A Project Report

On

"FITNESS HARMONY SIMULATOR"

Submitted in partial fulfillment of the requirements for the award of the degree in

Bachelors of Technology

In

Department of Electronics and Computer Engineering

Submitted

By

PAIDIMARRI ABHIRAM	17671A1932
DASURI MEGHNA	17671A1915
SAINDLA RAJU	17671A1935

Under the Guidance of

DR. D. PAVAN DAS

Assistant Professor

Department of Electronics and Computer Engineering

J.B INSTITUTE OF ENGINEERING AND TECHNOLOGY

(UGC Autonomous), Yenkapally, Moinabad – 500075

2017-21

“FITNESS HARMONY SIMULATOR”

A Dissertation Submitted in the Partial Fulfillment of the Academic Requirements

For the Award of the Degree of

Bachelor of Technology

In

Electronics & Computer Engineering

PAIDIMARRI ABHIRAM	**17671A1932**
DASURI MEGHNA	**17671A1915**
SAINDLA RAJU	**17671A1935**

Under the Guidance of

DR. D. PAVAN DAS

ASSISTANT PROFESSOR

Department of Electronics and Computer Engineering

J.B. INSTITUTE OF ENGINEERING AND TECHNOLOGY

(UGC Autonomous & Accredited by NBA& NAAC, Approved by AICTE & Affiliated to JNTU, Hyderabad)

Yenkapally, Moinabad Mandal, R.R. Dist., Hyderabad-500075

2017-2021

J.B. INSTITUTE OF ENGINEERING AND TECHNOLOGY

(UGC Autonomous & Accredited by NBA&NAAC, Approved by AICTE & Affiliated to JNTU, Hyderabad)

CERTIFICATE

This is to certify that the dissertation work entitled **“FITNESS HARMONY SIMULATOR”** was carried out by **PAIDIMARRI ABHIRAM, DASURI MEGHNA, SAINDLA RAJU** bearing **17671A1932, 17671A1915, 17671A1935** in partial fulfillment of the requirements for the degree of Bachelor of Technology in Electronics and Computer Engineering of the J.B. Institute of Engineering and Technology, Hyderabad, during the academic year 2020-21, is a bonafide record of work carried out under our guidance and supervision.

The results embodied in this report have not been submitted to any other University or Institution for the award of any degree or diploma.

DR. D. PAVAN DAS
Assistant Professor
Internal guide

Mr. K. Roshan
Associate Professor
HOD-ECM

External Examiner

ACKNOWLEDGEMENT

There are many people who helped us directly and indirectly to complete our project successfully. We would like to take this opportunity to thank one and all.

First of all, I would like to express our deep gratitude towards our internal guide**, DR. D. PAVAN DAS,** Assistant Professor**,** Department of Electronics and Computer Engineering for his support in the completion of our dissertation. I wish to express our sincere thanks to **Mr. K. Roshan,** HOD, Department of Electronics and Computer Engineering and also to our Principal **Dr. S. SUDHAKARA REDDY** for providing the facilities to complete the dissertation. I would like to thank the college management for the support in the completion of our dissertation.

I would like to thank all our faculty and friends for their help and constructive criticism during the project period. Finally, I am very much indebted to our parents for their moral and financial support and encouragement to achieve goals.

PAIDIMARRI ABHIRAM
17671A1932
DASURI MEGHNA
17671A1915
SAINDLA RAJU
17671A1935

DECLARATION

We are PAIDIMARRI ABHIRAM, DASURI MEGHNA, SAINDALA RAJU bearing **17671A1932,17671A1915, 17671A1935** hereby declare that the mini project entitled **“FITNESS HARMONY SIMULATOR”** is submitted in the partial fulfillment of the requirements for the award of the degree of Bachelor of T

echnology in Electronics and Computer Engineering from J.B Institute of Engineering and Technology (UGC Autonomous). The results embodied in this project have not been submitted to any other university or Institution for the award of any degree or diploma.

PAIDIMARRI ABHIRAM
17671A1932
DASURI MEGHNA
17671A1915
SAINDLA RAJU
17671A1935

PLACE : Hyderabad
DATE :

ABSTRACT

Technological advances have transformed almost every industry, including the fitness world. Earlier, gyms and fitness clubs were the only option where people can find a personal trainer who can help them build their dream bodies. But, hiring a personal trainer is not something that everyone can afford. And that’s why we are designing a Fitness Harmony Simulator. By using this Fitness Harmony Simulator we can do workouts at home without hiring a personal trainer and it is cost efficient.

List of Figures

1: INTRODUCTION

1.1 Introduction

Health is wealth we all refer to this old saying to highlight the importance of health and fitness in our lives. But how many of us do actually have fitness routine? Digging deeper into the facts; approximately 3/4th of adults worldwide do not exercise at all. In fact, inadequate physical activity has been identified as one of the main risk factors of death worldwide over the past decade.

Be it the lack of time, premium gym or fitness club membership charges, or any other excuse, not paying attention to your fitness can lead to many health issues. But the arrival of artificial intelligence in the fitness industry is gradually make a change in people's lifestyles.

Yes, Ai is revolutionizing the fitness worlds by making fitness equipment smarter and encouraging home workouts. Besides the smart wearable (apple watch, fit-bit, and more) that can help you to track your fitness, Ai based personal trainers are also becoming popular at an unprecedented rate. Those who have been using Ai based personal trainer might be aware of this new concept in the fitness world, but people who are amateurs must be feeling curious about the same.

1.2 Problem Statement

Designing a prototype Fitness Harmony Simulator which will focus on continuously and accurately monitoring the state of the person in real time to check whether he is doing correct workouts or not.

1.3 Objectives

Fitness harmony simulator is a personal gym trainer. By using this software, we can avoid hiring personal trainer and spending more money to hire them.

- The primary goal is to initially plan a framework to detect persons by persistently checking the pose of the person.
- The framework works disregarding of different lighting conditions.
- Providing quality training.
- Hiring personal trainer can be avoided by using this software.

SYSTEM REQUIREMENTS

□ • **Hardware Requirements**: -

Laptop, Camera, Monitor, Alarm, Mouse, Keyboard.

□ • **Software Requirements**: **-**

OpenCV, mediapipe, pyttsx3, Jupiter, and Python

The flowchart of the proposed system has been shown in the above figure. The camera captures the image and sends to the processor of the laptop which consists of 32-bit memory card installed with Open CV which helps in image processing.

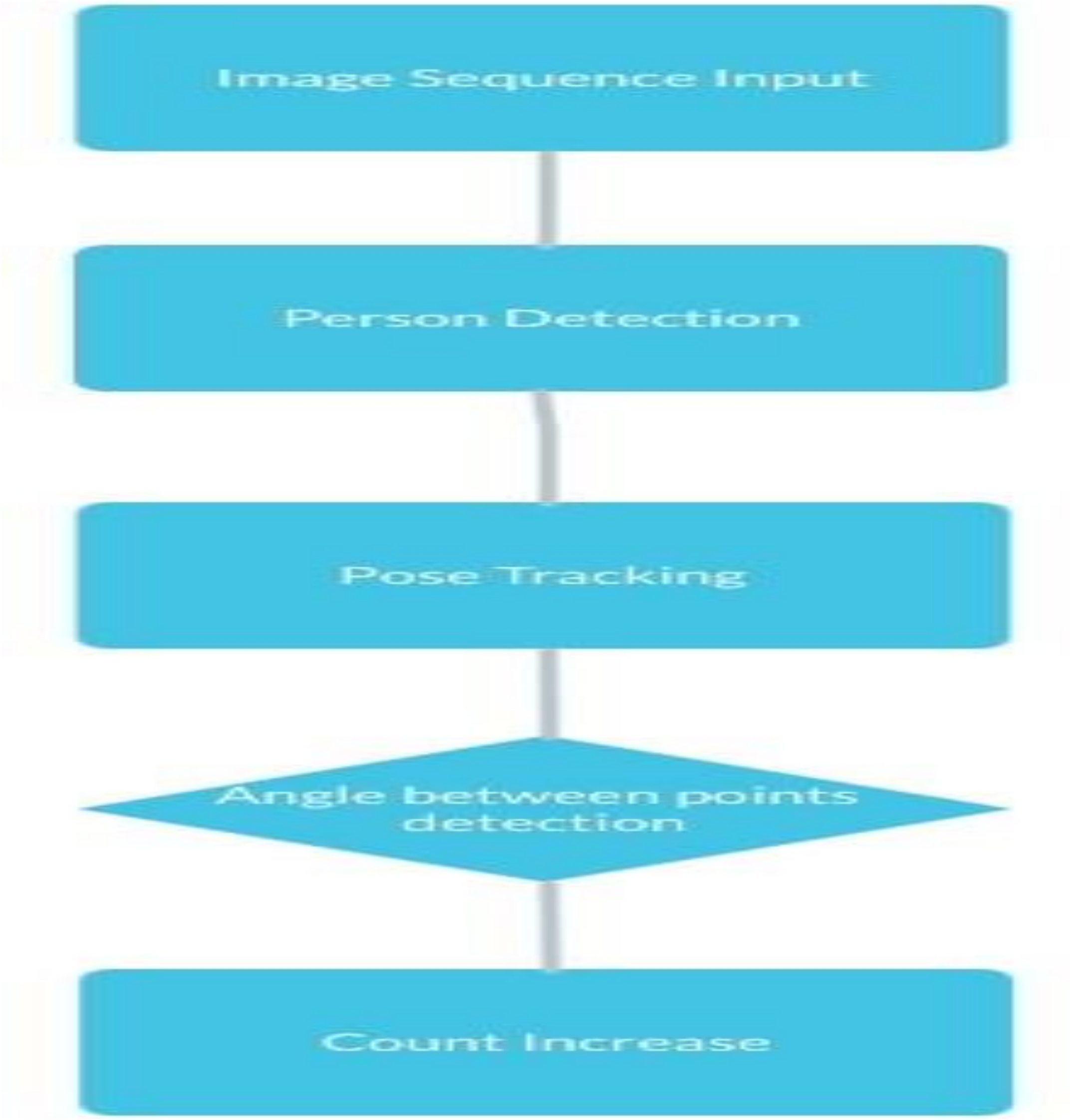


Fig1. Methodology

System Architecture of Fitness Harmony Simulator

If the signal crosses minimum and maximum threshold of a set of continuous frames with given minimum and maximum threshold value, it will automatically increase the count. Otherwise, the count will not get increases.

WORKING:

This project is mainly based on building an AI Trainer using OpenCV and Python. We will use the pose estimation running on the CPU to find the correct points and using these points we will get the desired angles. Then based on these angles we find many gestures including the number of biceps curls. We will write the code in a way that you will be able to find angles between any 3 points with just a single line of code.

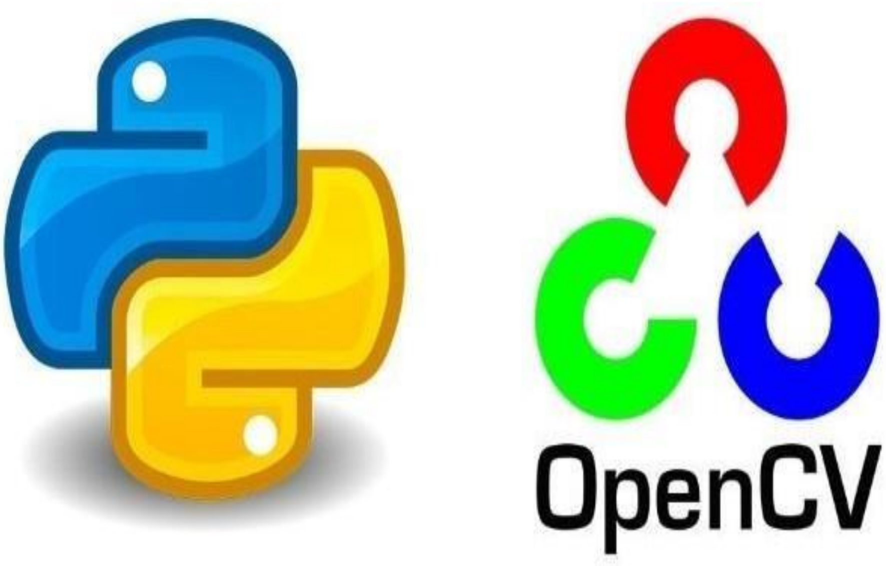
The computer will monitor a person’s workout to make sure all of their postures and angles are correct. Instead of being stuffed with hundreds of workouts, this trainer offers smart workouts to achieve fitness goals. Users have to enter a few details like body type, gender, current fitness level, future goals, and more to get a customized plan.

2: LITERARY SURVEY

2.1 OpenCV

OpenCV is an open-source computer vision library accessible in python coding language to code for visionary capabilities of our smart pc.

OpenCV was expected for computational capability and having a high focus on ongoing picture location and distinguishing proof. OpenCV is coded with streamlined C and can take work with multicore processors. If we need progressively programmed improvement utilizing Intel models [Intel], you can purchase Intel's Integrated Performance Primitives (IPP) libraries [IPP]. These comprise of low-level schedules in different algorithmic regions which are streamlined. OpenCV consequently utilizes the IPP library, at runtime if that library is introduced.



2.2 The Computer's Vision

PC's vision is the change of information from a still, or camcorder into either a depiction or another choice. Each and every such changes are performed to achieve a particular target. A Computer gains a cross section of numbers from a camera or from the circle, and it's just as simple as that. For the most part, there is no worked in example acknowledgment or programmed control of center and gap, no cross-relationship with long periods of experience. Generally, vision frameworks are still reasonably gullible.

2.3 The Origin of OpenCV

OpenCV left an Intel Research action proposed to drive CPU-raised applications. Toward this end, Intel moved various endeavors that included constant beam following and moreover 3D show dividers. One of the product engineers working for Intel at the time was visiting schools. He saw that several top school social affairs, like the MIT Media Lab, used to have well-made similarly as inside open PC vision frameworks—code which was supplied starting with one understudy then onto the next and which gave each resulting understudy an important establishment while building up his own vision application. Rather than rehashing the fundamental capacities from starting, another understudy may begin by adding to that which preceded.

2.4 OpenCV Structure and Content

OpenCV left an Intel Research movement planned to drive CPU-raised applications. Toward this end, Intel pushed various endeavors that included continuous beam following and moreover 3D show dividers. One of the product engineers working for Intel at the time was visiting schools. He saw that two or three top school social events, like the MIT Media Lab, used to have well-made similarly as inside open PC vision foundations

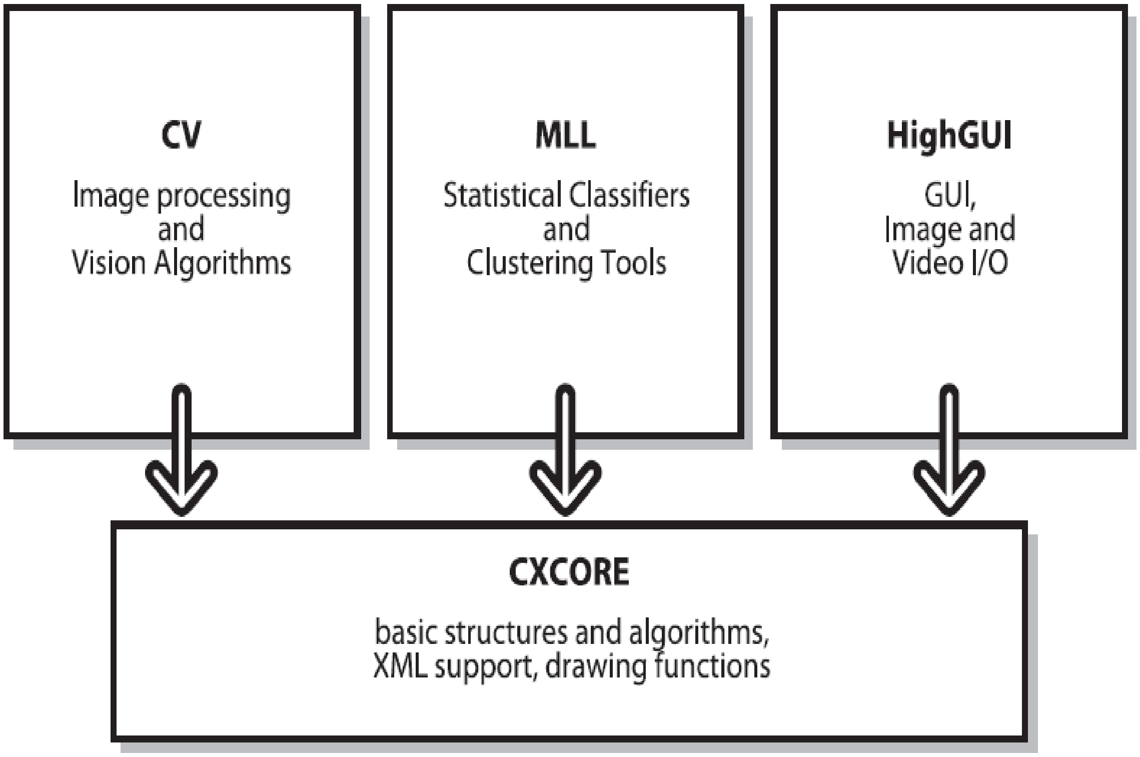


Fig 2. Parts of OpenCV

2.5 Why Open CV?

Specific

OpenCV was planned for picture handling. Each capacity and information structure has been arranged in perspective on an Image Processing application. Then, Matlab, is very conventional.

You can get almost everything on the planet by methods for tool compartments. It may be money related tool stash or then again concentrated DNA tool compartments

Speedy

Matlab is just excessively moderate. Matlab itself depended on Java. Similarly, Java depended on C. So when we run a Matlab program, our PC gets caught up with attempting to translate and assemble all that convoluted Matlab code. At that point it is transformed into Java, lastly executes the code.

In case we use C/C++, we don't waste such time. We direct give machine language code to the PC, and it gets executed. So in the end we get more picture taking care of, and not additionally interpreting.

Ensuing to doing some constant picture handling with both Matlab and OpenCV, we typically got low speeds, a point of confinement of around 4-5 outlines arranged each second with Matlab. With OpenCV in any case, we get genuine persistent dealing with at around 30 outlines being handled every second.

Beyond any confusion we give the prize for speediness – a progressively enigmatic language to handle, yet it's unquestionably of true worth . We can complete a lot more work, as calculate some extremely perplexing arithmetic on pictures utilizing C and still pull off adequate speeds for your application.

Efficient

MATLAB utilizes just an excessive amount of system assets. With OpenCV, we can pull off as pitiful as 10mb RAM for a constant application. Notwithstanding the way that with the present PCs, the RAM factor is surely not a noteworthy thing to be worried over. In any case, our tiredness identification framework is to be used inside a vehicle in a way that is non-meddlesome and little; so a low handling necessity is vital.

Subsequently we shall perceive as to how OpenCV is superior for a real-time detection system.

Machine Learning

The objective of AI is to transform information into data. Subsequent to having gained from a social affair of information, we need a machine that can address any question about the information:

- What are different information that are like given information? Is there a face in the picture?
- What sort of advertisement will impact the client?
- There is generally a cost parameter, hence the question arises:

Of the numerous items that we can profit from, which one will probably be purchased by the client if a promotion is appeared for it?

AI changes over information into data by identifying standards or examples from that infor1mation.

2.6 OpenCV’s Machine Learning Algorithms

The ML calculations that are incorporated into OpenCV are given as pursues. Every one of the calculations are available in the ML library separated from Mahala Nobis and K- implies, which are available in CVCORE, and the calculation of face recognition, which is available in CV.

Mahala Nobis:

It is a measure of distance that is responsible for the stretchiness of the data. We can divide out the covariance of the given data to find this out. In case of the covariance being the identity matrix (i.e. identical variance), this measure will be identical to the Euclidean distance.

K-means Algorithm:

It is an unsupervised clustering algorithm which signifies a distribution of data w.r.t. K centers, K being chosen by the coder. The difference between K-means and expectation maximization is that in K-means the centers aren't Gaussian. Also, the clusters formed look somewhat like soap bubbles, as centers compete to occupy the closest data points. All these cluster areas are usually used as a form of sparse histogram bin for representing the data.

Normal or Naïve Bayes classifier algorithm:

It is a generative classifier where features are often assumed to be of Gaussian distribution and also statistically independent from one another. This assumption is usually false. That's why it's usually known as a naïve Bayes classifier. That said, this method usually works surprisingly well.

Decision trees algorithm:

It is a partially discriminative classifier. The tree we talk about just finds a singular data feature and determines a threshold value of the current node which best divides the data into different classes. The data is broken into parts and the procedure is recursively repeated through the left as well as the right branches of the decision tree. Even if it is not the top performer, it's usually the first thing we try as it is fast and has a very high functionality.

Boosting:

It is a discriminative group of classifiers. In boosting, the final classification decision is made by taking into account the combined weighted classification decisions of the group of classifiers. We learn in training the group of classifiers one after the other. Each classifier present in the group is called a weak classifier. These weak classifiers are usually composed of single-variable decision trees known as ―stumps. Learning its classification decisions from the given data and also learning a weight for its vote based on its accuracy on the data are things the decision tree learns during training. While each classifier is trained one after the other, the data points are re-weighted to make more attention be paid to the data points in which errors were made. This continues until the net error over the entire data set, obtained from the combined weighted vote of all the decision trees present, falls below a certain threshold. This algorithm is usually effective when a very large quantity of training data is available.

Random trees algorithm:

It is a discriminative wood of a great deal of choice trees, every one of which is worked down to a maximal part profundity. At the season of adapting, each hub of each tree is permitted a decision of part factors, however just from an arbitrarily produced subset of the considerable number of information highlights. This makes sure that all the trees become statistically independent and a decision maker. In the run mode, all the trees get an unweighted vote. Random trees are usually quite effective. They can also perform regression by taking the average of the output numbers from every tree. Neural networks / Multilayer perceptron (MLP) algorithm:

It is a discriminative algorithm which almost always contains hidden units in between the output and the input nodes for better representation of the input signal. It is slow to train, however it is quite fast to run. It remains the best performer for applications like letter recognition.

Support vector machine (SVM) algorithm:

It is a discriminative classifier that is likewise equipped for doing relapse. Here, a separation work in the middle of two information focuses is characterized in a higher- dimensional space. (Anticipating information onto higher measurements helps in making the information more probable for direct division.) Support vector machine (SVM) gets the hang of isolating hyperplanes which maximally separate every one of the classes in the higher measurement. This will in general be the best when there is restricted information. Be that as it may, when huge informational indexes are accessible, boosting or arbitrary trees are liked.

Getting to use Machine Learning for Computer Vision

Typically, most calculations take an information vector having numerous highlights as info. Here, the quantity of highlights may number in the thousands. On the off chance that our undertaking is perceiving a particular sort of article— take for instance, an individual's face. The main issue that we experience is getting and marking the preparation information which falls into positive (for example there is a face in the window) and negative (for example no face) cases. We before long understand that countenances can show up at different scales: for example, their picture may comprise of just a couple of pixels, or we may take a gander at an ear which is filling the entire screen. More terrible still, faces are typically impeded. We need to characterize what we really mean when we state that a face is in the window.

Subsequent to having marked the information that was acquired from different sources, we ought to choose which highlights we have to remove from these items. Likewise, we should recognize what objects we are after. On the off chance that the countenances dependably seem upstanding, there is no purpose behind utilizing revolution invariant highlights and furthermore no explanation behind endeavoring to turn the articles before handling.

All in all, we should attempt to discover highlights that express a little invariance in the articles. These can be scale-tolerant histograms of angles or hues or even the prominent SIFT highlights.

When we have imperative foundation window data, we would first be able to evacuate it so as to enable different articles to emerge. At that point we play out our picture handling. This may comprise of normalizing the picture and after that registering the different highlights. The subsequent information vectors are altogether given the name that is related with the article, activity, or window. When the information is gotten and changed over into highlight vectors, we separate the information into preparing sets, approval sets and test sets. It is fitting to do our learning, approval, and testing utilizing a cross-approval structure. Here, the information is part into K subsets and we run different preparing (possibly approval) just as test sessions. Every session comprises of different arrangements of information that assume the jobs of preparing (approval) and test. The test outcomes acquired from these different sessions are utilized for averaging to get the last execution result. An increasingly precise picture of how the classifier performs when sent in task can be given by cross-approval.

Since our information is prepared, we should pick a classifier. Generally, the decision of the classifier is controlled by computational, information, and memory necessities. For specific applications, as online client inclination displaying, we have to prepare the classifier rapidly. In such a case, closest neighbors, ordinary Bayes, or choice trees ought to be a decent decision. At the point when memory is the essential thought, choice trees or neural systems are utilized for their space productivity.

When we have sufficient energy to prepare our classifier yet it needs to run rapidly, neural systems can be a decent decision, similarly as with ordinary Bayes classifiers and bolster vector machines. When we have room schedule-wise to prepare yet require high exactness, at that point boosting and irregular trees are great decisions. When we simply need a simple and reasonable check climate our highlights are picked well or not, at that point choice trees or closest neighbors ought to be utilized. For a decent out of the container characterization execution, boosting or irregular trees are attempted.

3.MediaPipe

3.1 Introduction:

MediaPipe offers ready-to-use yet customizable Python solutions as a prebuilt Python package. MediaPipe Python package is available on PyPI for Linux, macOS and Windows.

You can, for instance, activate a Python virtual environment:

```
$ python3 -m venv mp_env && source mp_env/bin/activate
```

Install MediaPipe Python package and start Python interpreter:

```
(Mp_env) $ pip install mediapipe
(Mp_env) $ python3
```

In Python interpreter, import the package and start using one of the solutions:

```
import mediapipe as mp
mp_face_mesh = mp. solutions. face_mesh
```

3.2 MediaPipe Python Framework:

The ready-to-use solutions are built upon the MediaPipe Python framework, which can be used by advanced users to run their own MediaPipe graphs in Python. Please see here for more info.

3.3 Building MediaPipe Python Package:

Follow the steps below only if you have local changes and need to build the Python package from source. Otherwise, we strongly encourage our users to simply run `pip install mediapipe` to use the ready-to-use solutions, more convenient and much faster.

MediaPipe PyPI currently doesn’t provide aarch64 Python wheel files. For building and using MediaPipe Python on aarch64 Linux systems such as Nvidia Jetson and Raspberry Pi, please read here.

1. Make sure that Bazel and OpenCV are correctly installed and configured for MediaPipe. Please see Installation for how to setup Bazel and OpenCV for MediaPipe on Linux and macOS.
2. Install the following dependencies.

Debian or Ubuntu:

```
$ sudo apt install python3-dev
$ sudo apt install python3-venv
$ sudo apt install -y protobuf-compiler

# If you need to build opencv from source.
$ sudo apt install cmake
```

macOS:

```
$ brew install protobuf

# If you need to build opencv from source.
$ brew install cmake
```

3.4 Windows:

Download the latest protoc win64 zip from the Protobuf GitHub repo, unzip the file, and copy the protoc.exe executable to a preferred location. Please ensure that location is added into the Path environment variable.

3. Activate a Python virtual environment.

```
4.     $ python3 -m venv mp_env && source mp_env/bin/activate
```

5. In the virtual environment, go to the MediaPipe repo directory.

6. Install the required Python packages.

```
7.     (Mp_env) mediapipe$ pip3 install -r requirements.txt
```

8. Generate and install MediaPipe package.

```
9.     (Mp_env) mediapipe$ python3 setup.py gen_protos
10.    (Mp_env) mediapipe$ python3 setup.py install --link-opencv
```

or

```
11.    (Mp_env) mediapipe$ python3 setup.py gen_protos
12.    (Mp_env) mediapipe$ python3 setup.py bdist_wheel
```

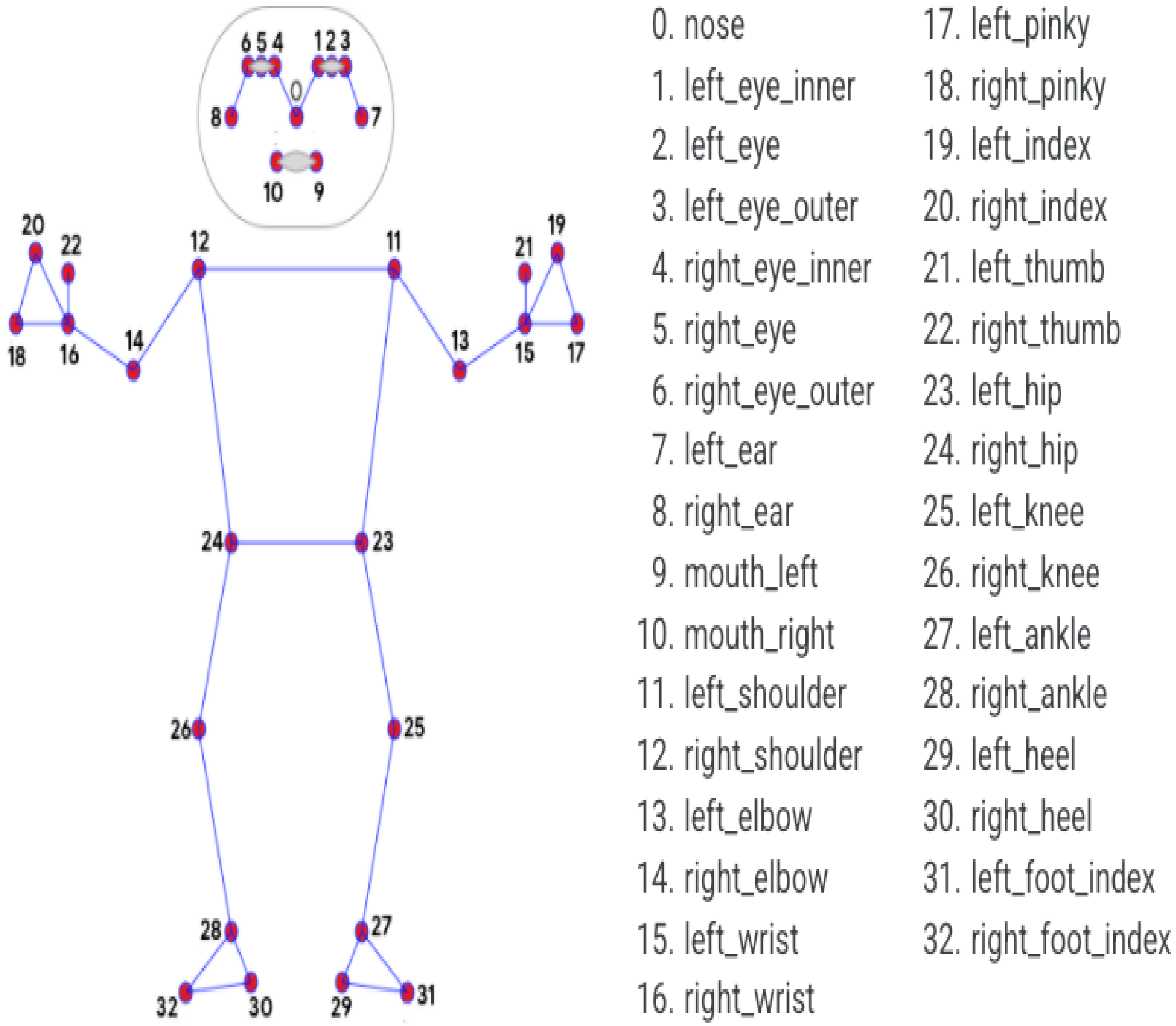


Fig3. Pose landmarks.

3.5 Mathematical functions

This module provides access to the mathematical functions defined by the C standard. These functions cannot be used with complex numbers; use the functions of the same name from the cmath module if you require support for complex numbers. The distinction between functions which support complex numbers and those which don't is made since most users do not want to learn quite as much mathematics as required to understand complex numbers. Receiving an exception instead of a complex result allows earlier detection of the unexpected complex number used as a parameter, so that the programmer can determine how and why it was generated in the first place.

The following functions are provided by this module. Except when explicitly noted otherwise, all return values are floats.

3.6 Number-theoretic and representation functions

math.ceil(*x*)

Return the ceiling of *x*, the smallest integer greater than or equal to *x*. If *x* is not a float, delegates to `x.ceil__()`, which should return an Integral value.

math.comb(*n*, *k*)

Return the number of ways to choose *k* items from *n* items without repetition and without order.

Evaluates to `n! / (k! * (n - k)!)` when `k <= n` and evaluates to zero when `k > n`.

Also called the binomial coefficient because it is equivalent to the coefficient of k-th term in polynomial expansion of the expression `(1 + x) ** n`.

math.isqrt(n)

Return the integer square root of the nonnegative integer n. This is the floor of the exact square root of n, or equivalently the greatest integer a such that $a^2 \le n$.

For some applications, it may be more convenient to have the least integer a such that $n \le a^2$, or in other words the ceiling of the exact square root of n. For positive n, this can be computed using `a = 1 + isqrt(n - 1)`.

math.lcm(*integers)

Return the least common multiple of the specified integer arguments. If all arguments are nonzero, then the returned value is the smallest positive integer that is a multiple of all arguments. If any of the arguments is zero, then the returned value is 0. lcm() without arguments returns 1.

math.ldexp(*x*, *i*)

Return x * (2**i). This is essentially the inverse of function frexp().

math.modf(*x*)

Return the fractional and integer parts of *x*. Both results carry the sign of *x* and are floats.

math.nextafter(x, y)

Return the next floating-point value after *x* towards *y*.

If *x* is equal to *y*, return *y*.

Examples:

- math.nextafter(x, math.inf) goes up: towards positive infinity.
- math.nextafter(x, -math.inf) goes down: towards minus infinity.
- math.nextafter(x, 0.0) goes towards zero.
- math.nextafter(x, math.copysign(math.inf, x)) goes away from zero.

3.7 Trigonometric functions

math. Cos(*x*)

Return the arc cosine of *x*, in radians. The result is between 0 and pi.

math.asin(*x*)

Return the arc sine of *x*, in radians. The result is between -pi/2 and pi/2.

math.atan(*x*)

Return the arc tangent of *x*, in radians. The result is between -pi/2 and pi/2.

math.atan2(*y*, *x*)

Return atan (y / x), in radians. The result is between -pi and pi. The vector in the plane from the origin to point (x, y) makes this angle with the positive X axis. The point of atan2() is that the signs of both inputs are known to it, so it can compute the correct quadrant for the angle. For example, atan (1) and atan2(1, 1) are both pi/4, but atan2(-1, -1) is -3*pi/4.

math.cos(*x*)

Return the cosine of *x* radians.

math. dist. (p, q*)*

Return the Euclidean distance between two points *p* and *q*, each given as a sequence (or iterable) of coordinates. The two points must have the same dimension.

4. SYSTEM DEVELOPMENT

4.1 Computational Analysis

Matching

System camera is used to monitor the person in the frame and detect the pose of the person.

4.1.1 Fitness Harmony Simulator Design:

A camera is setup that looks for persons in the input video stream and monitors frames of persons. In the event that a person is identified, mediapipe is connected and the pose of the person in the frame is detected.

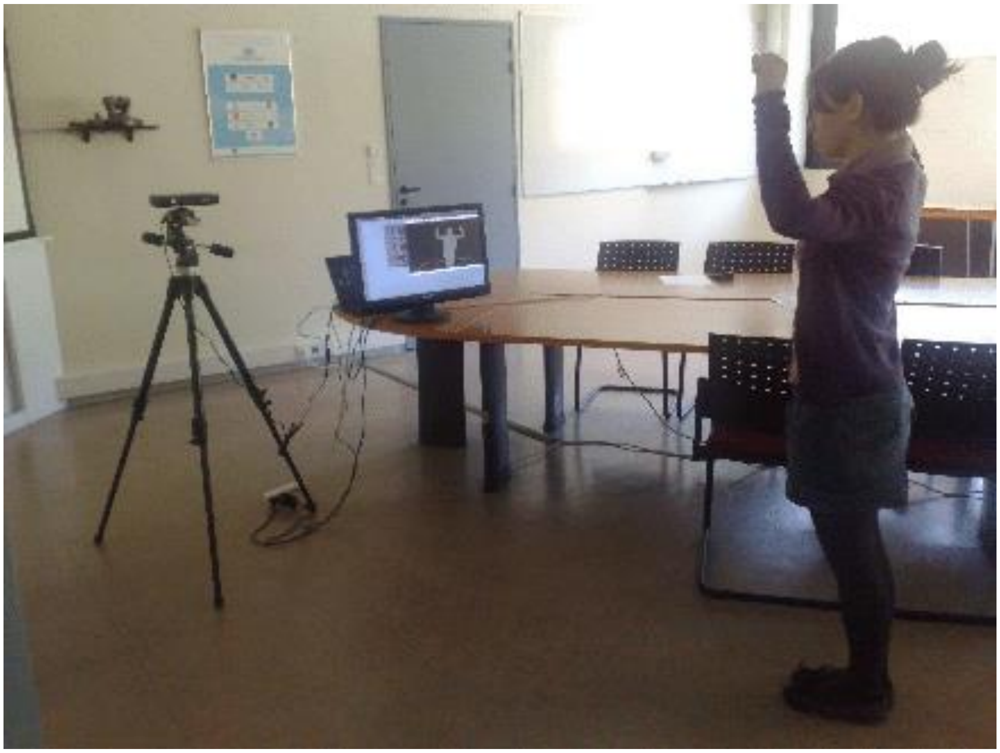


Fig4. Practical design of the system

If the angle crosses minimum threshold value and maximum threshold value for some frames time then it indicates that the person is doing the correct training. If the value of angle crosses minimum threshold and maximum threshold for a period of time then we will increase the count of the workout by 1. If the value of the count increases up to threshold value, then we will play the speech according to the workout they have completed. After the audio clip the audio clip follows the beep sound.

After listening the beep sound the person will get start training for next workout. It also follows the count. If the person completes all workouts, then we will play final audio as you have successfully completed all workouts. If the person listens final audio, then he wants to know that all workouts are completed

4.1.2.1 Developing Image Processing solutions using OpenCV & Mediapipe

OpenCV was developed keeping image processing in mind. Every function and data struct of OpenCV concerns itself with an Image Processing library. Comparatively, Matlab is hugely of generic use & slow.

Any usefulness can be accomplished by methods for tool kits in OpenCV, itmight be money related tool compartments or explicit DNA tool stash

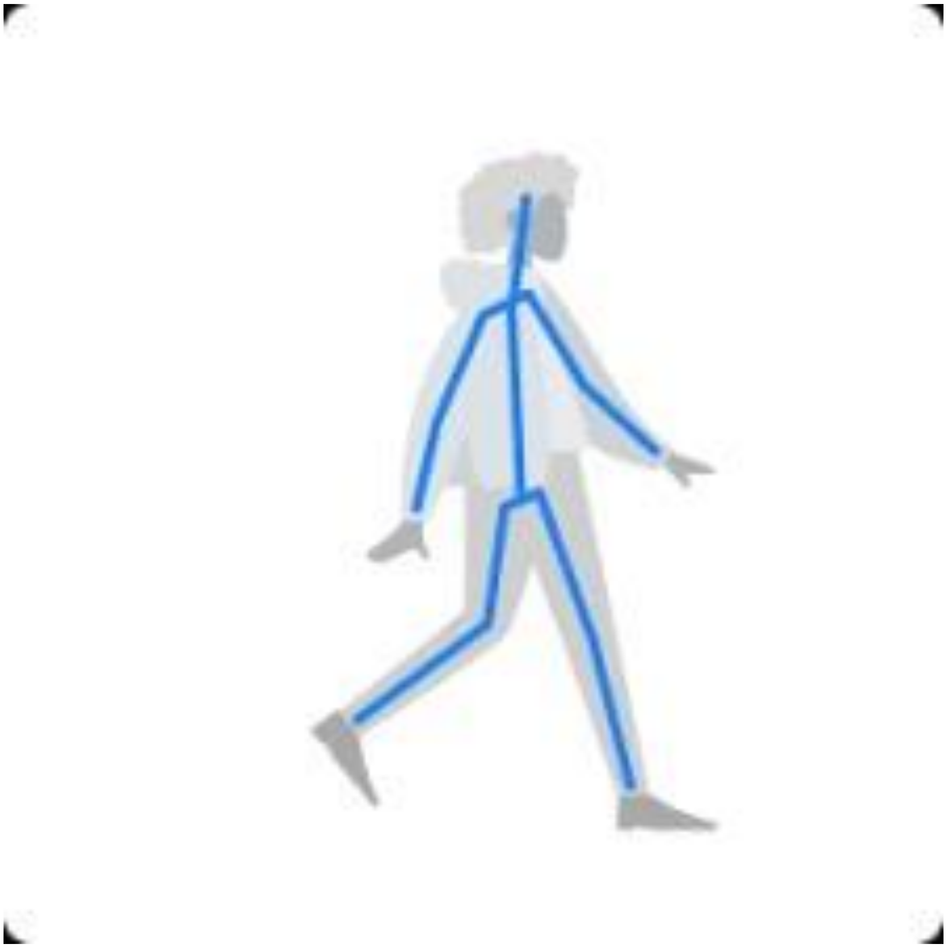


Fig5. Mediapipe using OpenCV

Also the Mediapipe library comes with a face detector and pose detector and hands detector.

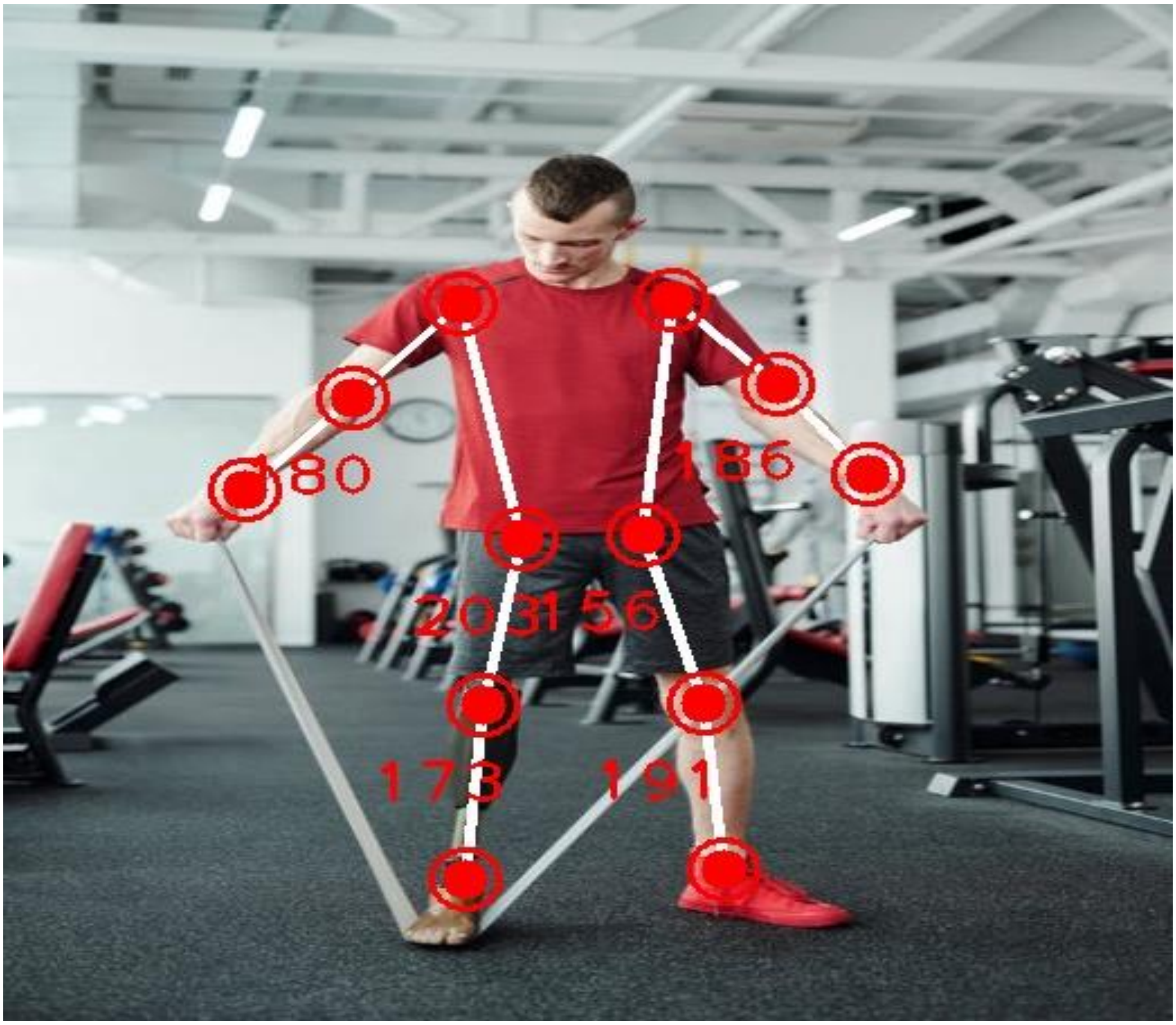


Fig6. Pose detection using mediapipe

5: ALGORITHM & IMPLEMENTATION

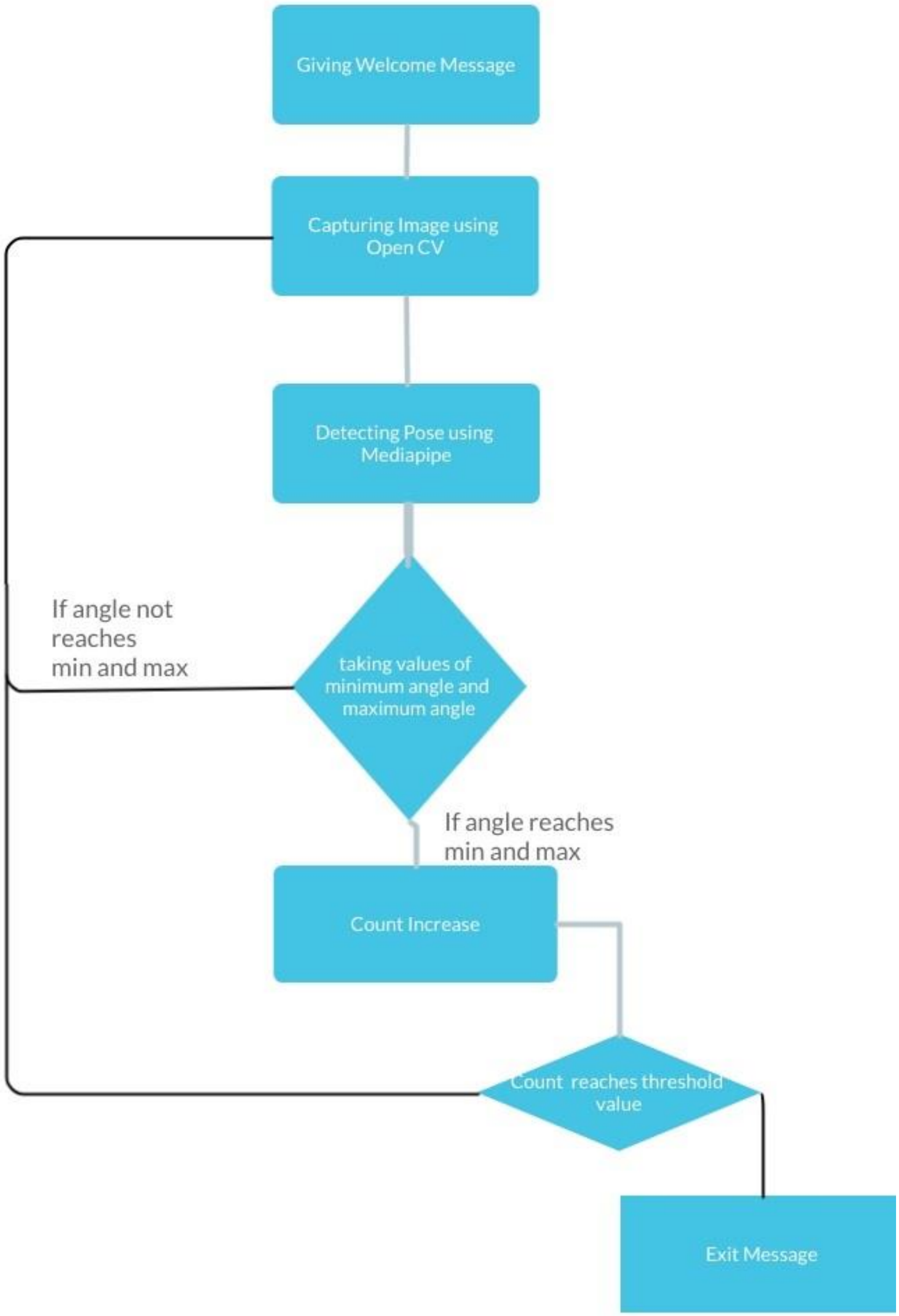

Giving Welcome Message
Capturing Image using Open CV
Detecting Pose using Mediapipe
If angle not reaches min and max
taking values of minimum angle and maximum angle
If angle reaches min and max
Count Increase
Count reaches threshold value
Exit Message

Fig7. Algorithm

5.2 Algorithmic steps: -

The overall algorithm is pretty straightforward one. First, we have used a camera which is setup at desirable position in front of person nearly 5 feet away.If person gets detected, the mediapipe detection task is applied and region of person is extracted.

Once we get the person region, we calculate the angle between two lines and if the angle is less than given threshold and angle greater than given threshold then count gets increases.

On the off chance that the angle demonstrates that the angle is not in region of threshold for long measure of time, the count will increases. For the functionalities of the system and to make it work efficiently we have used OpenCV, mediapipe and Python.

The implementation of the Fitness harmony simulator includes machine learning algorithms which are in turn included in OpenCV ML algorithms. There are numerous ML algorithms but for our purpose we required only the

Pose estimation algorithm.

It is fundamentally an item discovery ground-breaking application. Additionally, prepared frontal face identifier is accessible with the OpenCv circulation.

It works efficiently well overall. It can also be used to detect various different types of objects with the required software.

5.3 Starting to build the detector system with OpenCv

First, we create a new file fitness_harmony_simulator.py and write the following script in it .

```
import cv2
import numpy as np
import PoseModule as pm
import pyttsx3
from beeply import notes
```

Now for calculating the angles we need to compute the angle between two lines which are formed by using three points.

Next we need to import the pose estimation module in which we had written the code for finding angles and detecting the pose and in that module we had imported the mediapipe

Also, the package named pyttsx3 is needed for text to speech, beeply is needed for playing beep sound, and OpenCV for capturing video.

5.4 Pose detection and angle calculation

For detecting and localising person pose we will require the mediapipe library hence we import it.

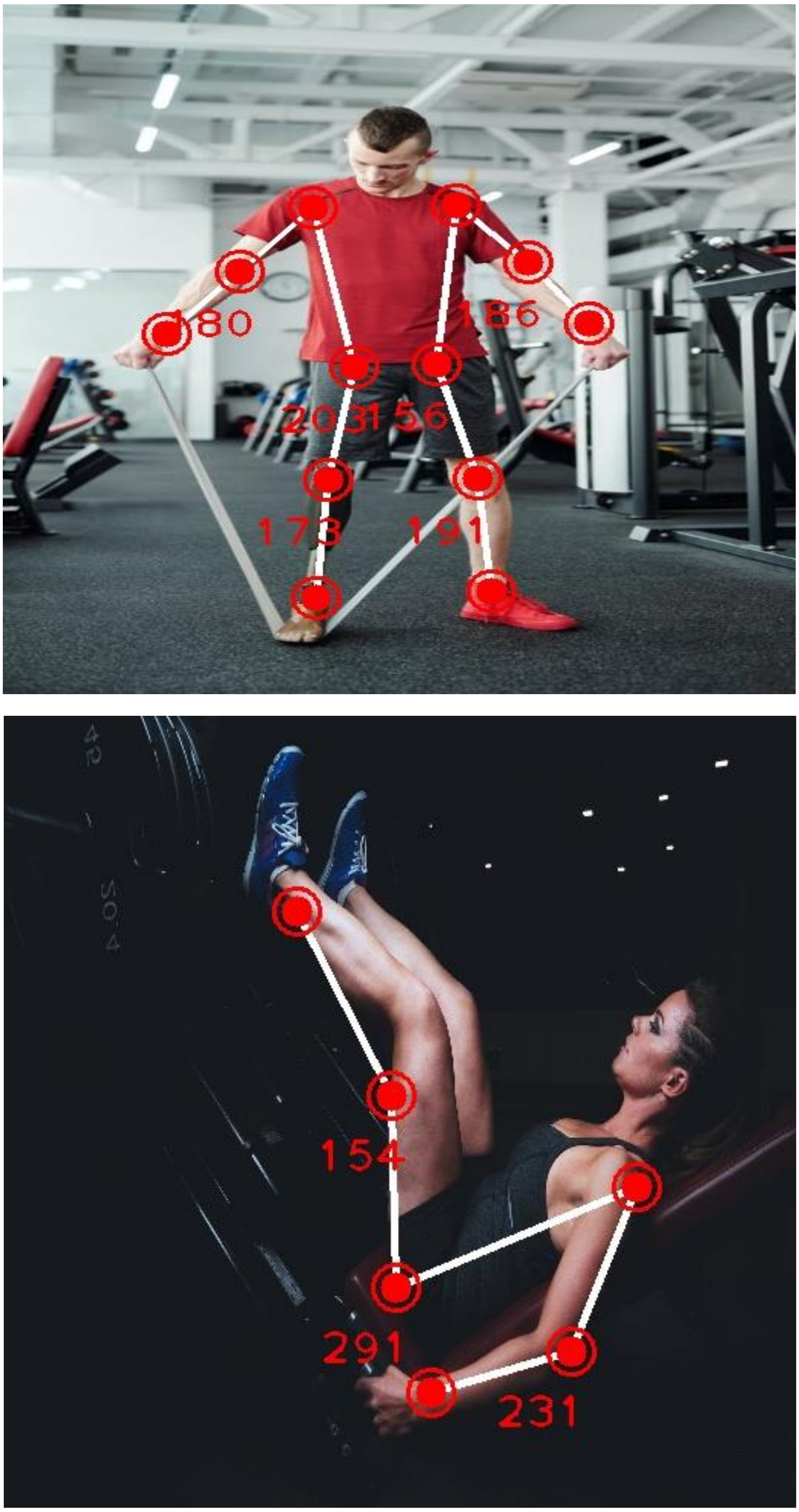
Angle between lines are calculated by using numpy and math library.

```
pyobj = pyttsx3.init()
pyobj.setProperty("rate", 100)
pyobj.say("Welcome to fitness harmony simulator Please
watch the video before beginning the workout")
pyobj.runAndWait()
mybeep = notes.beeps(1000)
mybeep.hear('C_')
# Sample workout
vid = cv2.VideoCapture('trimbicep.mp4')
detector = pm.poseDetector()
count = 0
dir = 0
while True:
    abc, img2 = vid.read()
    img2 = cv2.resize(img2, (1280, 720))
    img2 = detector.findPose(img2, False)
    lmList = detector.findPosition(img2, False)
    if len(lmList) != 0:
        # left Arm
        angle = detector.findAngle(img2, 12, 14, 16)
        # Left Arm
        per = np.interp(angle, (185, 50), (0, 100))
        bar = np.interp(angle, (185, 50), (650, 100))
        color = (255, 0, 255)
        if per == 100:
            color = (0, 255, 0)
            if dir == 0:
                count += 0.5
                dir = 1
        if per == 0:
            color = (0, 255, 0)
            if dir == 1:
                count += 0.5
                dir = 0
        print(count)
        cv2.rectangle(img2, (0, 550), (180, 720), (214,
245, 162), cv2.FILLED)
        cv2.putText(img2, str(int(count)), (45, 670),
cv2.FONT_HERSHEY_COMPLEX, 3,
                    (166, 108, 94), 10)
    cv2.imshow("Image", img2)
    if (count == 3):
        break
    if (cv2.waitKey(1) & 0xFF == ord('q')):
        break
vid.release()
cv2.destroyAllWindows()
```

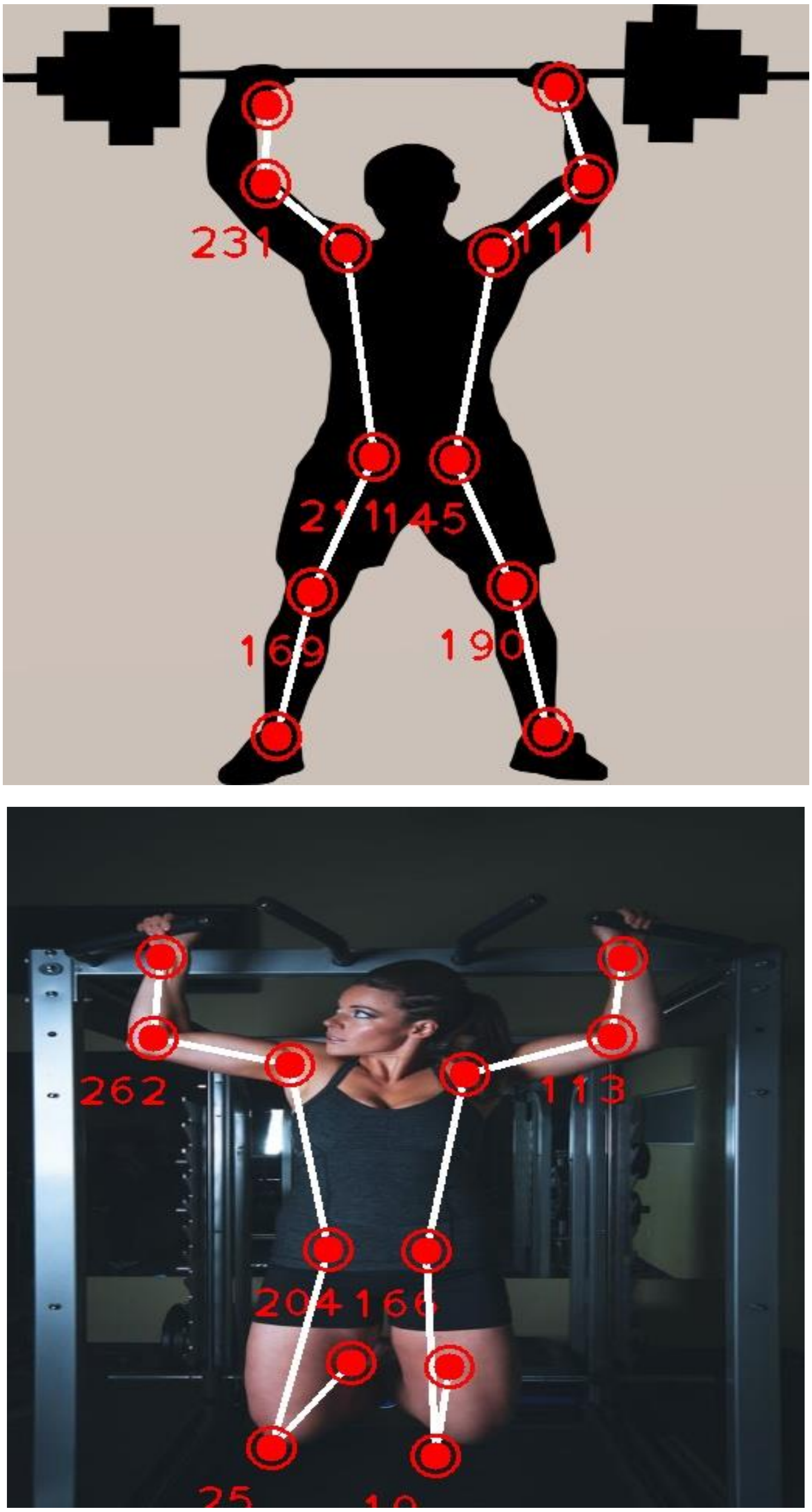
By using the above code we will capture the image or video from the computer camera by using OpenCV. After capturing the image or video we will applying the mediapipe library to that image or video. We are importing mediapipe in pose estimation module and by using pose estimation module we are using that mediapipe library. After getting the pose of the person we are calculating the angle between two lines.

Those angles are given by referring documents about the workouts. If the person who is doing the workouts Infront of camera and if he reaches minimum threshold angle and maximum threshold angle then we will increment the count by 1. If the count reaches threshold value, then we will get voice command to move to the next workout. If the angle doesn’t reach minimum or maximum threshold value so the count will not get increased.

6. RESULT & PERFORMANCE ANALYSIS



Sample Test 1&2.



Sample Test 3&4

7.CONCLUSION

Thus we would have successfully designed and developed partial implementation of the Fitness harmony simulator using Python and OpenCv, mediapipe along with the a cam to detect the face.

The system to be developed is to be tested and limitations are identified. The rest of the work will be done according to what is planned already.

Future Scope

- We can store all the data of training and recommend the good workouts.
- Recommending the nutrition food according to the level of workout they are doing.
- Creating an application for Android and IOS.

References

[1]. Ameratunga.S , Bailey.J, Connor.J, Civil.I, Dunn.R , Jackson.R , Norton.R, and Robinson.E, ―Driver sleepiness and risk of serious injury to car occupants: Population control study.‖*British Medical Journal*, vol. 324, 2002, pp. 1125–1129.

[2]. Bronte.S, Bergasa.L, Delgado.B, Garcia.I, Hernandez.N and Sevillano.M, ―Vision-based for a realistic driving simulator,‖ in *IEEE Intelligent Transportations Systems Conference (ITSC)*, 2010.

[3]. Distante.A, D'Orazio.T, Guaragnella.C and Leo.M, ―A visual approach for driver inattention detection,‖*Pattern Recogn.*, vol. 40, no. 8, 2007, pp. 2341–2355.

[4]. Bradski.G, Kaehler.A, -Learning OpenCV, *O'Reilly,* 2008

[5]. Igarashi.K ,Itou.K, Itakura.F, Miyajima.C, Ozawa.T, Takeda.K and Wakita.T, ―Driver identification using driving behavior signals,‖*IEICE - Trans. Inf. Syst.*, vol. E89- D, 2006.

[6]. Nakano.T, Suzuki.M, Yamamoto.N, Yamamoto.O and Yamamoto.S,

―Measurement of driver's consciousness by image processing a method for by eye- blinks coping with individual differences.‖*Systems, Man and Cybernetics*, vol. 4, 2006