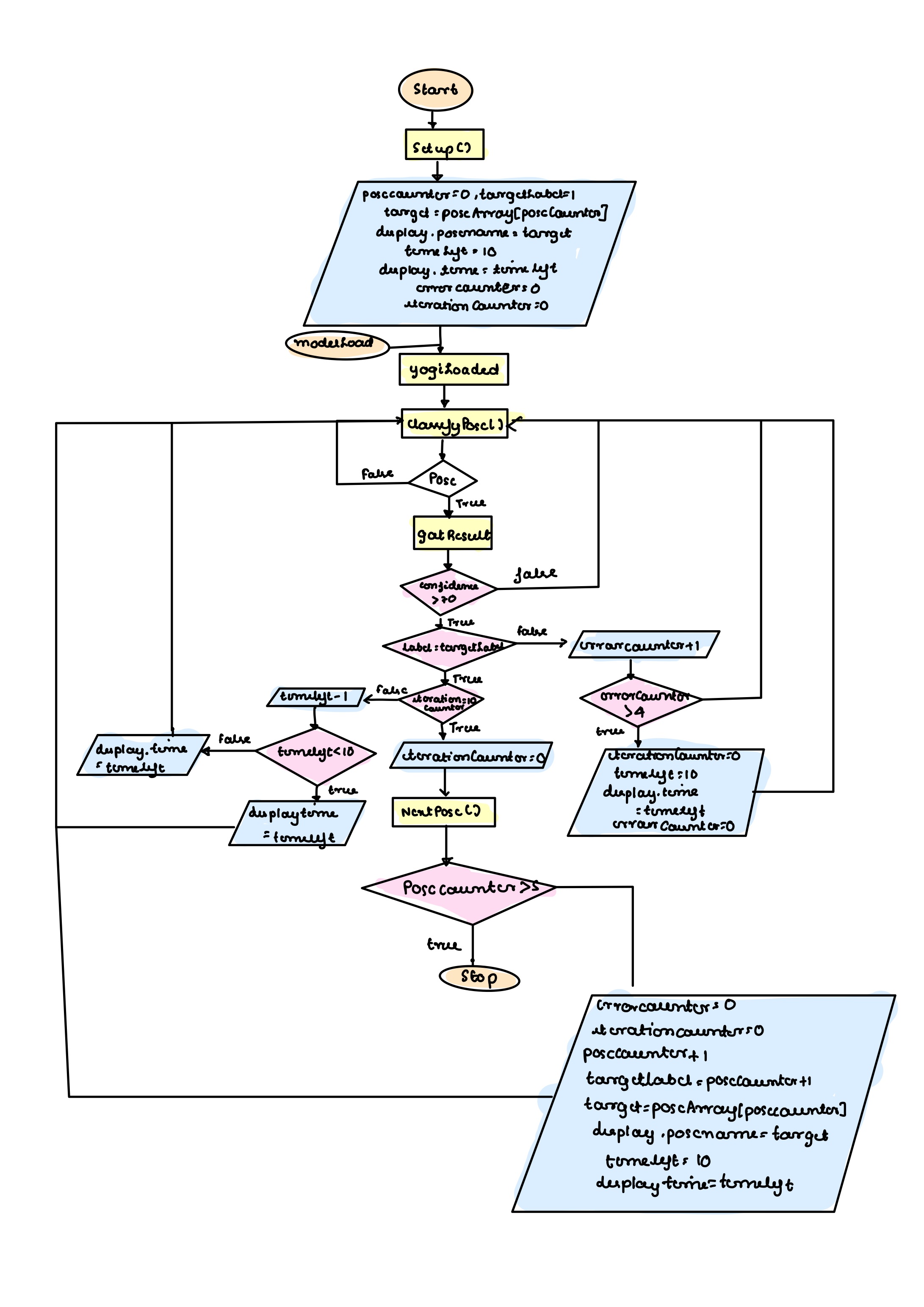


Fig 3.1: Program logic flow

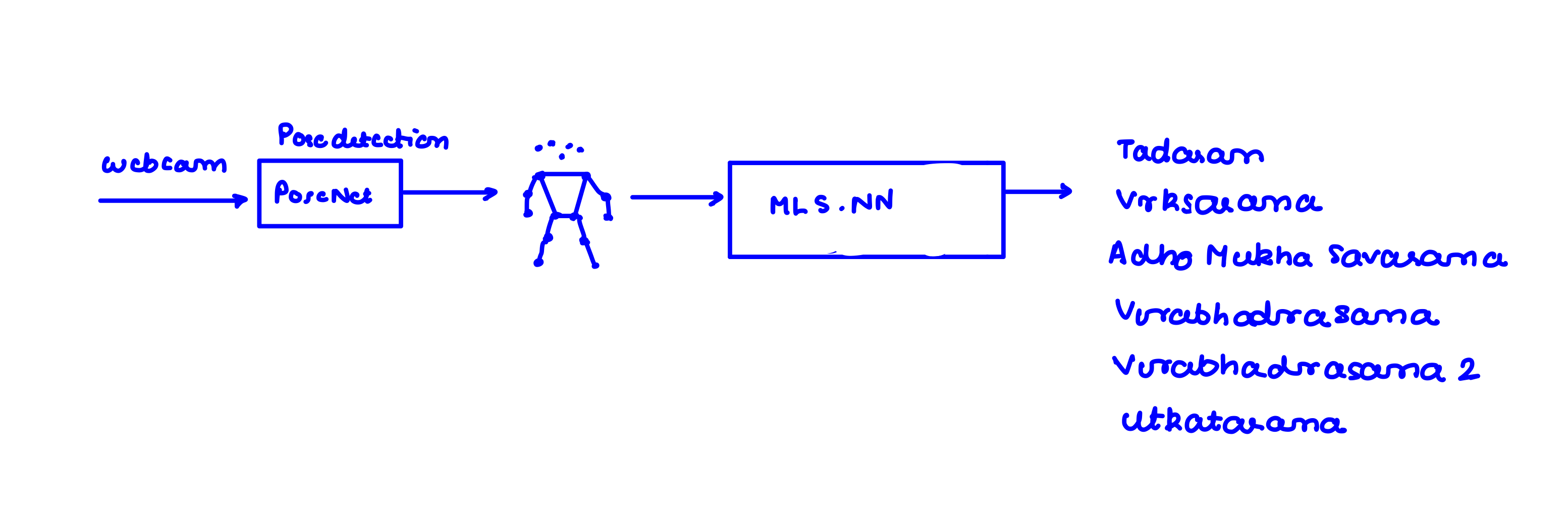


Fig 3.2: Flow diagram of project

*School of Computer Engineering, KIIT, BBSR*

*YOGA TRAINER USING POSENET*

Chapter 4

Implementation

4.1 Methodology OR Proposal

The 3 basic steps which can generalize any machine learning model are as follows: Model building, model training, model testing. PoseNet provides us with all these options but the model training here is more of a interactive one than writing codes. For model training we have to teach that some pose means something and if you are able to detect that pose than classify accordingly. As the visitors could be from novice to trained professionals the accuracy of our model depends how well are the poses are performed during training, it’s the special tool which will help us later. In the model building part, we are going to employ the capabilities that KNN (K-Nearest Neighbor) provides to us and classify 5 yoga poses. These will be ‘TADASAN’ (Mountain Pose), ‘VRKSASANA’ (Tree Pose), ‘ADHO MUKHA SHAVASANA’ (Downward dog pose), ‘VIRABHADRASANA’ (Warrior pose), ‘VIRABHADRASANA 2’ (Warrior pose 2), ‘UTKATASANA’ (Chair Pose).

Methodology and proposal

* + 1. P5.js

It’s an open-source JavaScript library, which enable even beginner to do creative coding and democratize the whole process to anyone with the basic knowledge of coding, it may not necessarily be the language JavaScript, and programming language will suffice. It gives the programmer many capabilities associated with drawing functionalities. Although it has many functionalities, we are going to discuss only those which are to use of our program. Starting with the most basic function from the where our programming begins.

The ‘setup()’ function

Even though the code starts here its main functionality is to define canvas or restrict the boundary that they want do the work on. As a website has multiple division and multiple moving elements so it’s a good practice to have a pre decided canvas. This function works once when the program is starting. In this project we are going to do a few things in this part of our programming.

1. Canvas creation: to do that we have ‘createCanvas()’, like any function it accepts some argument and this function uses that capability to define the pixels for which the canvas is going to span. E.g., I want a canvas 640\*480 pixels, then while calling the function I should pass the argument as ‘createCanvas(640,480)’. As our main function will be to open webcam feed here (off course except it we have other element on our website) so we are storing it in and variable called canvas. Some of the necessary condition for this function is that it should be defined in setup only and once, otherwise it will react unintentionally.
2. Positioning: erstwhile mentioned that, we have a particular architecture for ourselves designed and so we need the webcam feed to open at a particular location. For that purpose alone, we are going to format it using a function called ‘position()’. Similar to pixel distribution of above the webpage we are going to load has pixel position dedicated to it, which helps us positioning elements on it, so by sending that position we can place our canvas there, e.g., writing (130,210) as argument will tell the webpage to keep this element at 130 pixel and 210 pixel away and below the northwestern tip of the browser respectively.
3. Opening Webcam: with this step we finally enter the part of our project where something of substantial or in the visible spectrum happens. As p5 provided us with the tool to create a canvas it also empowers us with opening webcam for our real time data capturing. ‘createCapture(VIDEO)’ is the function which open the webcam, you do have to allow the permission for opening the webcam from your end. Here lies a problem the feed comes in 2 part, so not let that happen one can use the function ‘hide()’. For that purpose only, we are declaring a variable video on which we are going to take the feed.
4. Opening the PoseNet Model: this and model loading and initialization we will discuss in further sections.

‘draw()’ function

This function can be defined as the butter on the bread that is our setup function, if setup function define canvas this enables us with capabilities of drawing on this canvas. This function runs in a loop, so whatever you write here it keeps on repeating itself. Let’s quicky mentions some capabilities: ‘background()’ this function is used to color the canvas we created in setup function. If not provided with any argument it is set to transparent by default. This function isn’t unique to draw, it can be defined in it to either clear display window at each frame beginning, or in setup if it has to set once. It follows the same coloring sequence as the HTML STYLE SHEET initiated using grayscale integer value, RGB etc. draw function includes some predefined shape function which are – ‘ellipse(x,y,z,a)’ where x is the pixel left from northwestern tip of browser, y is the pixel down from northwestern tip of browser, z is the width of the elliptic structure and a is the height of the erstwhile mentioned elliptic structure. There few other arguments to but these are the basic to get ellipse, the 2d diagram will appear as the mentioned argument with black boundary in the backdrop of white background. You could further experiment with this functionality. One of the fun parts of this is create a snake, by simply replacing x and y with mouseX and mouseY respectively, which create sphere on the canvas as the mouse moves, the faster the movement of the mouse the more sparsely the ellipse appear and vice versa. Mention may be made to translate function which shift canvas according to given argument, which is x and y where x will shift the canvas x units horizontally and y will shift the canvas y units horizontally here we have shift the canvas on the basis of x=video.width and y=0. Now coming to scale function, usually canvas works in 4th quadrants meaning x is painted left to right and y is painted top to bottom, but by giving the argument -1,1 it will be in 3rd quadrants meaning the x pixel go right to left and y pixel goes top to bottom. This will help reverse the webcam image and have an accurate poses detection. We will return to draw function once we discuss ml5 library and will leverage its services to create the skeleton.

* + 1. ML5.js

The ml5 project was started with the goal of making machine learning more accessible to all in language like JavaScript, where the user can easily use them in browser over the TensorFlow platform. This tendency of ml5 to work on browser user with added benefits, the real-time data that this library takes directly from the webcam of user, stays within the user browser and no data is collected. This gives it a layer of privacy that makes any beginner at ease knowing that the chances of data breach are negligible. According to the folks at ml5, tensorflow.js engages the capabilities of built-in graphics processing unit (GPU), which is inbuilt in every we browser to do calculation faster than any central processing unit (CPU) only dreams off. Using this calculation accessibility provided by the browser’s GPU ml5 brings the machine learning to life in any browser supported by it. It makes the application par of tensorflow.js much easier and more refined. Here are the following capabilities it provided in our program.

1. Loading PoseNet: for the purpose of loading the PoseNet we have created a variable of the same name using ml5 inbuilt function the ‘poseNet’ the PoseNet is employed in our model. As it might be clear how a website to run needs 3 component JavaScript, html, stylesheet and knowing how things are working is hard to predict, but this is were some good folks from browser we use provide us with console, through which we can keep track of the execution of how the program is actually running. We know the whole project is dependent on the PoseNet model if it fails to load correctly we will be stuck in a limbo. So we have to be aware of this situation before it arises that’s why we are going to use a function called ‘modelLoaded’ which will tell our console whether the PoseNet model is ready for use or not and act as a callback function. Hence, we are passing two arguments in our function, first is the ‘video’ variable which contains the webcam feed and second the ‘modelLoaded’ function itself. This later function will include and echo statement in form of console.log which will tell us the PoseNet model is ready for use, and we can go ahead further.
2. On function: this method is an event-listener, which is a basic JavaScript functionality as the JavaScript provide more of a interaction between user and website, so these are there to react and wait until the user in some way shape or form has interacted with the website, these can range from keyboard click, mouse movement or in our case webcam feed. It will listen for poses which can be single or multiple. Like the previous ml5 function this one also has two arguments, a string ‘pose’ and callback function in our case it’s going to be ‘gotPoses’. This function receives an argument called poses, while debugging a console in the web browser it will show the erstwhile mentioned array of poses. This array contains two property a pose property and skeleton property. Further taking a deep dive into pose reveals that there a lot of data, such as confidence score of the entire pose, 17 key points which include all the points mention in table 4.1 (each of these 17 key points contains these 3 property: x co-ordinate (ordinate), y co-ordinate (abscissa), confidence score of that key point).

|  |  |
| --- | --- |
| Left Eye | Right Eye |
| Left Ear | Right Ear |
| Left Shoulder | Right Shoulder |
| Left Elbow | Right Elbow |
| Left Wrist | Right Wrist |
| Left Hip | Right Hip |
| Left Knee | Right Knee |
| Left Ankle | Right Ankle |
| Nose |  |

Table 4.1: Key points detected by PoseNet

Coming to back to our callback function as we are dealing with single pose, we will store the pose and skeleton property in a variable.

As we were able to gather basic information about the poses from the PoseNet, we will try to represent this data in our webpage and make it more interactive by drawing a virtual skeleton on our user webcam feed, so that can have more fun while performing the poses. In this quest of our an erstwhile mentioned function will help us, that function is ‘draw()’, we know that draw function keeps running in a loop so it will give up the advantage of continuously showing the user his skeleton. For this purpose, we run a loop continuously for the length of skeleton function keep getting two points and joining them with a line if the pose variable has some value.

1. PoseNet with its capability provide us with a location of all these points required to build a skeleton, but as our project requires the next step would be to use this result and create a model which will be trained on poses and able to recognize and classify these poses. Ml5 provide this capability and make it much easier. So, the output of PoseNet model will be used as input for the ml5 neural network creating a pipeline and this neural network will identify the poses. PoseNet output which will act as our input are 17 key points. Each with a x and y co-ordinate making the input count as 34, whereas the output is going to be 6 (‘TADASAN’ (Mountain Pose), ‘VRKSASANA’ (Tree Pose), ‘ADHO MUKHA SHAVASANA’ (Downward dog pose), ‘VIRABHADRASANA’ (Warrior pose), ‘VIRABHADRASANA 2’ (Warrior pose 2), ‘UTKATASANA’ (Chair Pose)). We will keep these as argument while creating neural network. Even though ML5 also provide the capability of regression, our task is to create a classifier, so the input for the argument task will be ‘classification’. We are training the model so debugging is important to us for keeping tracks, so argument debug will be true. Once done with the argument we are going to use the function ‘neuralNetwork()’ in built in ml5. Then in the gotPoses function where we are making the skeleton, we use an array to store the poses (setting a time out will help the code to determine when and for how long to collect the data.), then use these arrays and add these data to neural network. ‘addData’ function will help us in this endeavor. ‘saveData’ will help us to store this data into a ‘json’ file which can be later retrieved in other programs using ‘loadData’ function, after which comes the training part which will be done using ‘train’ function and we can save it again using save() giving us 3 file 2 in json and 1 bin making and preparing our model. Mention may be made that a target variable must be provided to each of these poses.
   * 1. Google Teachable Machine

Google which has provided so many tools to make the machine learning journey easier for JavaScript beginners, also provide another service called ‘[Teachable Machines](https://teachablemachine.withgoogle.com/train/pose)’ it’s a website which enable novice to create ML model without even writing a code. It can recognize images, poses and even sounds. One can easily go to website and choose how many classes he or she required, it requires webcam permission in order to capture the feed and you can train you classes on as many shots as possible. Once done with the training, it also gives you a window to ensure that your model is trained or not. Finally, you can download this trained model in form ‘json’ file. Saving the hassle of writing another program specially to create and save the model.

* + 1. DOM

Document Object Model. So, far we have cover multiple time that a webpage works by the harmoniously coming together of html, JavaScript and to some extent stylesheet (CSS). Html defines what is going to be on the web page. Stylesheet decide how they are going to be arranged on the website. And finally, JavaScript decide how these elements will react with each other and the user. So, we need to link all 3 of these documents together. What is DOM? It’s the JavaScript representation of a webpage or bunch of objects that you can interact with JavaScript. It provides for a way to the JavaScript to tinker with the html object. It represents all the element of html in a tree format where the root element is <html> tag while other object appears as it child nodes. As erstwhile mentioned, JavaScript can tinker with it, change HTML elements, attributes, CSS styles, remove existing HTML element and attributes, react to these, create new. While using the console you can get these DOM tags, it may not be its corresponding HTML elements, but will be what the browser is using to denote these elements. To access these tags we can use ‘getElementById()’ function, the tag you want to find is used as argument, and it’s up to the developer how he interact with these. In our case we are going to use this, to update yoga poses, timer and etc.

4.2 Testing OR Verification Plan

This project is going to be user interactive so that the user can interact with the browser, the main target democratic of it, belongs to people who want to learn yoga with correct posture. The interactive element of the website will include webcam footage (on which their artificial skeleton will be shown), name of the yoga pose being performed, an image showcasing the pose and of course a timer to keep track. Out of all these things, timer will be of ten seconds, and will be updated each second. Once the timer hits zero, another 2 elements will come forward congratulating the user on completion of pose and congratulating image (it will be a bock level element). These will appear for a brief second and will vanish along with updating the yoga pose name and image showcasing the pose. This will continue until all the 6 poses are done. After which another congratulating text for completion of all the poses will be shown.

For the purpose of using the image we will create a image array, as this will be done once we can declare it at the start of the code, which will be setup. Next would be task of initializing some of the variable required in our program as mentioned in table 4.2

|  |  |
| --- | --- |
| poseCounter | Initially initialized to 0. Will be used as iterator. |
| targeLabel | Help us in identifying the pose when we traverse through the pre trained model ‘json’ file. |
| poseArray | Array which will be used to store pose name, here we will use poseCounter as iterator. |
| Target | We will use this variable, to update pose name on the browser display and will help us traverse poseArray. |
| timeLeft | Our time counter, we will update it to ten once at initialization and after pose updating. |
| errorCounter | The training logic works in a flow if user start doing any other thing this will time out as errorCounter reaches 4 |
| iterationCounter | This will make sure the exactly 10 sec (10 times to be more precise) the pose has been estimated correctly |
|  |  |
|  |  |

Table 4.2 variable list

The model which downloaded either by self-training or through google teachable machine is downloaded as 3 files, model.json, model\_meta.json and model\_weight.bin. These are files need to fused in the main JavaScript file in order to feed them the PoseNet info get the classification done. As done in python ML project we create a model object to do the further processing here to we’ll have something like that. ‘yogi’ will be the object and it will be trained through the function ‘neuralNetwork’ function. It has multiple arguments but as we have already chosen some arguments during training we are going to continue with them these are: input = 34 (x and y position of the 17 key points estimated by the PoseNet), output = 6 (Our 6 yoga poses ‘TADASAN’ (Mountain Pose), ‘VRKSASANA’ (Tree Pose), ‘ADHO MUKHA SHAVASANA’ (Downward dog pose), ‘VIRABHADRASANA’ (Warrior pose), ‘VIRABHADRASANA 2’ (Warrior pose 2), ‘UTKATASANA’ (Chair Pose)), task = ‘classification’, debugging = true. Once our object is ready the next task its supposed to do is training here comes the saved files we were talking about earlier. To employ it in our project, we are going to create a JavaScript dictionary whose element will be key will be model, metadata, weights and value pairs will be the erstwhile mentioned file location. We will use this dictionary as an argument to another function called ‘load’ to load our model into yogi, which will act as place holder. Another argument to this function will be a callback function called ‘yogiLoaded’ to signify us in the console that the model was successfully loaded. Only when we ensure this fact, we can be sure that our code will not behave un-naturally and move on to actual classification of current incoming real-time data.

classifyPose()

This function comes into play after we have loaded the pre-trained model and marks the beginning of testing of phase ML cycle. Here we will create input array to store the current array coming output from the PoseNet model which is taking the current running webcam feed as input. Once we have this array we can call ‘classify’ function this is the function that uses the PoseNet output, saved poses, loaded classification model to determine the pose being currently made by the user. The input array we just made will be used as one argument other will be the gotResult function. classifyPose will be our looping point in this project as and when any error occurs, we will wait for a few second loop back to it, this can be achieve using the function ‘setTimeout’.

gotResult()

This function is where the main logic of program will be done it’s the most important function on how the user will be trained on different poses. It receives two argument one is called result and other is called the error. There is a confidence score attached to the result, which leads us to dilemma. That is what should be the confidence score we should decide as the user is doing the correct posture. 80% is a pretty hard number as the person training the model may have a completely different build from the one performing it, and hence hard to achieve. This is more of a trade off than anything, after discussion we decided to settle on 70%. Next step would be to make sure that the estimated pose is the pose which is currently in our pipeline that is the targetLabel, if yes, we can decrease the time. As the time for each pose is allocated as 10 sec, when the times reaches this golden second, we can reset the timer and move to the next pose. But everything is not hunky dory and we could encounter some error along the way, to deal a situation like when user does one pose for 4 sec and changes it half way. Situation like this need a error handling condition so that we don’t make a bad rep for our AI, for this purpose alone we will measure the error using an iterator which will keep track if the current pose matches the pose in the pipeline. If a situation arises where this isn’t followed, we will timeout. Now reach the last part which isn’t as big of deal but need a mention, when the confidence score is less than out expected estimate that is 70% than the only option for us is take a timeout and keep the loop running. This function as mentioned earlier is more of the logical once or you could this function spin the thread of our project.

nextPoses()

This our last function although not playing a bigger role in the backend it’s going to be playing a huge role in what JavaScript is supposed to, make front-end experience interactive. We arrive here once the timer hits 0, that leads out to path, either we completed a single pose or all the 6 poses have been done successfully. In the case of later there’s isn’t much to do we just need to congratulate our user, for taking their precious time and using our website. But in the case of latter, we need to change of a lot of things that appear on the screen. These include, changing the target label, increase the count of poses, replacing the name of pose on the web browser API, reset the clock, and change the image used to show the instruction meant to teach the user. Of course to loop back to our classifier class so that a new pose a new beginning and a new epoch of the loop can start.

4.3 Result Analysis OR Screenshots

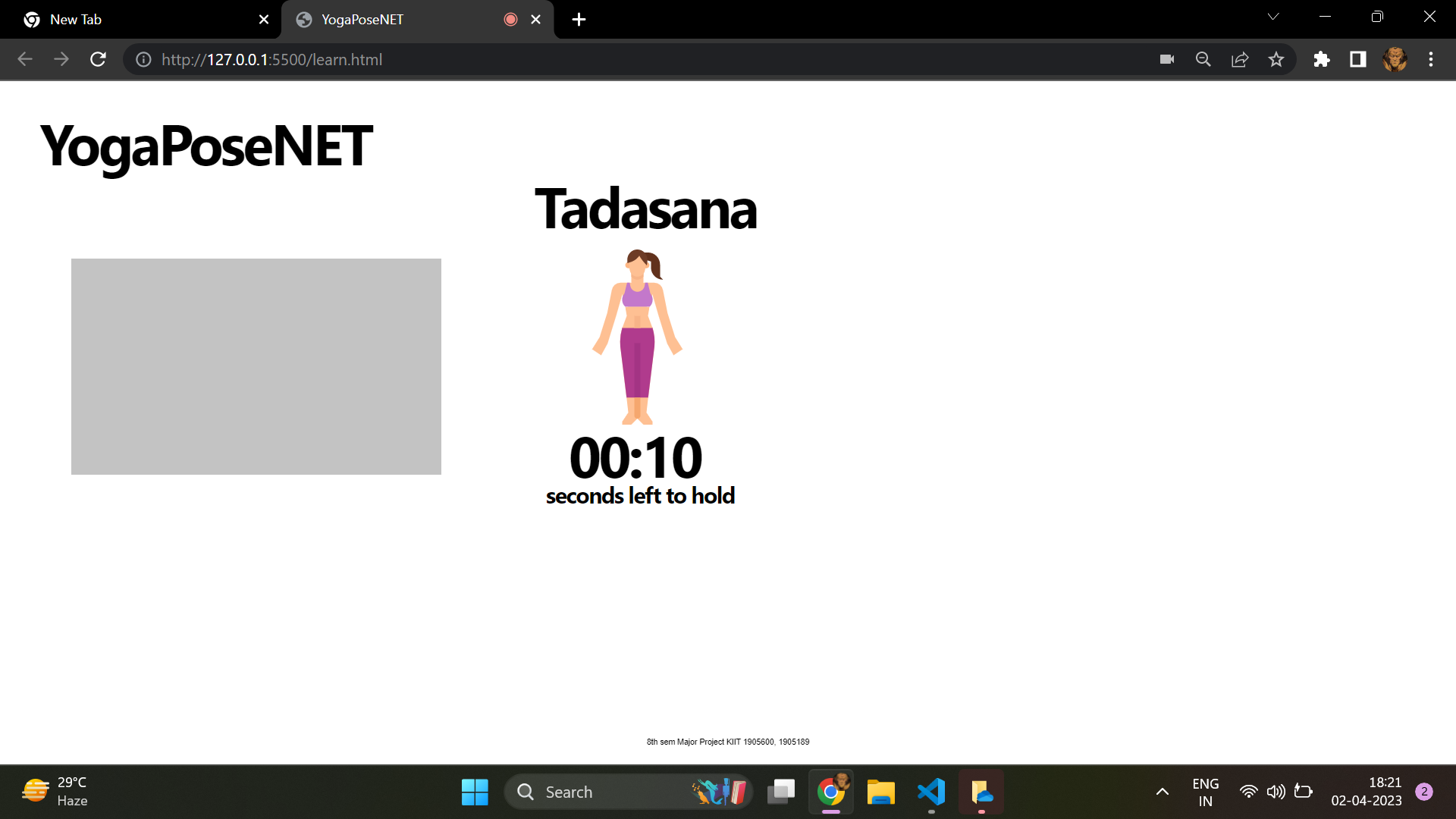


FIG 4.1: Entry Screen

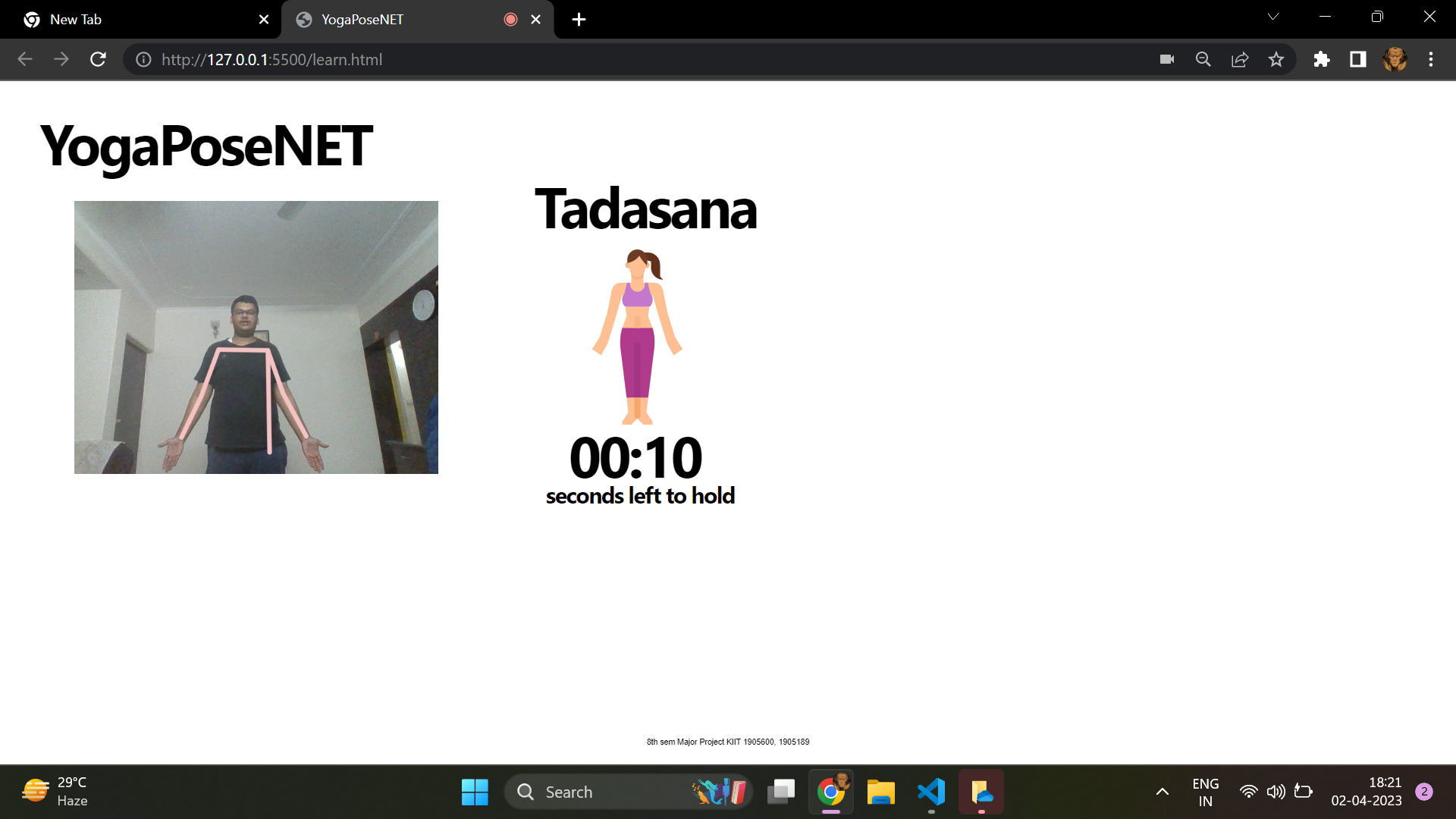


Fig 4.2: Tadasana

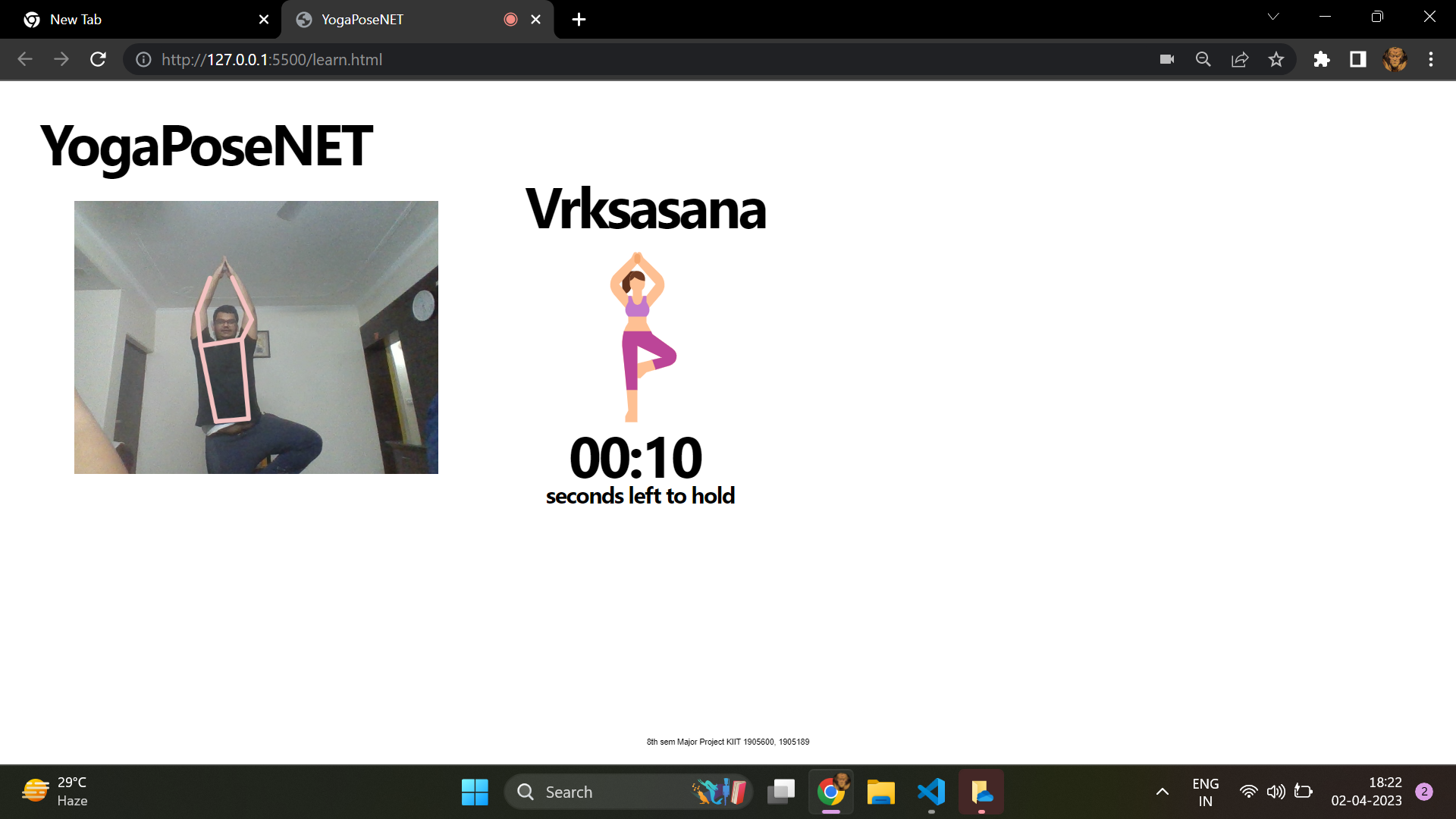


Fig 4.3: Vrksasana

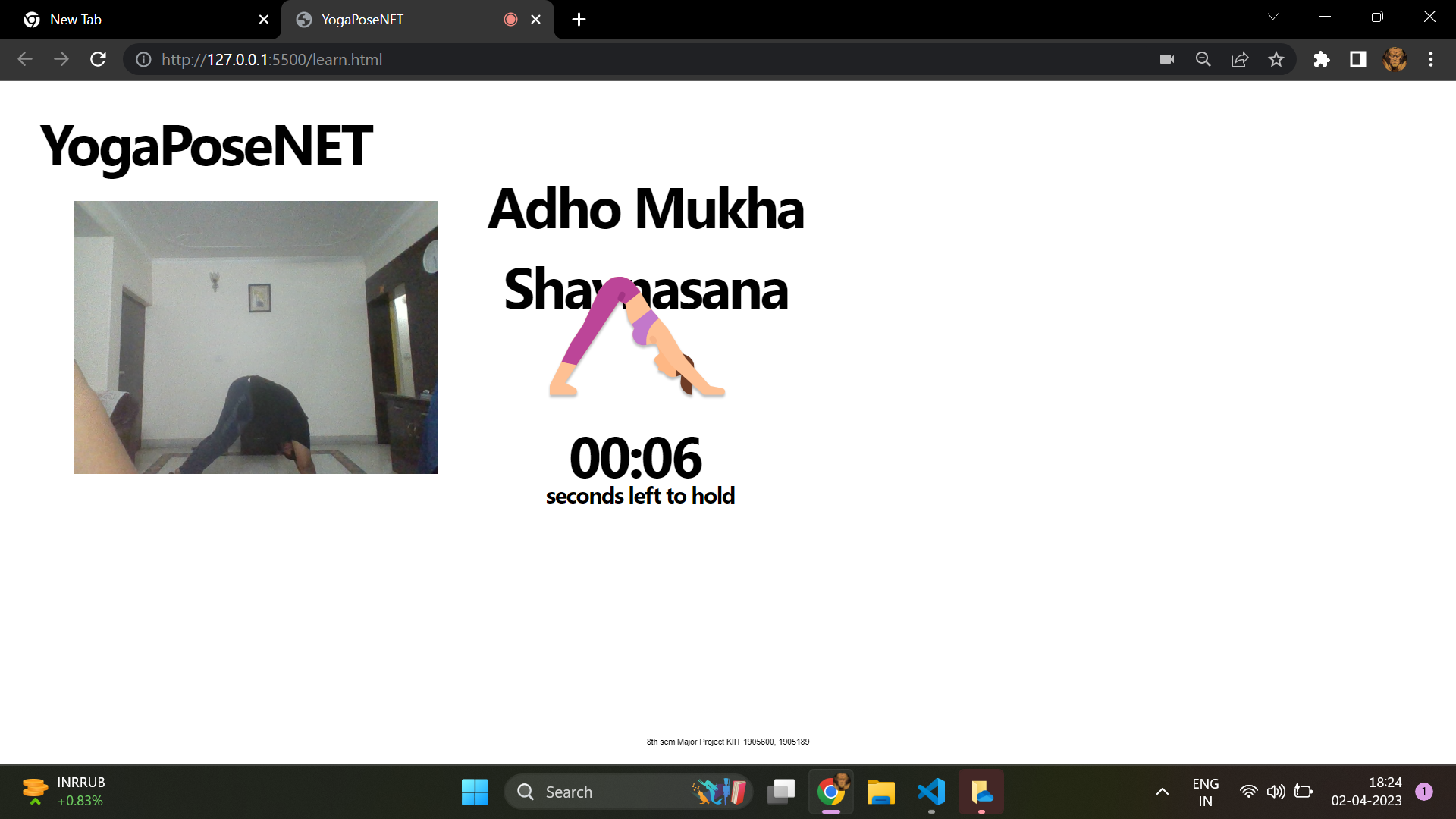


Fig 4,4: Adho Mukha Shavasana

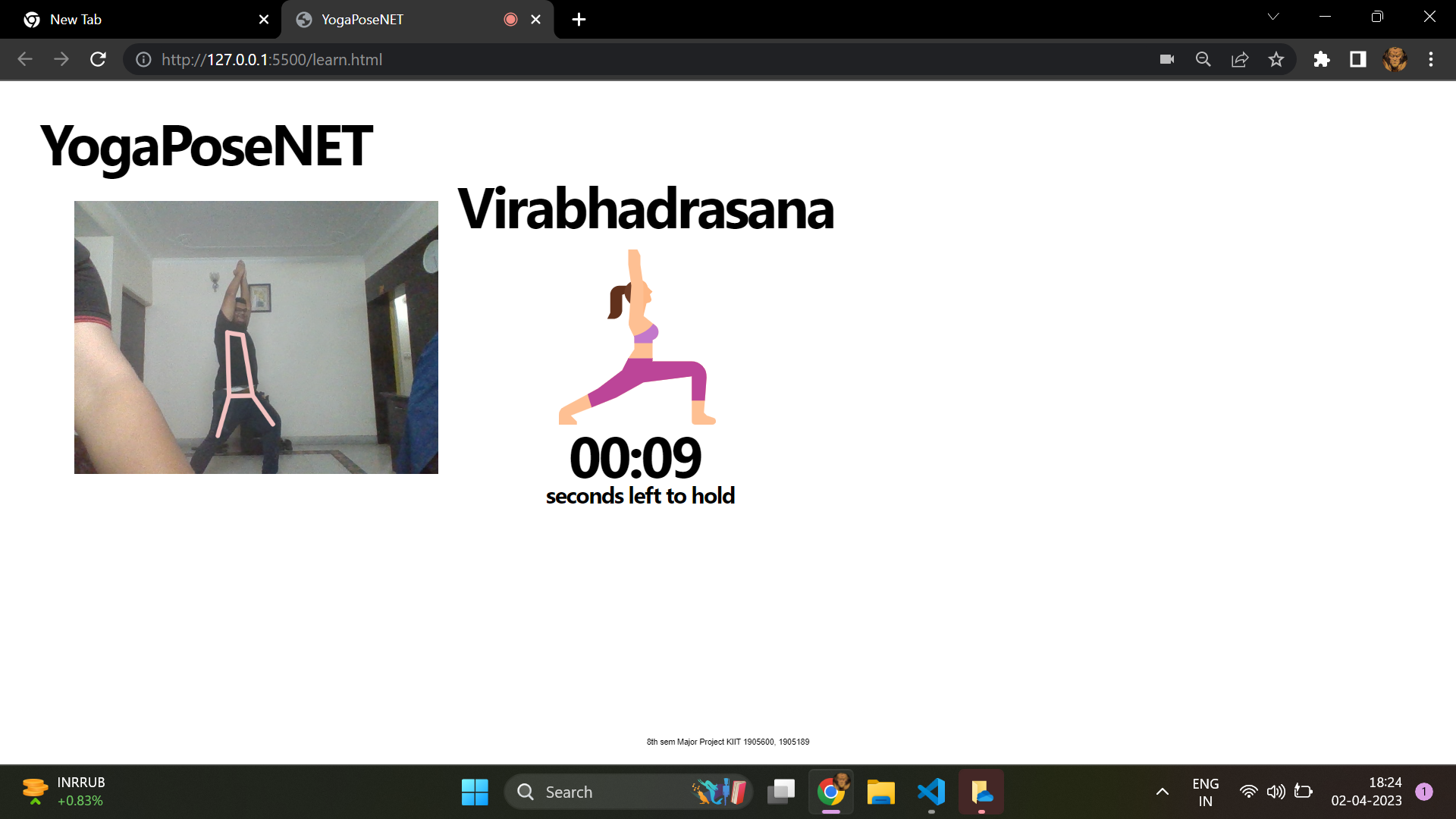


Fig 4.5: Virabhadrasana

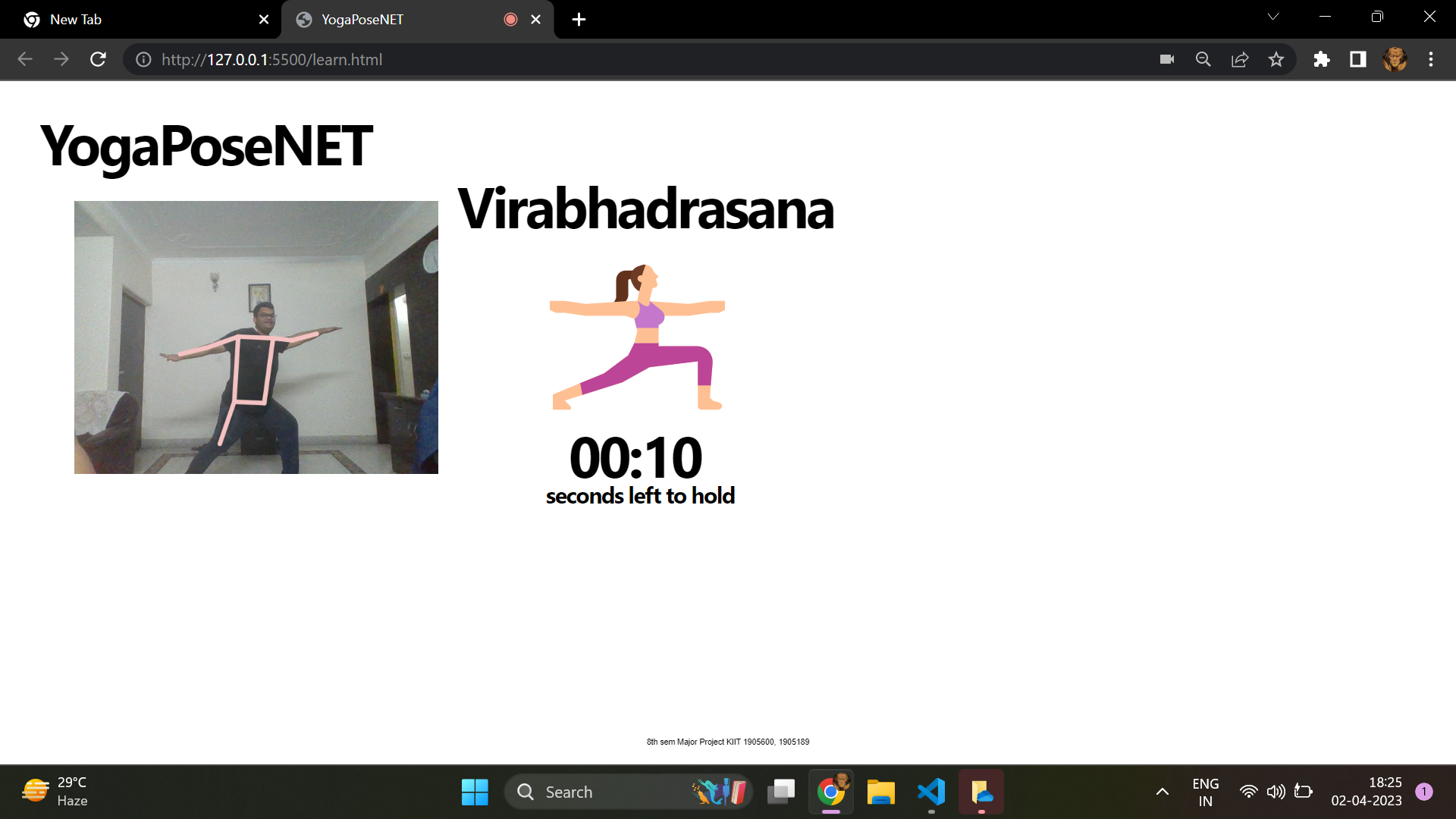


Fig 4.6: Virabhadrasana 2

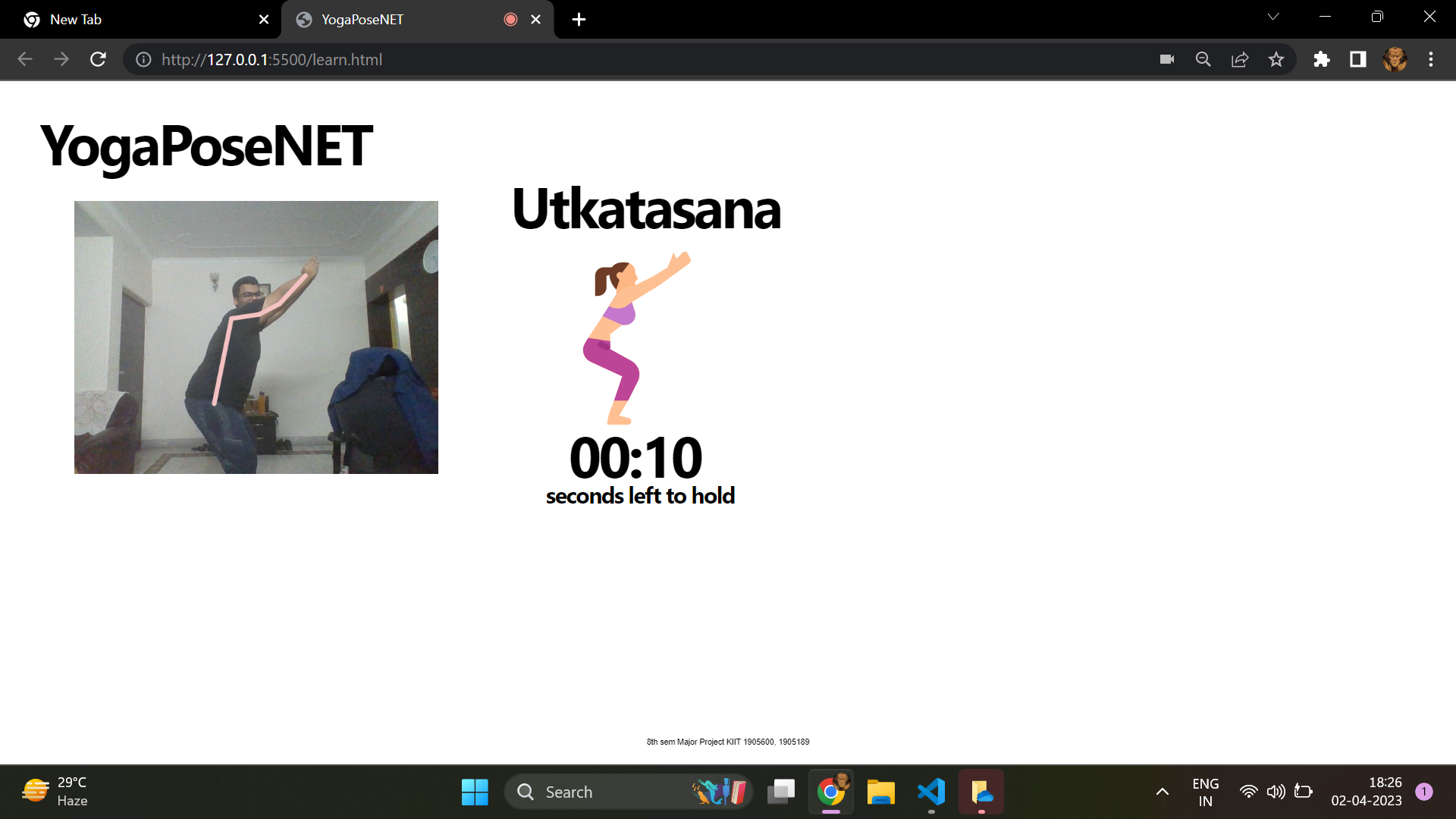


Fig 4.7: Utkatasana

*YOGA TRAINER USING POSENET*

Chapter 5

Standards Adopted

5.1 Design Standards

Our project ‘Yoga Trainer using PoseNet’ developed with the aim of making yoga more accessible, people doing it more flexible and providing them maximum benefit they can reap from it. It also had the undertones of making machine learning more accessible to novice. It involved building a machine learning neural network with the capability of classifying, for this we trained the model and tested it on real time data. The basic schema of this project is as follow:

1. The complete project is web based only consisting of front-end work.
2. The whole coding for this program was done in Microsoft’s Visual Studio Code with web development framework such as HTML, stylesheet and JavaScript
3. The project started with creating the basic HTML file which included a division that was supposed to be canvas in the next stage, Name of the pose to done along the instruction to be followed along with a timer to clock in the pose performance period.
4. Next phase was to arrange these elements in some what presentable fashion which was done through a HTML Stylesheet. Which created 2 division one for the canvas area another for instruction, clock, yoga pose and its name
5. Important libraries such as ML5, P5 which we were necessary for successful implementation of this project was added to the html source.
6. Late a JavaScript file to decide how these things react with each other was created and linked to the html file.
7. Basic function required to run PoseNet were done.
8. Using google Teachable Machine the ml5 neural network was trained.
9. With the output produced by the erstwhile Neural Network was fed to the already trained model to classify the poses as yoga poses.
10. A logic was decoded on how to train user with different poses, which took care of all the thing that the website was required to do.
11. All these steps were performed using Microsoft’s Visual Studios code’s extension called Live Server
12. Finally, once satisfied with code, it was hosted on GitHub and deployed.

5.2 Coding Standards

Creating a website from scratch is difficult but implementing it to more understandable, readable is in league of itself. For this only and future update, fixing error, extending the project it is recommended to follow certain guideline to create the project, making it easy in the future and helping anyone looking the project even for the understanding purpose. In this following project we have worked using [Sun document code convention](https://gist.github.com/antonrogov/1216380/21800f463af3d3a98e98083c4bc109e44f981ef4) for java programming language, there have been some liberty as the exact coding convention of java cannot be applied to JavaScript.

1. JavaScript file should be created with the extension called ‘.js’. It should be session specific only then we should link it to HTML file. It also takes away the opportunity of mitigation. Hence the statement src=’filename.js’ should be added to the HTML file at last.
2. The maximum tag used should be 4, as use of tab after that will make the program unnecessarily lengthy.
3. It is recommended the ideal length size of any program should by 80 characters after which you should split the line along a operator preferably a coma. After this break happened you should make sure that the next line starts with a gap of 8 spaces.
4. Leave as much comments as you can as these will provide your window in to the mind of the programmer, and even help you get back in the mood that you were while coding it so that making connection becomes easier, you could be funny while writing this as humor co-related to good memories which can be easily accessible in the time of need.
5. Even though the strives made by JavaScript allows it to work without declaring a variable, but this practice should never be followed you must always declare you variable first, to avoid a scenario of implied global.
6. Similarly, every function should be declared before used. Another nuanced rule that function should follow is that there should be no space between the function’s name and its argument but there should be gap between the first and second argument.
7. Naming conventions: the name should only contain either an alphabet or underscore. Its recommended no to use special character that may lead conflict of understanding in the international forums. Never start a name with underscore. Function name should start with a small case letter. You can use camelCase in second prefix of the function name.
8. One line one statement: each single line should contain one single statement ending with semicolon, except of course in case of conditional statement and function start.

5.3 Testing Standards

The project entirely works on training and testing, so the testing results are directly proportional to how much we train the model. The training of the model is done on ML5 neural network. Neural network works on the principle of ensemble learning and also widely known as the deep learning the coding community. The neural network creates a complex web of co-relation between different dependent variable, which are pitted against many layers contained in it. This entire process gives every entity some weight and calculated output, based on this output is backtrack all the path it followed and returns to weighing part. With the knowledge it had gained once going through the process is used to readjust the weight so that, it can achieve better result. It keeps doing this until we allow to find the best weights it can find in-order do either classification or prediction. Once this model is trained on already existent data, we pit it against some new form of data. With knowledge it acquired, it tries it best to use those convention and fit them on the incoming new data to find a result. In all these two situations arises. One is underfitting which means the model performed well on the training data but did very poorly on test data. And another condition is overfitting, in which data get trained on so much data that it starts considering noise as important leading to once again a failure. In our case the data used to train the neural network is obtained from another network which is PoseNet model. This model provides us with a confidence score that is used as our testing condition, if the confidence score is greater that 70% and our user is doing the pose that is the pipeline of our program, we call it a success. If not, we wait until the user does so.

*School of Computer Engineering, KIIT, BBSR*

*NAME OF PROJECT*

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

In this day and age, when the most of our time is spend in fronts of machine the need of the exercise cannot be over emphasized, but one can risk injuries without the lack of flexibility, that why yoga is so important as it makes one body strong and flexible enough to take any form of exercise. But with growing load of work, it is very difficult to find a trainer who can properly train yoga, and proper commercialization of this field is yet not achieved. So, an AI trained on yoga poses, was the best next thing one can hope for.

So, with the goal to create a ML project capable of training people how to do correct yoga in comfort of their vicinity. We were aided by stride made many of our senior working in this field, their aid in the form of TensorFlow, ML5, J5 and google teachable machine made it really easy to understand them and implement them according to our needs. We used the pre-trained PoseNet model in-order to draw the human skeleton so that we track that movement. We trained this capability to build our ML5 model which was trained on 6 yoga poses. Later we implemented a project that was capable of classifying these 6 poses. And using that we created a pipeline to train the user of our website the yoga poses.

6.2 Future Scope

We are just novice, who have basic level knowledge of web development, ml5, j5. The model prepared by us isn’t the best model in the world it is the best model we could create with limited knowledge we had at our disposal. And hence there is a lot scope for improvement. We employed the basic property of ML5, P5 model to create the project but, if it were in the hands of a veteran, they could have created a very interactive and responsive model. Not to mention they may have created the model from scratch manually training the machine. We were also inept in the web development cycle of the program. So, we created a very bland looking website, the website needs a lot of new features a constant updating to keep with the trend, new and improved ideas to make it more interactive. And of course, we are not an actually trained yogis who dedicated their live to this art form so the poses that were performed by us were also of beginner phase, if we were able to get some really well-trained yogi, our model could have better data to train upon and more poses added to the inventory. There a lot can be done with this field, this has gotten steam in the last 2-3 years yet the strive made here un precedented, using the knowledge that we have acquired we can create a model for anything and everything we want be it fitness trainer, martial art trainer, dance trainer. The future hold so much possibility.

*School of Computer Engineering, KIIT, BBSR*

*NAME OF PROJECT*

***References***

1. [*https://hcahealthcaretoday.com/2019/04/30/stress-the-health-epidemic-of-the-21st-century/*](https://hcahealthcaretoday.com/2019/04/30/stress-the-health-epidemic-of-the-21st-century/)
2. [*https://www.apa.org/news/press/releases/2000/12/anxiety#:~:text=In%20fact%2C%20the%20studies%20find,patients%20did%20during%20the%201950's*](https://www.apa.org/news/press/releases/2000/12/anxiety#:~:text=In%20fact%2C%20the%20studies%20find,patients%20did%20during%20the%201950's)*.*
3. [*https://blog.tensorflow.org/2018/05/real-time-human-pose-estimation-in.html*](https://blog.tensorflow.org/2018/05/real-time-human-pose-estimation-in.html)
4. [*https://p5js.org/*](https://p5js.org/)
5. [*https://www.w3schools.com/js/js\_htmldom.asp*](https://www.w3schools.com/js/js_htmldom.asp)
6. [*https://teachablemachine.withgoogle.com/*](https://teachablemachine.withgoogle.com/)
7. [*https://ml5js.org/*](https://ml5js.org/)
8. [*www.flaticon.com*](http://www.flaticon.com)
9. *https://www.youtube.com/playlist?list=PLRqwX-V7Uu6YPSwT06y\_AEYTqIwbeam3y*

*School of Computer Engineering, KIIT, BBSR*

*YOGA TRAINER USING POSENET*

**YOGA TRAINER USING POSENET**

ASHUTOSH MISHRA

1905600

**Abstract:** With life getting connected and busy people are becoming more frustrated and anxious. Just like earth is reaching its peak carrying capacity, human too are reaching the mental breakdown. In the world where things are so fast one need yoga that he can use as easy relaxer and could be done on the go. That where our project comes in providing them that necessary breather along with anytime anywhere access, so that even the novice can learn achieve mental piece.

**Individual contribution and findings:** Task given to me was mostly backend. I started with the most important part of the program which included JavaScript file. Next was to understand how PoseNet works, once familiar with vlog provided by the TensorFlow team, I visited ML5, PS5 official website through which I was able to understand how the basic functions like setup and draw supposed to be integrated in the program, which pushed us to the stage where we were able to see the video feed and able to detect a single person through the webcam. Next on my list was to create a logic that will govern the flow of my program and how user will react to the program, so I came up with a false timer which will decrement once user pose is detected to be correct. So once the pose is performed correct ten times the timer will hit 0 mark. In this case if the user starts performing another pose or it posture is wrong I have set up confidentiality score. If the score is above 70% the posture is assumed to be correct else, wrong and that will mark one fault on error counter. Once this counter reaches the value 4 after a time out and timer will return to 10. Once the user has performed all the poses correctly the timer will congratulate him marking the end of the project.

**Individual contribution to project report preparation:** I wrote chapter 2, 3, 4 and any unquoted diagram is prepared by me.

**Individual contribution for project presentation and demonstration:** I prepared the slides of back end which including things that are working in the background of our project like, PS5, ML5, PoseNet.

Full Signature of Supervisor: Full signature of the student:

……………………………. ……………………………..

*School of Computer Engineering, KIIT, BBSR*

*YOGA TRAINER USING POSENET*

**YOGA TRAINER USING POSENET**

PRATYUSH AANAND

1905189

**Abstract:** With life getting connected and busy people are becoming more frustrated and anxious. Just like earth is reaching its peak carrying capacity, human too are reaching the mental breakdown. In the world where things are so fast one need yoga that he can use as easy relaxer and could be done on the go. That where our project comes in providing them that necessary breather along with anytime anywhere access, so that even the novice can learn achieve mental piece.

**Individual contribution and findings:** Following task were assigned to me. HTML: my first task was to prepare the canvas on which the whole program depended, and I wrote a very basic html program. Then updated it accordingly. I prepared an stylesheet to format who I wanted the website to look. I accordingly settled for a heading, a canvas which was supposed to show the webcam feed once the permission was given. And for instruction I showcase pose name and image showing the pose. Next, I was given the task of using google teachable machine to prepare a classification machine. The website is quite user friendly you can enter and choose however many classes you want to classify in my case it was the 5 yoga poses. Next would to train them, here after you have provided you webcam access you can train the poses by actually doing the pose. You can take however many images to train it keeping mind to overfit it. once done you can simply download the file it appears as 3 Json file and bin file.

**Individual contribution to project report preparation:** I wrote chapter 1, 5, 6 and abstract.

**Individual contribution for project presentation and demonstration:** I prepared the slides of front end which including things that are working related to html, CSS and google teachable machine.

Full Signature of Supervisor: Full signature of the student:

……………………………. ……………………………..

*School of Computer Engineering, KIIT, BBSR*

ORIGINALITY REPORT



5% 5% 1% 2%

SIMILARITY INDEX INTERNET SOURCES PUBLICATIONS STUDENT PAPERS



PRIMARY SOURCES



1. www.coursehero.com 3% Internet Source



1. Li Li, Tara Martin, Xu Xu. "A novel vision-based 1% real-time method for evaluating postural risk



factors associated with musculoskeletal

disorders", Applied Ergonomics, 2020

Publication



1. www.researchgate.net <1% Internet Source



1. mmhuman3d.readthedocs.io <1% Internet Source



1. www.worldleadershipacademy.live <1% Internet Source



1. blog.tensorflow.org <1% Internet Source



1. Submitted to Institute of Technology, Sligo <1% Student Paper



1. Submitted to Roosevelt High School <1% Student Paper



1. Submitted to Bournemouth & Poole College <1% of Further Education



Student Paper



1. ebuah.uah.es <1% Internet Source



1. Richard Joseph, Manoj Ayyappan, Tanvi <1% Shetty, Gurudatt Gaonkar, Aashish Nagpal.



"Chapter 24 BeFit—A Real-Time Workout

Analyzer", Springer Science and Business

Media LLC, 2022

Publication



1. dokumen.pub <1% Internet Source



1. ebin.pub <1% Internet Source



Exclude quotes Off Exclude matches Off Exclude bibliography On

GRADEMARK REPORT

|  |  |
| --- | --- |
| FINAL GRADE  /11 | GENERAL COMMENTS  **Instructor** |
| PAGE 1 |  |
| PAGE 2 |  |
| PAGE 3 |  |
| PAGE 4 |  |
| PAGE 5 |  |
| PAGE 6 |  |
| PAGE 7 |  |
| PAGE 8 |  |
| PAGE 9 |  |
| PAGE 10 |  |
| PAGE 11 |  |
| PAGE 12 |  |
| PAGE 13 |  |
| PAGE 14 |  |
| PAGE 15 |  |
| PAGE 16 |  |
| PAGE 17 |  |
| PAGE 18 |  |
| PAGE 19 |  |
| PAGE 20 |  |

PAGE 21 PAGE 22 PAGE 23 PAGE 24 PAGE 25 PAGE 26 PAGE 27 PAGE 28 PAGE 29 PAGE 30 PAGE 31 PAGE 32 PAGE 33 PAGE 34

