­­­



# A Project Report on

**AI Curl Tracker**

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

**Bachelor of Technology** **in**

# Information Technology

**Submitted By**

**DEEPAK , SHASHANK , MAYANK RAMAN**

**2819460, 2819394, 2819410**

# Under the Supervision of Ms. Karuna (A.P., IT)

**Panipat Institute of Engineering & Technology, Samalkha, Panipat** **Affiliated to**

# Kurukshetra University Kurukshetra, India (2022)

# DECLARATION

I hereby declare that the work presented in this project report entitled “AI Curl Tracker”, in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Information Technology, submitted to Kurukshetra University, Kurukshetra, India is an authentic record of my own work carried out during the period from April, 2022 to June 2022 under the guidance of (Ms. Karuna Khurana).

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature

Name: DEEPAK , SHASHANK , MAYANK RAMAN

Roll No: 2819460, 2819394, 2819410

## CERTIFICATE

This is to certify that the Project-I report (IT-411N) entitled “AI Curl Tracker” done by DEEPAK,Roll No. 2819460 is an authentic work carried out by him/her at PIET, Samalkha, Panipat under my guidance. The matter embodied in this project work has not been submitted earlier for the award of any degree or diploma to the best of my knowledge and belief.

### Ms. Karuna Khurana Ms. Ashima Arya

(Project Guide) (Project Coordinator)

### Dr. Mukesh Chawla

HOD (IT)

PIET, Samalkha

**ACKNOWLEDGEMENT**

It gives us a great sense of pleasure to present the report of the B. Tech. Project undertaken during

B. Tech. Third Year. I owe special debt of gratitude to Ms. Karuna Mam, Department of Information Technology, PIET, Samalkha for his/her constant support and guidance throughout the course of our work. Her sincerity, thoroughness and perseverance have been a constant source of inspiration for us/me. It is only his cognizant efforts that our/my endeavors have seen light of the day.

I also take the opportunity to acknowledge the contribution of Dr. Mukesh Chawla, Head, Department of Information Technology, PIET, Samalkha for his full support and assistance during the development of the project.

I also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during the development of my project. Last but not the least, I acknowledge our friends for their contribution in the completion of the project.

Signature:

Name: DEEPAK , SHASHANK , MAYANK RAMAN

Roll No: 2819460, 2819394, 2819410

# ABSTRACT

AI Curl Tracker is an AI based Fitness Tracker using MediaPipe , OpenCV and Python. It can detect various body postures like bicep curls, squats and so on. We have used the pose estimation to find the correct points and using these points we’ll get the desire angles. This tracker is going to track different fitness activities and render the results to the screen.

We go to the gym on a daily basis and workout on a regular basis. Every day, we have at least three to four sets. If we have done the first two types of exercises and are bored of going to the third, which is the Biceps exercise, we will spend the entire time doing our activity counting our steps instead of making the appropriate angle of the exercise. To solve this issue, we're developing software that will track and measure the optimum workout angle. The issue is that the trainee is always focused on their count, and they occasionally overlook the angle of the activity, which is crucial in an exercise.This software can count your proper set counts, and the activity will aid in proper muscle breaking. The detection of the joints measures the trainee's angle and directs him to execute things correctly.

**Keywords: - Machine Learning, Artificial Intelligence, Python, Curl Tracker**

# Table of Contents

[DECLARATION](#_TOC_250050)

[CERTIFICATE](#_TOC_250049)

ACKNOWLEDGEMENTS

[ABSTRACT](#_TOC_250048)

LIST OF FIGURES

[CHAPTER 1: INTRODUCTION 1](#_TOC_250047)0

* 1. INTRODUCTION TO AI 10
  2. [APPLICATIONS OF AI 10](#_TOC_250046)
  3. WHY AI? 11
  4. What comprises to AI 11
  5. Types of AI 12
  6. Features of ML…………………………………………………………………………14
  7. Classification of ML……………………………………………………………………14

CHAPTER 2: Technology Used 15

* 1. INTRODUCTION TO OPENCV………………………………………………………15
  2. [OPEN CV INSTALLATION 15](#_TOC_250042)
  3. OPERATIONS OF OPENCV 15
  4. APPLICATIONS 16
  5. [INTRODUCTION TO MEDIAPIPE 16](#_TOC_250040)
  6. [MEDIAPIPE MODELS 17](#_TOC_250039)
  7. MEDIAPIPE TOOLKITS 17
  8. INTRODUCTION TO PYTHON…………………………………………………….21
  9. CHARACTERISTICS OF PYTHON…………………………………………………21

CHAPTER 3: Software Used (JUPYTER) 24

* 1. [INTRODUCTION TO JUPYTER 24](#_TOC_250035)
  2. [INTSALLATION 24](#_TOC_250034)
  3. [FEATURES 25](#_TOC_250033)
  4. [LANGUAGE SUPPORT 26](#_TOC_250032)
  5. [PACKAGE INSTALLATION 26](#_TOC_250031)

[CHAPTER 4: REQUIREMENT AND ANALYSIS 27](#_TOC_250023)

* 1. [PROBLEM DEFINITION 27](#_TOC_250022)
     1. REQUIREMENT SPECIFICATION 27

[CHAPTER 5: Coding 29](#_TOC_250021)

[CHAPTER 6: SCREENSHOTS 38](#_TOC_250020)

[CHAPTER 7: TESTING 42](#_TOC_250020)

[CHAPTER:8 CONCLUSION AND FUTURE SCOPE 44](#_TOC_250001)

CHAPTER 9: REFERENCE & BIBLOGRAPHY 45

# CHAPTER-1 INTRODUCTION

## Introduction to Artificial Intelligence?

Artificial Intelligence (AI) is a method of programming a computer, robot, or other object to think like a smart human. AI is the study of how the human brain thinks, learns, makes decisions, and works to solve problems. Finally, this research results in intelligent software systems. The goal of artificial intelligence is to improve computer functions that are linked to human understanding, such as reasoning, learning, and problem-solving.

The intelligence is intangible. It is composed of

* + - Reasoning
    - Learning
    - Problem Solving
    - Perception
    - Linguistic Intelligence

Reasoning, knowledge representation, planning, learning, natural language processing, realisation, and the ability to move and manipulate objects are all goals of AI research. In the field of general intelligence, there are long-term objectives. Statistical methodologies, computational intelligence, and classical coding AI are among the approaches. We use a variety of tools in our AI research on search and mathematical optimization, artificial neural networks, and methodologies based on statistics, probability, and economics. In the fields of science, mathematics, psychology, linguistics, philosophy, and others, computer science attracts AI.

## Applications of AI

* **Gaming** − In strategic games, AI plays a crucial role in allowing machines to consider a vast number of viable positions based on deep knowledge. For instance, chess, river crossings, and N-queens difficulties, to name a few..
* **Natural Language Processing** − Interact with a machine that understands human speech in natural language.
* **Expert Systems** − Users are given explanations and assistance by machines or software.
* **Vision Systems** − Visual input on the computer is understood, explained, and described by systems..
* **Speech Recognition** − While a human speaks to it, some AI-based speech recognition systems have the ability to hear, express as sentences, and understand their meanings. Siri and Google Assistant, for example.
* **Handwriting Recognition** − Handwriting recognition software reads handwritten text on paper, recognises letter shapes, and converts it to editable text.
* **Intelligent Robots** − Human-given instructions can be carried out by robots.

## Why Artificial Intelligence?

Before learning about Artificial Intelligence, it is necessary to understand what AI is and why it is important to learn. The following are some of the most compelling reasons to learn about AI:

• Using AI, you can design software or gadgets that can address real-world problems quickly and accurately, such as health difficulties, marketing, traffic congestion, and so on.

• You can develop your own personal virtual assistant using AI, such as Cortana, Google Assistant, Siri, and so on.

• With the help of AI, you can create robots that can work in environments where humans' lives is in jeopardy.

• AI paves the way for new technologies, devices, and business opportunities.

## What Comprises to Artificial Intelligence?

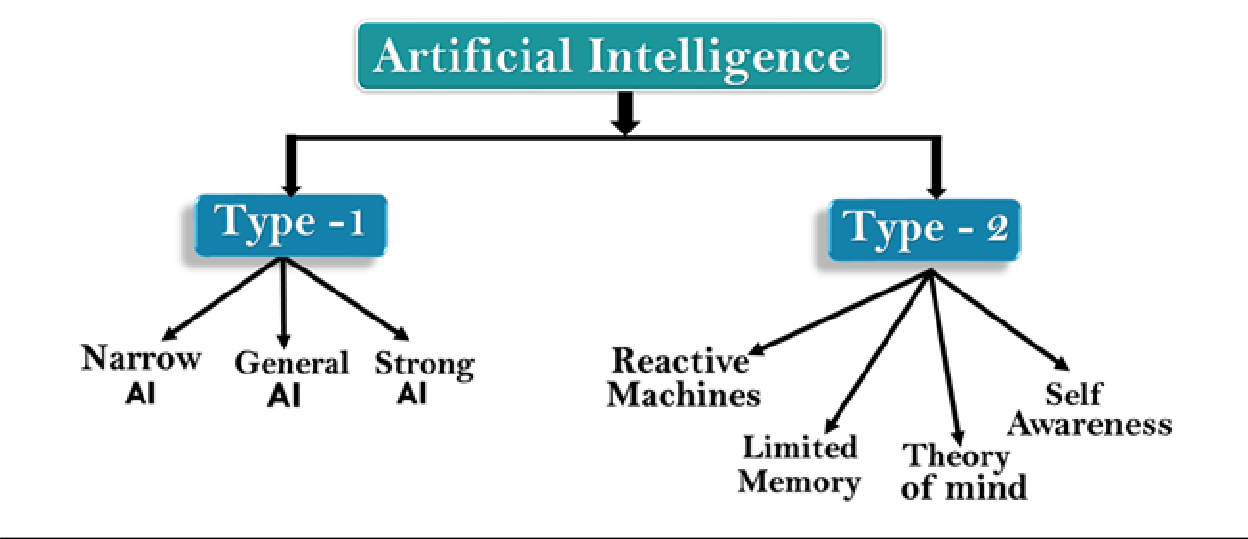
Artificial Intelligence is not solely a branch of computer science, despite the fact that it is so broad and encompasses a wide range of other aspects. To construct AI, we must first understand how intelligence is made up. Intelligence is an intangible component of our brain that is made up of reasoning, learning, problem-solving perception, language comprehension, and so on. Artificial Intelligence requires the following discipline to attain the above factors for a machine or software:

* + - Mathematics
    - Biology
    - Psychology
    - Sociology
    - Computer Science
    - Neurons Study
    - Statistics

## Types of Artificial Intelligence:

Machine learning is a rapidly evolving technology that allows computers to learn from previous data automatically. Machine learning employs a variety of algorithms to create mathematical models and make predictions based on past data or knowledge. It is being utilised for picture identification, speech recognition, email filtering, Facebook auto-tagging, recommender systems, and many more activities.

Artificial Intelligence can be divided in various types, there are mainly two types of main categorization which are based on capabilities and based on functionally of AI. Following is flow diagram which explain the types of AI.



AI type-1: Based on Capabilities

### Weak AI or Narrow AI:

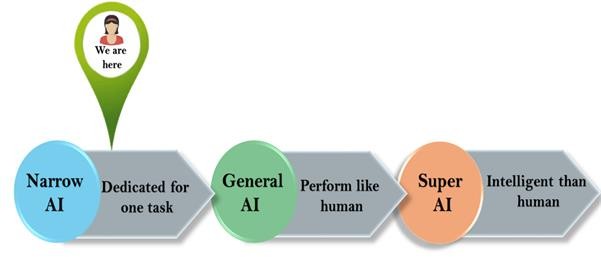
* + Narrow AI is a sort of AI that can intelligently do a certain task. In the area of Artificial Intelligence, Narrow AI is the most frequent and currently available AI. Because narrow AI is exclusively educated for one specific task, it cannot perform beyond its field or boundaries. As a result, it's also known as "weak AI." When narrow AI reaches its boundaries, it can fail in unexpected ways.
  + Apple Siri is an excellent example of Narrow AI, yet it only performs a limited set of functions.
  + IBM's Watson supercomputer falls under Narrow AI since it employs a combination of Expert systems, machine learning, and natural language processing..
  + Some Examples of Narrow AI are playing chess, purchasing suggestions on e-commerce site, self-driving cars, speech recognition, and image recognition.

### General AI:

* + General AI is a sort of intelligence that can execute any intellectual task as efficiently as a human.
  + The goal of general AI is to create a system that can think and act like a human on its own..
  + At the moment, there is no system that falls under general AI and can execute any work as well as a person.
  + The worldwide researchers are now focused on developing machines with General AI.
  + As systems with general AI are still under research, and it will take lots of efforts and time to develop such systems.

### Super AI:

* + • Super AI is a level of system intelligence at which machines may outsmart humans and execute any task better than humans with cognitive qualities. It's a result of AI in general.
  + • Some fundamental properties of powerful AI are the ability to think, reason, solve puzzles, make judgements, plan, learn, and communicate independently.
  + Super AI is still a hypothetical concept of Artificial Intelligence. Development of such systems in real is still world changing task.



Artificial Intelligence type-2: Based on functionality

Reactive Machines

Purely reactive machines are the most basic types of Artificial Intelligence.

Such AI systems do not store memories or past experiences for future actions.

These machines only focus on current scenarios and react on it as per possible best action.

IBM's Deep Blue system is an example of reactive machines.

Google's AlphaGo is also an example of reactive machines.

## Features of Machine Learning :

* Machine learning uses data to detect various patterns in a given dataset.
* It can learn from past data and improve automatically.
* It is a data-driven technology.
* Machine learning is much similar to data mining as it also deals with the huge amount of the data.

## Classification of Machine Learning:

machine learning can be classified into three types:

1. **Supervised learning**
2. **Unsupervised learning**
3. **Reinforcement learning**

### Supervised Learning

### Supervised learning is a form of machine learning method in which we feed sample labelled data to the machine learning system in order to train it, and it then predicts the output based on that data.

### 2) Unsupervised Learning

Unsupervised learning is a type of learning in which a machine learns without any human intervention. The machine is taught given a collection of data that hasn't been labelled, classified, or categorised, and the algorithm is expected to act on it without supervision. Unsupervised learning aims to reorganise input data into new features or a collection of objects with similar patterns.

### 3) Reinforcement Learning

Reinforcement learning is a feedback-based learning strategy in which a learning agent is rewarded for correct actions and punished for incorrect ones. With these feedbacks, the agent learns automatically and improves its performance. The agent interacts with and investigates the environment in reinforcement learning. An agent's purpose is to earn the greatest reward points, so it enhances its performance.

# CHAPTER-2

**Technology Used**

## OPEN CV:

## OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was established to provide a standard infrastructure for computer vision applications and to accelerate the use of machine perception in consumer products. As a BSD-licensed programme, OpenCV makes it straightforward for businesses to utilise and modify the code. OpenCV is one of the many prepackaged packages and libraries that make our lives easier..

## OpenCV Installation :

## Install using Anaconda - Anaconda is a free and open-source conditional distribution of the Python and R computer languages for scientific computing, with the goal of making package management and deployment easier.

## For Windows - Pip may be used to install OpenCV on Windows. Pip is a de facto standard package management system for installing and maintaining Python software packages, and it comes with Python by default..

## For Mac - You can use homebrew to install OpenCV as it makes it really easy.

## Basic Operation on images using opencv :

## Adding Images – We can add two images using a single command

## Result = cv2.add(image\_1 , image\_2)

## Blending Images - Blending images is similar to adding images, except that the contribution of each image to the final image can be regulated. Basically, we'll utilise blending instead of simple addition if we want one image to be more emphasised and the other to be more dim after they're blended..

## Image Smoothing - Image smoothing is a useful feature that is usually applied before photos are sent to a machine learning model. Passing a picture via a low-pass filter is commonly used to remove noise and high-frequency features.

## Scaling - Scaling is essentially the same as resizing an image, i.e. making it bigger or smaller. In OpenCV, the resize function is used to scale the images. INTER CUBIC, INTER LINEAR, and INTER AREA are the three types of resizing.

## Rotation – We can rotate a picture around an axis for a specific angle by using rotation.

## Applications of OpenCV:

### Robotics Application-

* Localization − Determine robot location automatically
* Navigation
* Obstacles avoidance
* Assembly (peg-in-hole, welding, painting)
* Manipulation (e.g. PUMA robot manipulator)
* Human Robot Interaction (HRI) − Intelligent robotics to interact with and serve people

### Medicine Application-

* Classification and detection (e.g. lesion or cells classification and tumor detection)
* 2D/3D segmentation
* 3D human organ reconstruction (MRI or ultrasound)
* Vision-guided robotics surgery

### Industrial Automation Application-

* Industrial inspection (defect detection)
* Assembly
* Barcode and package label reading
* Object sorting
* Document understanding (e.g. OCR)

### Security Application-

* Biometrics (iris, finger print, face recognition)
* Surveillance − Detecting certain suspicious activities or behaviors

## Features :

* Read and write images
* Capture and save videos
* Process images (filter, transform)
* Perform feature detection
* Detect specific objects such as faces, eyes, cars, in the videos or images.
* Analyze the video, i.e., estimate the motion in it, subtract the background, and track objects in it.

## Introduction to Mediapipe :

## MediaPipe is a framework for creating machine learning pipelines for time-series data such as video and audio. This cross-platform Framework is compatible with desktop/server, Android, iOS, and embedded devices such as the Raspberry Pi and Jetson Nano.

## It provides us:

## End to end acceleration: for building models and applications

## Build one, deploy anywhere: use the unified solution across Android/iOS, desktop, IOT devices etc.

## Ready-to-use solutions: ****c****utting-edge ML solutions demonstrating full power of the framework.

## Free and open source: the whole framwork is publically available and customizable.

## Mediapipe Models :

## Anatomical Models –

## Hand Tracking

## Pose Tracking

## Facial Mesh Tracking

## Holstic Tracking

## Segmentation Models

## Seflie Segmentation

## Hair Segmentation

## Object Models

## 2D Object Detection/Tracking

## 3D Object Detection and Pose Estimation

## Mediapipe Toolkit:

## MediaPipe Toolkit comprises the ****Framework****and the ****Solutions.****The following diagram shows the components of the MediaPipe Toolkit.

## Graphical user interface, chart, website, treemap chart Description automatically generated

1. **Framework -** The Framework is written in C++, Java, and Obj-C, which consists of the following APIs.
   1. **Calculator API**
   2. **Graph construction API**
   3. **Graph Execution API**
2. **Graphs**  - A Graph is the name given to the MediaPipe perception pipeline. Let's take the first option, Hands, as an example. We supply a stream of photographs as input, and the images are produced with hand landmarks.

A picture containing text, indoor

Description automatically generated

1. **Calculator -** These are specific computation units written in C++ with assigned tasks to process. The packets of data ( Video frame or audio segment ) enter and leave through the **ports**in a calculator. When initializing a calculator, it declares the packet payload type that will traverse the port. Every time a graph runs, the Framework implements **Open**, **Process,**and **Close**methods in the calculators. **Open**initiates the calculator; the **process**repeatedly runs when a packet enters. The process is **closed**after an entire graph run.calculator types :-
   1. **Pre-processing calculators**- are family of image and media processing calculators. The ImageTransform and ImageToTensors in the graph above fall in this category.
   2. **Inference calculators-**allow native integration with Tensorflow and Tensorflow Lite for ML inference.
   3. **Post-processing calculators-** perform ML post-processing tasks such as detection, segmentation, and classification. TensorToLandmark is a post-processing calculator.
   4. **Utility calculators-** are a family of calculators performing final tasks such as image annotation.
2. **MediaPipe Solutions -** Solutions are open-source pre-built examples based on a specific TensorFlow or TFLite model that has already been trained. The Framework is the foundation for MediaPipe Solutions. It currently offers sixteen options, which are shown below.
   1. **Face Detection**
   2. **Face Mesh**
   3. **Iris**
   4. **Pose**
   5. **Holistic**
   6. **Selfie Segmentation**
   7. **Hair Segmentation**
   8. **Object Detection**
   9. **Box Tracking**
   10. **Instant motion tracking**
   11. **Objectron**
   12. **Knift**
   13. **Autoflip**
   14. **Mediasequence**
   15. **Youtube 8**



## Introduction to Python.

Python is a high-level programming language that is interpreted, interactive, object-oriented, and general-purpose. Guido van Rossum designed it between 1985 and 1990. Python source code is also available under the GNU General Public License, just like Perl (GPL). This tutorial provides sufficient knowledge of the Python programming language..

### Why to Learn Python?

Python is a scripting language that is high-level, interpreted, interactive, and object-oriented. Python is intended to be a very understandable language. It typically uses English terms instead of punctuation, and it has fewer syntactical structures than other languages.

Python is a must-have skill for students and working professionals who want to become exceptional software engineers, especially if they work in the Web Development field. I'll go over some of the primary benefits of learning Python:

## • Python is Interpreted Python is handled by the interpreter at runtime. Before running your software, you do not need to compile it. This is similar to the programming languages PERL and PHP.

## • Python is Interactive You can sit at a Python prompt and write your programmes by interacting directly with the interpreter.

## • Python is Object-Oriented Python supports the Object-Oriented programming style or approach, which encapsulates code inside objects.

## • Python is a Beginner's Language Python is an excellent language for beginning programmers, as it allows for the creation of a wide range of programmes, ranging from simple text processing to web browsers and games.

## Characteristics of Python:

Following are important characteristics of Python Programming −

* + It supports functional and structured programming methods as well as OOP.
  + It can be used as a scripting language or can be compiled to byte-code for building large applications.
  + It provides very high-level dynamic data types and supports dynamic type checking.
  + It supports automatic garbage collection.
  + It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.



## Python Environment Variables

Here are important environment variables, which can be recognized by Python −

### Sr.No. Variable & Description

**PYTHONPATH**

1. It has a role similar to PATH. This variable tells the Python interpreter where to locate the module files imported into a program. It should include the Python source library directory

and the directories containing Python source code. PYTHONPATH is sometimes preset by the Python installer.

### PYTHONSTARTUP

1. It contains the path of an initialization file containing Python source code. It is executed every time you start the interpreter. It is named as .pythonrc.py in Unix and it contains commands that load utilities or modify PYTHONPATH.

### PYTHONCASEOK

3

It is used in Windows to instruct Python to find the first case-insensitive match in an

import statement. Set this variable to any value to activate it.

### PYTHONHOME

4

It is an alternative module search path. It is usually embedded in the PYTHONSTARTUP

or PYTHONPATH directories to make switching module libraries easy.

# CHAPTER-3

**SOFTWARE USED (JUPYTER)**

## Introduction to Jupyter Notebook :

Jupyter is an internet tool that allows users to create and share collaborative documents called notebooks. Text, equations, and images can all be included in Jupyter notebooks, as well as live Python code that the user can interact with. Python is a widely used programming language for a variety of purposes, including scientific computing, and is well-known for being simple to learn for new programmers.

The goal of this introduction is to provide the reader enough information on Jupyter and Python to browse and interact with the Jupyter Guide to Linear Algebra. For individuals interested in learning more about these themes, there are various resources available online. At the end, there is a list of references.

## Installation :

In python - $ pip install jupyter

Anaconda is the second most popular Python distribution. You can use Anaconda's own installer tool, called conda, to install a third-party package. However, Anaconda comes preconfigured with a number of scientific libraries, including the Jupyter Notebook, so you don't need to do anything but install Anaconda.

**Graphical user interface, application, email

Description automatically generated**

**Fig: Jupyter Notebook**

**.**

## Features

A fundamental feature of the Jupyter notebook is the ability to display charts that are generated by running code cells. The IPython kernel is designed to function in combination with the matplotlib plotting package to allow this feature. Specific charting libraries are supported by the kernel.

**Code autocompletion**. This is the first feature I recommend activating because it is extremely useful for all coders.

**Collapsible Headings**. Headings can be used to divide the notebook into several sections. This is very beneficial if your programme is quite long.

**Codefolding in Editor**. This functionality allows us to fold and unfold multiple lines of code within a single code cell.

**Table of contents**. Every book has a table of contents, and I believe that every Jupyter Notebook programme should have one as well (TOC).

**ExecuteTime** If you're still using percent percent timeit to keep track of how long your code takes to run, you'll enjoy the nbextension 'ExecuteTime.'

## Language support

Jupyter Notebook comes with basic support for the majority of programming languages out of the box. Syntax highlighting, bracket matching, code folding, and customisable snippets are all included in this basic package. Python, R, and more programming languages are supported

## Packages Installation

We can install packages using (pip) command.

E.g.- pip install opencv -python

# CHAPTER-4 REQUIREMENT AND ANALYSIS

System Analysis is about complete understanding of existing systems and finding where the existing system fails. The solution is determined to resolve issues in the proposed system. It defines the system. The system is divided into smaller parts. Their functions and inter relation of these modules are studied in system analysis. The complete analysis is followed below.

## Problem Statement

## We go to the gym on a daily basis and workout on a regular basis. Every day, we have at least three to four sets. If we have done the first two types of exercises and are bored of going to the third, which is the Biceps exercise, we will spend the entire time doing our activity counting our steps instead of making the appropriate angle of the exercise. To solve this issue, we're developing software that will track and measure the optimum workout angle. The issue is that the trainee is always focused on their count, and they occasionally overlook the angle of the activity, which is crucial in an exercise. This software can count your proper set counts, and the activity will aid in proper muscle breaking. The detection of the joints measures the trainee's angle and directs him to execute things correctly.

## HARDWARE AND SOFTWARE REQUIREMENTS

The software is designed to be light-weighted so that it doesn’t be a burden on the machine running it. This system is being build keeping in mind the generally available hardware and software compatibility. Here are the minimum hardware and software requirement for AI curl tracker.

Hardware:

* Processor : core i3
* Storage Required : 2 GB
* RAM : 4 GB
* Storage type : SSD/HDD

**Software Requirements**:

* Operating System : Windows 7/10/11
* Programming Language : Python
* Technology Used : Artificial Intelligence , Machine Learning

# CHAPTER- 5 CODING

AI\_Curl\_Tracker

June 12, 2022

[ ]:

# 1 Install and Import Dependencies

[2]:

pip install mediapipe opencv-python

*#mediapipe*

**import cv2**

**import mediapipe as mp import numpy as np**

*#numpy is a python library that provide N-D Array*

mp\_drawing=mp.solutions.drawing\_utils

*# mp.solutions.drawing\_utils class will allow us to visualize the landmarks*␣

*‹→after detection,*

mp\_pose=mp.solutions.pose

*#Include pose library*

[10]:

*#Video Feed*

cap=cv2.VideoCapture(0)

*#VideoCapture() method of cv2 library is used to read and start live streaming.*

*‹→ Its Possible values if either 0 and -1*

*#device index ie 0 : It is just the number to specify the camera.*

**while** cap.isOpened: ret,frame=cap.read()

*#When we apply command cap.read() the first frame from our video file will*␣

*‹→be loaded.*

*#It will be stored in a variable frame. If we call this command again, the*␣

*‹→second frame will be loaded and so on.*

*#Variable ret is a boolean data type that returns True if we are able to*␣

*‹→execute the read function successfully.*

*#ret will obtain return value getting from the camera frame either true or*␣

*‹→false*

cv2.imshow('Mediapipe Feed',frame)

*#Displays an image in the specified window.*

**if**(cv2.waitKey(10) & 0xFF==ord('q')):

*#delay*

*# Waits for a pressed key.*

*#where 10 is the delay in miniseconds*

*#if we press 'q' then it will return in string but we need answer in*␣

*‹→binary form*

*# so we use hexadecimal 0xFF i.e 255 in decimal, so it will convert string*␣

*‹→into binary*

*#with the help of and operation*

**break**

cap.release()

*# Closes video file or capturing device.*

cv2.destroyAllWindows()

*#Destroys all of the HighGUI windows.*

# 2 Make Detections

[14]: cap=cv2.VideoCapture(0)

*#setup mediapipe instance*

**with** mp\_pose.Pose(min\_detection\_confidence=0.5 ,min\_tracking\_confidence=0.5) **as**␣

*‹→*pose:

*# It is used to specify the minimum confidence value with which the*␣

*‹→detection from the landmark-tracking model*

*#must be considered as successful. #Its default value is 0.5*

*#Setting it to a higher value can increase robustness of the solution, at*␣

*‹→the expense of a higher latency.*

**while** cap.isOpened():

*#Returns true if video capturing has been initialized already.*

ret,frame=cap.read()

*#Grabs, decodes and returns the next video frame.*

*#ret will obtain return value getting from the camera frame either true*␣

*‹→or false*

*#Recolor image to RGB*

image=cv2.cvtColor(frame,cv2.COLOR\_BGR2RGB)

*#cvtColor:- Converts an image from one color space to another.*

image.flags.writeable=**False**

*#Make Detections*

results=pose.process(image)

*# Recolor back to BGR*

image.flags.writeable = **True**

image = cv2.cvtColor(image, cv2.COLOR\_RGB2BGR)

*#Render Detections # Draw landmarks*

mp\_drawing.draw\_landmarks(image,results.pose\_landmarks,mp\_pose.

*‹→*POSE\_CONNECTIONS,

mp\_drawing.DrawingSpec(color=(245,117,66),␣

*‹→*thickness=2, circle\_radius=2),

mp\_drawing.DrawingSpec(color=(245,66,230),␣

*‹→*thickness=2, circle\_radius=2)

)

*#Drawingspec:-Draws the detection bounding box and keypoints on the*␣

*‹→image*

cv2.imshow('Mediapipe Feed', image)

**if** cv2.waitKey(10) & 0xFF == ord('q'): *# # Break gracefully*

**break**

cap.release() cv2.destroyAllWindows()

[17]:

*#mp\_drawing.DrawingSpec??*

[ ]:

cap = cv2.VideoCapture(0)

*## Setup mediapipe instance*

**with** mp\_pose.Pose(min\_detection\_confidence=0.5, min\_tracking\_confidence=0.5) **as**␣

*‹→*pose:

**while** cap.isOpened():

ret, frame = cap.read()

*# Recolor image to RGB*

image = cv2.cvtColor(frame, cv2.COLOR\_BGR2RGB) image.flags.writeable = **False**

*# Make detection*

results = pose.process(image)

*# Recolor back to BGR*

image.flags.writeable = **True**

image = cv2.cvtColor(image, cv2.COLOR\_RGB2BGR)

*# Extract landmarks*

**try**:

landmarks = results.pose\_landmarks.landmark print(landmarks)

**except**:

**pass**

# Determining Joints

*# Render detections*

mp\_drawing.draw\_landmarks(image, results.pose\_landmarks, mp\_pose.

*‹→*POSE\_CONNECTIONS,

mp\_drawing.DrawingSpec(color=(245,117,66),␣

*‹→*thickness=2, circle\_radius=2),

mp\_drawing.DrawingSpec(color=(245,66,230),␣

*‹→*thickness=2, circle\_radius=2)

)

cv2.imshow('Mediapipe Feed', image)

**if** cv2.waitKey(10) & 0xFF == ord('q'):

**break**

cap.release() cv2.destroyAllWindows()

[21]:

len(landmarks)

[21]: 33

[23]:

**for** lmark **in** mp\_pose.PoseLandmark: print(lmark)

PoseLandmark.NOSE PoseLandmark.LEFT\_EYE\_INNER PoseLandmark.LEFT\_EYE PoseLandmark.LEFT\_EYE\_OUTER PoseLandmark.RIGHT\_EYE\_INNER PoseLandmark.RIGHT\_EYE PoseLandmark.RIGHT\_EYE\_OUTER PoseLandmark.LEFT\_EAR PoseLandmark.RIGHT\_EAR PoseLandmark.MOUTH\_LEFT PoseLandmark.MOUTH\_RIGHT PoseLandmark.LEFT\_SHOULDER PoseLandmark.RIGHT\_SHOULDER PoseLandmark.LEFT\_ELBOW PoseLandmark.RIGHT\_ELBOW PoseLandmark.LEFT\_WRIST PoseLandmark.RIGHT\_WRIST PoseLandmark.LEFT\_PINKY PoseLandmark.RIGHT\_PINKY PoseLandmark.LEFT\_INDEX PoseLandmark.RIGHT\_INDEX

PoseLandmark.LEFT\_THUMB PoseLandmark.RIGHT\_THUMB PoseLandmark.LEFT\_HIP PoseLandmark.RIGHT\_HIP PoseLandmark.LEFT\_KNEE PoseLandmark.RIGHT\_KNEE PoseLandmark.LEFT\_ANKLE PoseLandmark.RIGHT\_ANKLE PoseLandmark.LEFT\_HEEL PoseLandmark.RIGHT\_HEEL PoseLandmark.LEFT\_FOOT\_INDEX PoseLandmark.RIGHT\_FOOT\_INDEX

[24]:

landmarks[mp\_pose.PoseLandmark.LEFT\_SHOULDER.value].visibility

[24]: 0.9353919625282288

[26]:

landmarks[mp\_pose.PoseLandmark.LEFT\_ELBOW.value]

[26]: x: 1.0102804899215698

y: 1.2574375867843628

z: -1.1176338195800781

visibility: 0.17017368972301483

[27]:

landmarks[mp\_pose.PoseLandmark.LEFT\_WRIST.value]

[27]: x: 0.9589055180549622

y: 1.3104153871536255

z: -1.8593040704727173

visibility: 0.030339263379573822

[28]:

**def** calculate\_angle(a,b,c): a = np.array(a) *# First* b = np.array(b) *# Mid*

c = np.array(c) *# End*

radians = np.arctan2(c[1]-b[1], c[0]-b[0]) - np.arctan2(a[1]-b[1],␣

*‹→*a[0]-b[0])

angle = np.abs(radians\*180.0/np.pi)

**if** angle >180.0: angle = 360-angle

**return** angle

# ANGLES CALCULATING

[29]:

shoulder = [landmarks[mp\_pose.PoseLandmark.LEFT\_SHOULDER.value].

*‹→*x,landmarks[mp\_pose.PoseLandmark.LEFT\_SHOULDER.value].y]

elbow = [landmarks[mp\_pose.PoseLandmark.LEFT\_ELBOW.value].x,landmarks[mp\_pose.

*‹→*PoseLandmark.LEFT\_ELBOW.value].y]

wrist = [landmarks[mp\_pose.PoseLandmark.LEFT\_WRIST.value].x,landmarks[mp\_pose.

*‹→*PoseLandmark.LEFT\_WRIST.value].y]

[31]:

shoulder, elbow, wrist

[31]: ([0.8768285512924194, 0.9720253348350525],

[1.0102804899215698, 1.2574375867843628],

[0.9589055180549622, 1.3104153871536255])

[33]:

calculate\_angle(shoulder, elbow, wrist)

[33]: 110.82031204327163

[34]:

tuple(np.multiply(elbow, [640, 480]).astype(int))

[34]: (646, 603)

[35]: cap = cv2.VideoCapture(0)

*## Setup mediapipe instance*

**with** mp\_pose.Pose(min\_detection\_confidence=0.5, min\_tracking\_confidence=0.5) **as**␣

*‹→*pose:

**while** cap.isOpened():

ret, frame = cap.read()

*# Recolor image to RGB*

image = cv2.cvtColor(frame, cv2.COLOR\_BGR2RGB) image.flags.writeable = **False**

*# Make detection*

results = pose.process(image)

*# Recolor back to BGR*

image.flags.writeable = **True**

image = cv2.cvtColor(image, cv2.COLOR\_RGB2BGR)

*# Extract landmarks*

**try**:

landmarks = results.pose\_landmarks.landmark

*# Get coordinates*

shoulder = [landmarks[mp\_pose.PoseLandmark.LEFT\_SHOULDER.value].

*‹→*x,landmarks[mp\_pose.PoseLandmark.LEFT\_SHOULDER.value].y]

elbow = [landmarks[mp\_pose.PoseLandmark.LEFT\_ELBOW.value].

*‹→*x,landmarks[mp\_pose.PoseLandmark.LEFT\_ELBOW.value].y]

wrist = [landmarks[mp\_pose.PoseLandmark.LEFT\_WRIST.value].

*‹→*x,landmarks[mp\_pose.PoseLandmark.LEFT\_WRIST.value].y]

*# Calculate angle*

angle = calculate\_angle(shoulder, elbow, wrist)

*# Visualize angle*

cv2.putText(image, str(angle),

tuple(np.multiply(elbow, [640, 480]).astype(int)),

cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, (255, 255, 255), 2,␣

*‹→*cv2.LINE\_AA

)

**except**:

**pass**

*# Render detections*

mp\_drawing.draw\_landmarks(image, results.pose\_landmarks, mp\_pose.

*‹→*POSE\_CONNECTIONS,

mp\_drawing.DrawingSpec(color=(245,117,66),␣

*‹→*thickness=2, circle\_radius=2),

mp\_drawing.DrawingSpec(color=(245,66,230),␣

*‹→*thickness=2, circle\_radius=2)

)

cv2.imshow('Mediapipe Feed', image)

**if** cv2.waitKey(10) & 0xFF == ord('q'):

**break**

cap.release() cv2.destroyAllWindows()

[39]:

cap = cv2.VideoCapture(0)

*# Curl counter variables*

counter = 0 stage = **None**

*## Setup mediapipe instance*

**with** mp\_pose.Pose(min\_detection\_confidence=0.5, min\_tracking\_confidence=0.5) **as**␣

*‹→*pose:

# Curl Counter

**while** cap.isOpened():

ret, frame = cap.read()

*# Recolor image to RGB*

image = cv2.cvtColor(frame, cv2.COLOR\_BGR2RGB) image.flags.writeable = **False**

*# Make detection*

results = pose.process(image)

*# Recolor back to BGR*

image.flags.writeable = **True**

image = cv2.cvtColor(image, cv2.COLOR\_RGB2BGR)

*# Extract landmarks*

**try**:

landmarks = results.pose\_landmarks.landmark

*# Get coordinates*

shoulder = [landmarks[mp\_pose.PoseLandmark.LEFT\_SHOULDER.value].

*‹→*x,landmarks[mp\_pose.PoseLandmark.LEFT\_SHOULDER.value].y]

elbow = [landmarks[mp\_pose.PoseLandmark.LEFT\_ELBOW.value].

*‹→*x,landmarks[mp\_pose.PoseLandmark.LEFT\_ELBOW.value].y]

wrist = [landmarks[mp\_pose.PoseLandmark.LEFT\_WRIST.value].

*‹→*x,landmarks[mp\_pose.PoseLandmark.LEFT\_WRIST.value].y]

*# Get coordinates*

shoulder = [landmarks[mp\_pose.PoseLandmark.LEFT\_SHOULDER.value].

*‹→*x,landmarks[mp\_pose.PoseLandmark.LEFT\_SHOULDER.value].y]

elbow = [landmarks[mp\_pose.PoseLandmark.LEFT\_ELBOW.value].

*‹→*x,landmarks[mp\_pose.PoseLandmark.LEFT\_ELBOW.value].y]

wrist = [landmarks[mp\_pose.PoseLandmark.LEFT\_WRIST.value].

*‹→*x,landmarks[mp\_pose.PoseLandmark.LEFT\_WRIST.value].y]

*# Calculate angle*

angle = calculate\_angle(shoulder, elbow, wrist)

*# Visualize angle*

cv2.putText(image, str(angle),

tuple(np.multiply(elbow, [640, 480]).astype(int)),

cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, (255, 255, 255), 2,␣

*‹→*cv2.LINE\_AA

)

*# Curl counter logic*

**if** angle > 160:

stage = "down"

**if** angle < 30 **and** stage =='down': stage="up"

counter +=1 print(counter)

**except**:

**pass**

*# Render curl counter # Setup status box*

cv2.rectangle(image, (0,0), (225,73), (245,117,16), -1)

*# Rep data*

cv2.putText(image, 'REPS', (15,12),

cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, (0,0,0), 1, cv2.LINE\_AA)

cv2.putText(image, str(counter), (10,60),

cv2.FONT\_HERSHEY\_SIMPLEX, 2, (255,255,255), 2, cv2.LINE\_AA)

*# Stage data*

cv2.putText(image, 'STAGE', (65,12),

cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, (0,0,0), 1, cv2.LINE\_AA)

cv2.putText(image, stage,

(60,60),

cv2.FONT\_HERSHEY\_SIMPLEX, 2, (255,255,255), 2, cv2.LINE\_AA)

*# Render detections*

mp\_drawing.draw\_landmarks(image, results.pose\_landmarks, mp\_pose.

*‹→*POSE\_CONNECTIONS,

mp\_drawing.DrawingSpec(color=(245,117,66),␣

*‹→*thickness=2, circle\_radius=2),

mp\_drawing.DrawingSpec(color=(245,66,230),␣

*‹→*thickness=2, circle\_radius=2)

)

cv2.imshow('Mediapipe Feed', image)

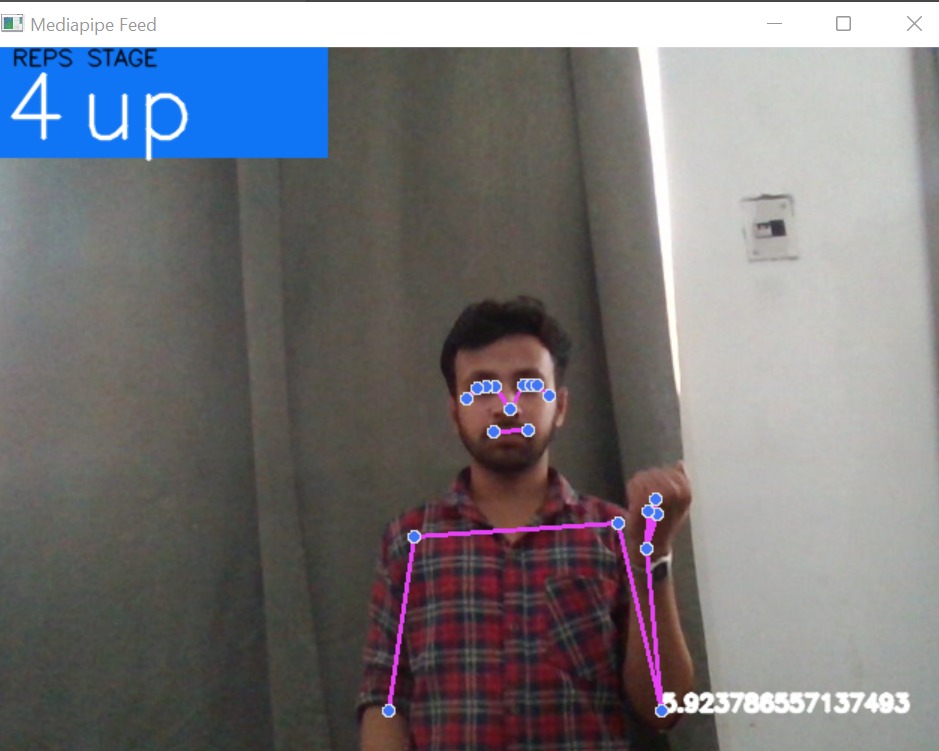
**if** cv2.waitKey(10) & 0xFF == ord('q'):

**break**

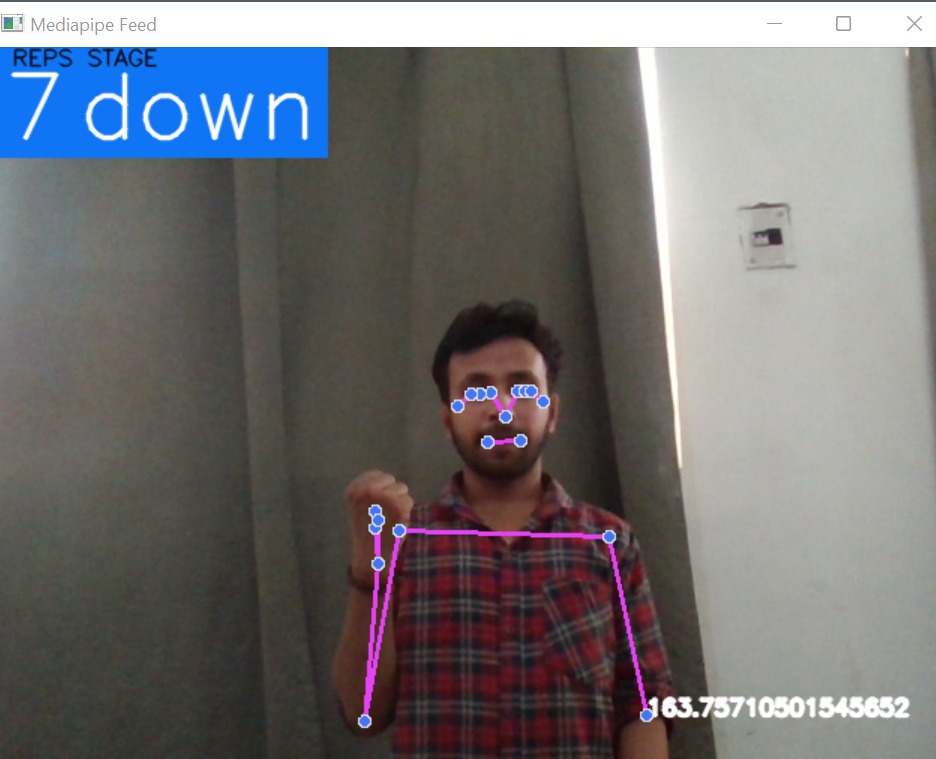
cap.release() cv2.destroyAllWindows()

# 

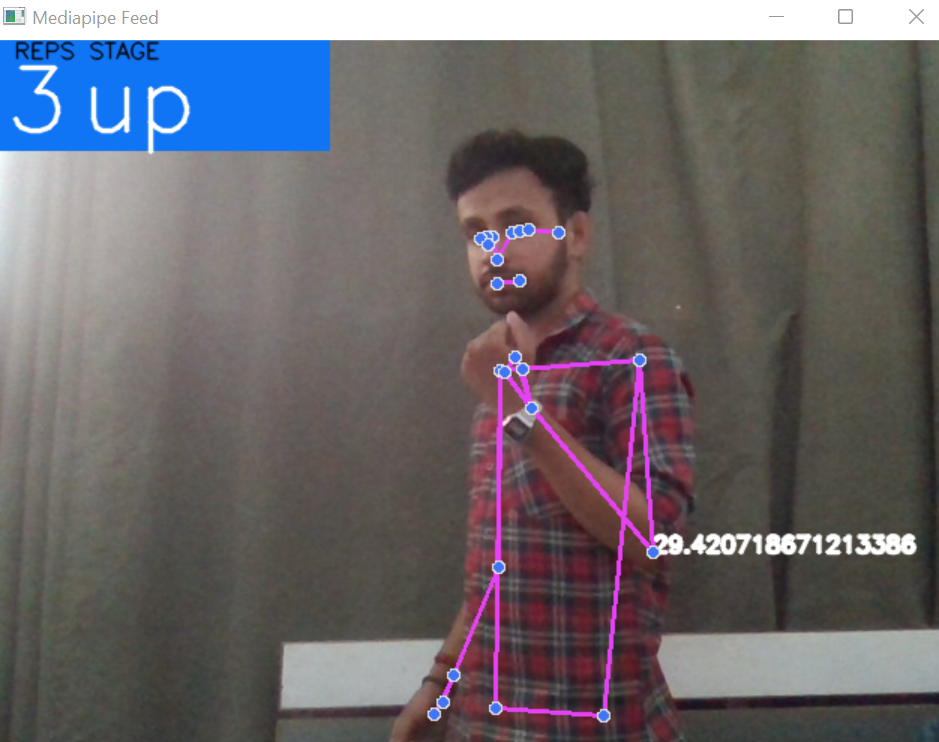
# CHAPTER-6 SCREENSHOTS

****

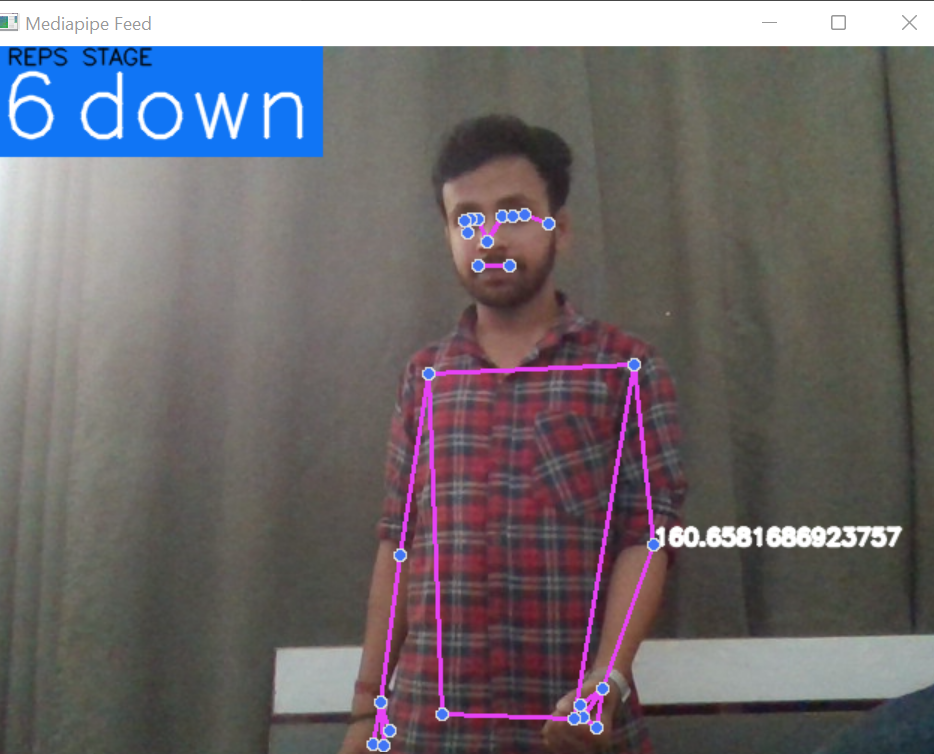
**Fig 6.1 [** Straight body with left hand up ]



**Fig 6.2 [**Straight body with left hand down]

****

**Fig 6.3 [** Left hand up**]**

****

**Fig 6.4 [** Left hand down **]**

Table

Description automatically generated

**Fig 6.5 [Terminal Output]**

**CHAPTER 7**

**TESTING**

* **Testing :-** Testing is the process of executing a program to find errors. To make our software perform well it should be error-free. If testing is done successfully it will remove all the errors from the software.
* **Principles of Testing :–**
* All the tests should meet the customer requirements
* To make our software testing should be performed by a third party
* All the tests to be conducted should be planned before implementing it
* Start testing with small parts and extend it to large parts.
* **Types of Testing :–**
* Unit Testing
* Integration Testing
* Regression Testing
* Smoke Testing
* Alpha Testing
* Beta Testing
* System Testing
* Stress Testing
* Performance Testing
* **Acceptance Testing :-** Acceptance Testing is a method of software testing where a system is tested for acceptability. The major aim of this test is to evaluate the compliance of the system with the business requirements and assess whether it is acceptable for delivery or not.

**Testing Screenshot :-**

* **Terminal Output :-**

Table

Description automatically generated

**Fig 7.1**

* **Testing for counter and angle :-**

**A picture containing text, person

Description automatically generated**

**Fig 7.2**

**CHAPTER-8**

**Conclusion and Future scope**

# Conclusion -

* At last the Conclusion of project is to build a model which is used to count the number of curls on the basis of Human Pose Estimation Model by using Mediapipe and OpenCV.

**Future scope –**

* We can implement this project in gym to make them digitalize (People can easily start doing exercise just by tapping on device).
* We’ll also be adding a function where the proper angle that we choose is recorded in the software with the proper database of the trainee.
* We can implement this in upcoming smart device with that user can do exercise by staying at their home.
* Add Deadlift and many other exercises.

**CHAPTER-9**

**REFERENCE AND BIBLIOGRAPHY**

* **Curl Tracker** - Samhitha, G., Rao, D.S., Rupa, C., Ekshitha, Y. and Jaswanthi, R., 2021, September. Vyayam: Artificial Intelligence based Bicep Curl Workout Tacking System. In *2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)* (pp. 1-5). IEEE.
* **Mediapipe -** Chunduru, V., Roy, M., & Chittawadigi, R. G. (2021). Hand tracking in 3d space using mediapipe and pnp method for intuitive control of virtual globe. In *2021 IEEE 9th Region 10 Humanitarian Technology Conference (R10-HTC)* (pp. 1-6). IEEE.
* **OPEN CV -** Culjak, I., Abram, D., Pribanic, T., Dzapo, H., & Cifrek, M. (2012, May). A brief introduction to OpenCV. In *2012 proceedings of the 35th international convention MIPRO* (pp. 1725-1730). IEEE.
* **AI -** Winston, P. H. (1992). *Artificial intelligence*. Addison-Wesley Longman Publishing Co., Inc..
* **Jupyter Notebook -** Perkel, J. M. (2018). Why Jupyter is data scientists' computational notebook of choice. *Nature*, *563*(7732), 145-147
* **Python -** Subasi, A. (2020). *Practical Machine Learning for Data Analysis Using Python*. Academic Press.
* **Pose Landmark Model -** Zhu, X., & Ramanan, D. (2012, June). Face detection, pose estimation, and landmark localization in the wild. In *2012 IEEE conference on computer vision and pattern recognition* (pp. 2879-2886). IEEE.
* **Numpy -** McKinney, W., 2012. *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython*. " O'Reilly Media, Inc.".
* **Hand Tracking Algorithm -** Pan, Z., Li, Y., Zhang, M., Sun, C., Guo, K., Tang, X., & Zhou, S. Z. (2010, March). A real-time multi-cue hand tracking algorithm based on computer vision. In *2010 IEEE Virtual Reality Conference (VR)* (pp. 219-222). Ieee.
* **ML -** Mitchell, T. M., & Mitchell, T. M. (1997). *Machine learning* (Vol. 1, No. 9). New York: McGraw-hill.