

# POSTURE ANALYSIS FOR GYM WORKOUT

**INT 400 – INTERNSHIP PROJECT REPORT**

***Submitted by***

**AAKASH S - E0120005**

***In partial fulfilment for the award of the degree of***

**BACHELOR OF TECHNOLOGY**

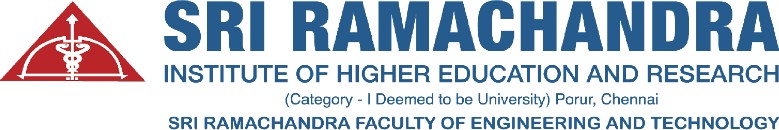
**in**

**COMPUTER SCIENCE AND ENGINEERING**

**(Artificial Intelligence & Machine Learning) Sri Ramachandra Engineering and Technology**

**Sri Ramachandra Institute of Higher Education and Research, Porur, Chennai - 600116**

**DECEMBER, 2022**



# POSTURE ANALYSIS FOR GYM WORKOUT

**INT400 INTERNSHIP PROJECT REPORT**

***Submitted by***

**AAKASH S - E0120005**

***In partial fulfilment for the award of the degree of***

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE AND ENGINEERING**

**(Artificial Intelligence & Machine Learning) Sri Ramachandra Engineering and Technology**

**Sri Ramachandra Institute of Higher Education and Research, Porur, Chennai - 600116**

**DECEMBER, 2022**



**BONAFIDE CERTIFICATE**

Certified that this project report POSTURE ANALYSIS FOR GYM WORKOUT is the bonafide record of work done by **AAKASH S** who carried out the internship work under my supervision.

|  |  |
| --- | --- |
| **Signature of Supervisor** | **Signature of Vice-Principal** |
| **Prof. Ramya** | **Prof. M. Prema** |
| Lecturer, | Vice-Principal, |
| Department of Computer Science and Engineering | Department of Computer Science and Engineering |
| Sri Ramachandra Faculty of Engineering and Technology | Sri Ramachandra Faculty of Engineering and Technology |
| SRIHER, Porur, Chennai - 600116 | SRIHER, Porur, Chennai - 600116 |

**Evaluation Date:**



# ACKNOWLEDGEMENT

I express my sincere gratitude to our Chancellor, Vice-Chancellor and our sincere gratitude to our Provost **Dr. V.Raju** and our Vice-Principal **Prof.M.Prema** for their support and for providing the required facilities for carrying out this study.

I wish to thank my faculty supervisor, **Prof. Ramya**, Department of Computer Science and Engineering, Sri Ramachandra faculty of Engineering and Technology for extending help and encouragement throughout the project. Without their continuous guidance and persistent help, this project would not have been a success for me.

I am grateful to all the members of Sri Ramachandra Faculty of Engineering and Technology, my beloved parents and friends for extending the support, who helped us to overcome obstacles in the study.

## TABLE OF CONTENT

|  |  |  |  |
| --- | --- | --- | --- |
| **Chapter** | **Title** |  | **P.NO** |
|  | ABSTRACT |  | 6 |
|  | LIST OF FIGURES | | 7 |
|  | LIST OF ABBREVIATIONS | | 8 |
| 1 | INTRODUCTION | | 9 |
|  | 1.1 | Literature Review | 10 |
|  | 1.2 | Human Pose Estimation | 11 |
|  | 1.3 | Machine Learning | 11 |
| 2 | OBJECTIVE |  | 12 |
|  | 2.1 | Aim | 12 |
|  | 2.2 | Problem Statement | 12 |
| 3 | WORKFLOW |  | 13 |
|  | 3.1 | RepCount Work flow | 13 |
|  | 3.2 | Posture Work flow | 13 |
| 4 | TOOLS AND TECHNOLOGIES | | 16 |
|  | 4.1 | OpenCV | 16 |
|  | 4.2 | MediaPipe | 17 |
|  | 4.3 | Python | 17 |
| 5 | IMPLEMENTATION | | 19 |
|  | 5.1 | Video Detection & Training | 19 |
|  | 5.2 | Extracting Coordinates | 20 |
|  | 5.3 | Data Collection & Preprocessing | 21 |
|  | 5.4 | Model Training & Evaluation | 23 |
|  | 5.5 | Model Deployment | 24 |
| 6 | RESULTS & DISCUSSION | | 25 |
|  | 6.1 | Deadlift - Concentric and Eccentric phases | 25 |
|  | 6.2 | Posture Detection: Left grip and Right grip | 26 |
|  | 6.3 | Posture Detection - Hip positioning | 27 |
|  | 6.4 | Model Metrics | 28 |
| 7 | CONCLUSION |  | 29 |
|  | REFERENCES |  | 30 |
|  |  |  |  |
|  |  |  |  |

**ABSTRACT**

Everyone benefits from exercise and physical activity. Staying active can benefit you in a variety of ways, regardless of your health or physical abilities. Exercises are often practiced in training centers, through personal tutors, and may also be learned on one’s own with the help of the recorded clips, etc. In fast-paced lifestyles, many people prefer self-learning because the above-mentioned resources might not be available all the time. Mostly incorrect postures often occurred in self learning forms. One's health might suffer from improper posture, which can cause both short-term acute discomfort and long-term chronic problems. We investigated many applications that can be implemented using data provided by a pre-trained posture estimation model called MediaPipe. Applications include motion capture, gait analysis, sign language detection etc. Building a gym posture monitoring system, which analyses and tracks user motions and postures for faults, the major goal of this project is to reduce one-sided training load using the implementation of proper exercise forms to reduce myofascial imbalances in the musculoskeletal system and core stability. Rep Count analyzes videos/images of the deadlift and scores the posture of the person performing the exercise from a range of 0 to 1.

|  |  |  |
| --- | --- | --- |
|  | **LIST OF FIGURES** |  |
| **Figure No.** | **Title** | **Page** |
| Fig 3.1 | RepCount Workflow | 13 |
| Fig 3.2 | Posture Work flow |  |
| Fig 4.1 | OpenCV Framework | 16 |
| Fig 4.2 | Mediapipe | 17 |
| Fig 4.3 | Python | 17 |
| Fig 5.1 | Video Detection & Training | 19 |
| Fig 5.2 | Extracting Coordinates | 20 |
| Fig 5.3 | Pose Landmark Model | 21 |
| Fig 5.4 | Data Creation | 21 |
| Fig 5.5 | Data Preprocessing | 22 |
| Fig 5.6 | Model Evaluation | 23 |
| Fig 5.7 | Model Deployment | 24 |
| Fig 6.1 | Concentric and Eccentric phases | 25 |
| Fig 6.2 | Left grip and Right grip | 26 |
| Fig 6.3 | Hip positioning | 27 |
| Fig 6.4 | Model Training & Results | 28 |

## LIST OF ABBREVIATIONS

|  |  |  |
| --- | --- | --- |
| **S.NO** | **ABBREVIATION** | **EXPANSION** |
| 1. | ML | Machine Learning |
| 2. | AI | Artificial Intelligence |
| 3. | OpenCV | Open Source Computer Vision |
| 4. | UI | User Interface |

**CHAPTER-1 INTRODUCTION**

Development in the field of fitness and well-being of health has grown exponentially in the last decade which includes fit-bands, Calorie counter, diet planner, run tracker, etc. Supporting the advancement in this field, we focused on the problem of getting assistance while doing exercises and focusing on the prevention of injuries. Measuring the diurnal exercises of body composition, resting heart rate and blood pressure, knowledge of physical exertion actions, and perception of heartiness were used to determine the impact of the fitness trackers.

Postural Analysis can be defined as the evaluation of various body parts of an individual in a position in an integrated manner in relation to ideal posture. Posture can be assessed in static as well as while performing sporting activities (dynamic) with modern techniques. Posture analysis is an important tool that helps assess the reasons behind various sports injuries and life-style related problems. The deviation from normal posture puts stress on structures that maintain the normal alignment of the neck, back, upper and lower limbs.

Losing track of repetitions is a common problem with weightlifters. The problem arises when all the attention is focused towards achieving correct form and not towards the number of reps that have been completed. This project plans to solve this problem, as well as eliminate the notepad and paper most serious weightlifters use to track their repetitions and the weights.

### Literature Review

There are numerous applications available in the market which guide the user about the exercises to be performed. But through our application, we not only guide the user regarding which exercise to perform but also about the correct posture and counting the repetitions using computer vision. The application can not only be used by individuals at homes but by increasing the scope can be used in gyms as smart trainers thus reducing the human intervention.

* + - Robust vision- based workout analysis. Test results show the prevalence of their proposed 3D posture assessment over the past ones. It identifies incorrect motions but not along with the timing of these motions and hence does not provide timely feedback to users. They could not integrate their model with video tutorials.
    - Models for Human Pose Estimation and worked on Home-based Physical Therapy with an Interactive Computer Vision System. They could not provide users a side-view option and could not develop an algorithm that gives more detailed feedback on how the patient is doing i.e Instead of giving feedback based on the overall performance.
    - Incorporated computer vision strategies and proposed system that examines the practitioner’s stance from both front and side perspectives by separating the body shape, skeleton, dominant axes, and points. Improving or even redesigning the methods of feature point detection and assistant axis generation for some poses, can make the system more solid.
    - Various classification techniques to detect yoga poses out of which random forest classifiers gave the best results. It detects different Title Suppressed Due to Excessive Length 3 yoga poses using this application along with identification of the pose, if the accuracy is calculated then the user will be able to track and improve performance.

#### Human Pose Estimation

Human pose estimation is a computer vision-based technology that detects and analyzes human posture. The main component of human pose estimation is the modeling of the human body. There are three of the most used types of human body models: skeleton-based model, contour-based, and volume-based.

* Skeleton-based model consists of a set of joints (key points) like ankles, knees, shoulders, elbows, wrists, and limb orientations comprising the skeletal structure of a human body.
* Contour-based model consists of the contour and rough width of the body torso and limbs, where body parts are presented with boundaries and rectangles of a person’s silhouette.
* Volume-based model consists of 3D human body shapes and poses represented by volume-based models with geometric meshes and shapes, normally captured with 3D scans.

#### 1.3 Machine Learning

Machine learning (ML) is a field of inquiry devoted to understanding and building methods that 'learn', that is, methods that leverage data to improve performance on some set of tasks. It is seen as a part of [artificial intelligence](https://en.wikipedia.org/wiki/Artificial_intelligence).

Machine learning algorithms build a model based on sample data, known as [training data](https://en.wikipedia.org/wiki/Training_data), in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as in medicine, [email filtering](https://en.wikipedia.org/wiki/Email_filtering), [speech recognition](https://en.wikipedia.org/wiki/Speech_recognition), [agriculture](https://en.wikipedia.org/wiki/Agriculture), and [computer vision](https://en.wikipedia.org/wiki/Computer_vision), where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

## CHAPTER-2 OBJECTIVE

#### Aim

Currently there are mobile applications for smartphones that calculate the number of reps for certain exercises. Moreover, these applications are not very accurate and record false positives for a rep-count. This system helps  the user to correct the set of common exercise like deadlift, bench press, etc.  The objective of this project is to develop a model that will count the number of reps for a weightlifter, determine the number of correct and incorrect reps.

#### Problem Statement

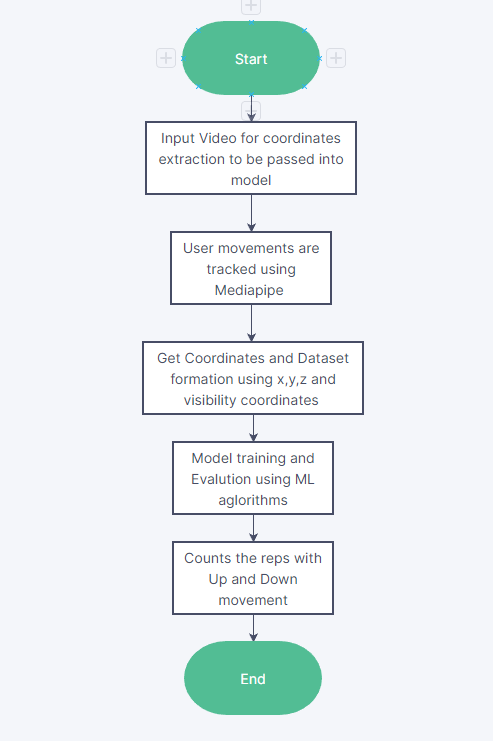
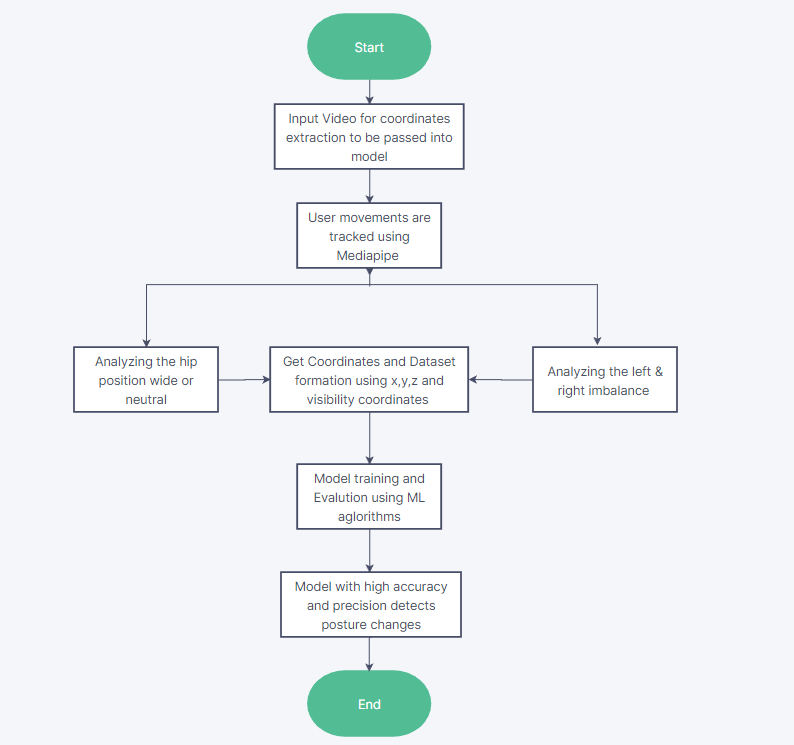
Building a gym posture monitoring system, which analyses and tracks user motions and postures for faults, the major goal of this project is to reduce one-sided training load using the implementation of proper exercise forms  to reduce myofascial imbalances in the musculoskeletal system and core stability.

This project analyzes videos/images of the deadlift (one of the most fundamental weightlifting exercises) and scores the posture of the person performing the deadlift from a range of 0 to 1.

## CHAPTER-3 WORKFLOW

### 3.1 Project Workflow

The application’s overall workflow is illustrated below:

**Figure 3.1 - RepCount Work flow Figure 3.2 - Posture Work flow**

### This project analyzes video of the deadlift (one of the most fundamental weightlifting

### exercises) and scores the posture of the person performing the deadlift from a range

### of 0 to 1.

The video is taken as input using a camera which can be either a smartphone camera which is now ubiquitously available or we can use a webcam. Camera acts as the input component of the system. Camera is used for image acquisition and as data input to the model.

Firstly, one must work on getting the source video to be passed on the model and openCV detects the video using videocapture function. Then the model works on video training and gets the height, width and fps .

**DATA SET**

The system must read the input video uploaded by the user as a set of coordinates in the landmarks of x,y & z. Selected coordinates of the movement in deadlift gets stored in csv file with a visibility mark. There are ‘n’ number of datasets are generated by extracting coordinates.

**POSE ESTIMATION**

The list of coordinates in the X,Y and Z axis for 33 major key-points of a person. These key points define the location of each major body part in the input image. Using these key points we can build an accurate skeletal orientation of the user. So, three datasets formed individually from deadlift movement from Up - Down , Weight imbalance in Left - Right and Hips correction using same preprocessing and model training.

**TRACKING MODEL**

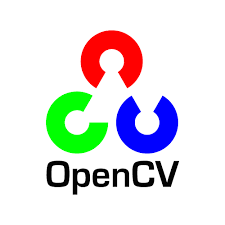
All the key points having 3 degrees of freedom each x, y, z location and visibility and in our model we used a regression & classifier approach. The dataset gets normalised using standscalar and various ML algorithms like logistic regression, ridge classifier, random forest classifier and gradient boosting classifier are used to train the model.

**REPETITION COUNTING**

To count the repetitions, the algorithm monitors the probability of a target pose class. When the probability of the “down” pose class passes a certain threshold for the first time, counter increases by one. Similarly, position signify the correct body posture using hips and arm movement. Then successful transition of hand from down or up increases counter.

## CHAPTER 4 TOOLS AND TECHNOLOGIES

### OpenCV



**Fig 4.1 – OpenCV Framework**

OpenCV is the huge open-source library for the computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today’s systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When it integrated with various libraries, such as NumPy, python is capable of processing the OpenCV array structure for analysis. To Identify image pattern and its various features we use vector space and perform mathematical operations on these features.

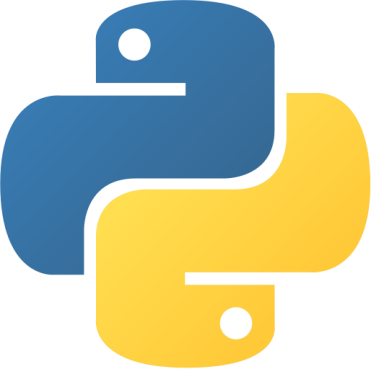
## MediaPipe



**Fig 4.2 - MediaPipe**

MediaPipe is an open-source framework for building pipelines to perform computer vision inference over arbitrary sensory data such as video or audio. Using MediaPipe, such a perception pipeline can be built as a graph of modular components. It also facilitates the deployment of computer vision applications into demos and applications on different hardware platforms.

### Python



**Fig 4.3 - Python**

Python is now one of the most popular and widely used programming languages in the world. Besides web and software development, Python is used for data analytics, machine learning, and even design

## CHAPTER 5 IMPLEMENTATION

### 5.1 Video Detection :

The input video is taken as input using a camera which can be either a smartphone camera which is now ubiquitously available or we can use a webcam. Camera acts as the input component of the system. Camera is used for image acquisition and as data input to the model. Initial step follows detection and analysis of videos of the deadlift (one of the most fundamental weightlifting exercises).

This process can be simplified into a few main parts as

Analyze videos frame by frame (for high fps videos we can analyze every X frames since frames will be very repetitive and tune X accordingly to satisfy our needs). We will use the OpenCV object VideoWriter.

For each frame we score the exercise posture



**Fig 5.1 – Video Detection (code)**

**5.2 Extraction of Coordinates:**

For each frame we score the exercise posture.

First we feed our image into an open source Open Pose key point detection model for human pose detection. This gives us locations of specific joints (right shoulder, left elbow, etc) and then using the locations of these joints we can feed this data to a classification model to predict whether the performer of the exercise is in correct posture or not. All the key points having 3 degrees of freedom each x, y location and visibility is predicted using human pose estimation model and in our model we used a regression approach.

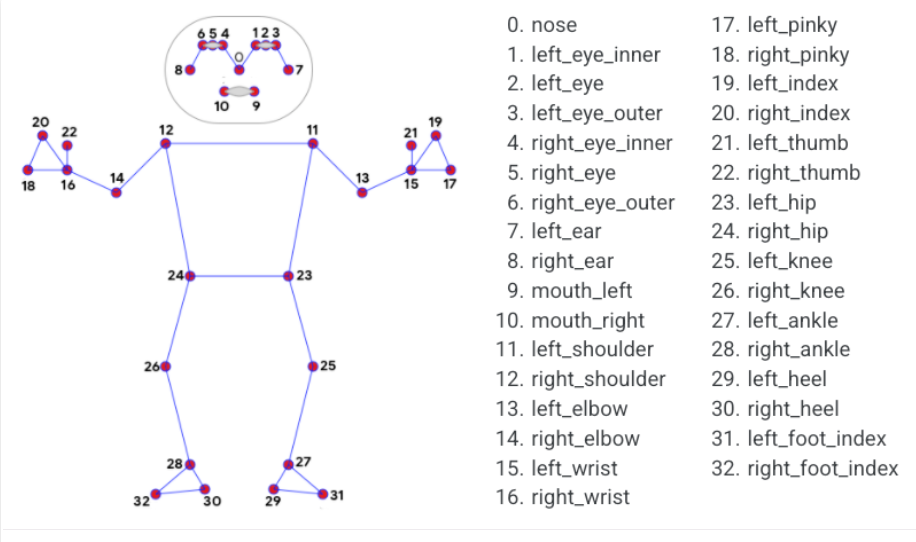
For Concentric and Eccentric phases - shoulder points and knee points are utilized

For Weight Balance Movement: Shoulder points and elbow points utilized.

For Hip Correction: Both Hip and Knee points utilized.



**Fig 5.2 – Coordinates Extraction (code)**



**Fig 5.3 - Pose Landmark Model**

### 5.3 Data Collection & Pre-processing:

**5.3.1 Data Creation:**

Gym poses of professional trainer need to be feeded to build our system that can identify ideal values of the factors involved in the exercises. The system must read the input video uploaded by the user as a set of coordinates in the landmarks of x,y & z. Selected coordinates of the movement in deadlift gets stored in csv file with a visibility mark. There are ‘n’ number of datasets are generated by extracting coordinates.



**Fig 5.4 – Dataset Creation (code)**

**5.3.2 Data Pre-processing:**

Preprocessing data is required in nearly every application. For that reason, in scikit-learn we use Pipeline class which allows you to chain preprocessing steps and models together. We implementing Standard Scalar into the pipeline to directly feed processed data to regression and classifiers. Standard Scalar provides scaling technique where the values are centered around the mean with a unit standard deviation. This means the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation.

Initial Step: The data are split into TRAINING data and TEST data according to the cv parameter.

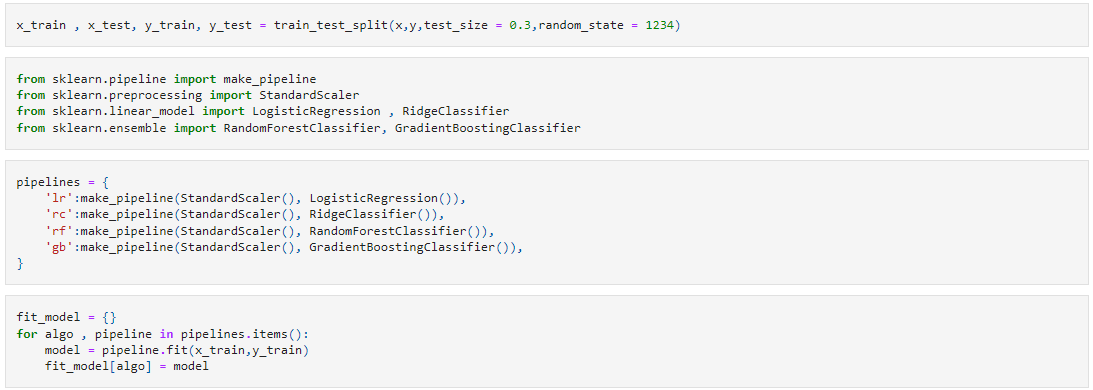
Step 1: the scaler is fitted on the TRAINING data

Step 2: the scaler transforms TRAINING data

Step 3: the models are fitted/trained using the transformed TRAINING data

Step 4: the scaler is used to transform the TEST data

Step 5: the trained models predict using the transformed TEST data



**Fig 5.5 – Data Preprocessing (code)**

### 5.4 Model Training & Evaluation:

Machine learning algorithms can be used to estimate the number of repetitions (reps) performed during a deadlift exercise.

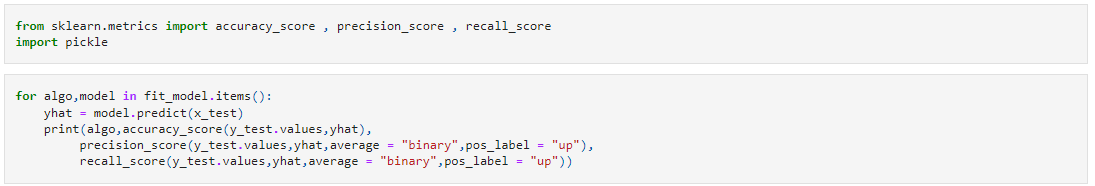
In regression algorithm to predict the number of reps based on features extracted from the movement data, such as joints & distances between joints. Regression used to model the relationship between the features and the target variable (number of reps).

In classification algorithm to predict the end of each repetition, and then count the number of repetitions based on the predicted end points. Logistic Regression, Ridge Classifier, Random Forest Classifier, or Gradient Boosting Classifier being used to predict whether a given point in the movement data corresponds to the end of a repetition

.

In general, Random Forest Classifier and Gradient Boosting Classifier tend to perform well on a wide range of classification problems, but they can be more computationally expensive and may require more fine-tuning than Logistic Regression or Ridge Classifier

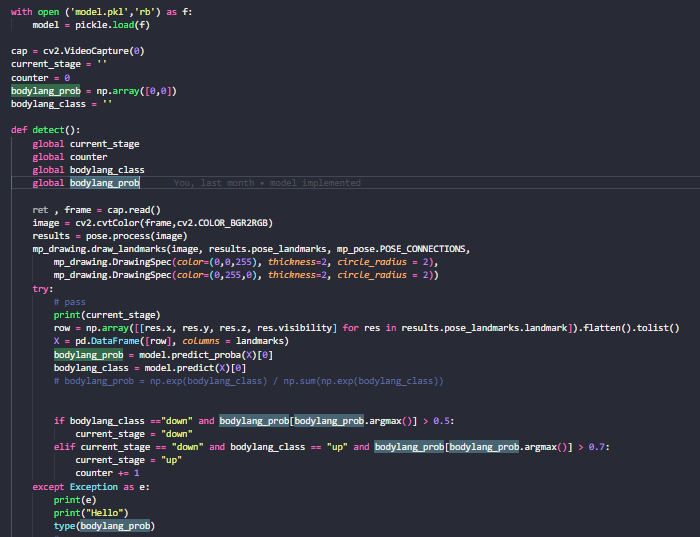
Evaluation their performance on the test data using metrics such as accuracy, precision, recall, and F1-score. The algorithm that performs best on the test data been selected as the final model for deployment.



**Fig 5.6 Model Evaluation (code)**

### 5.5 Model Deployment:

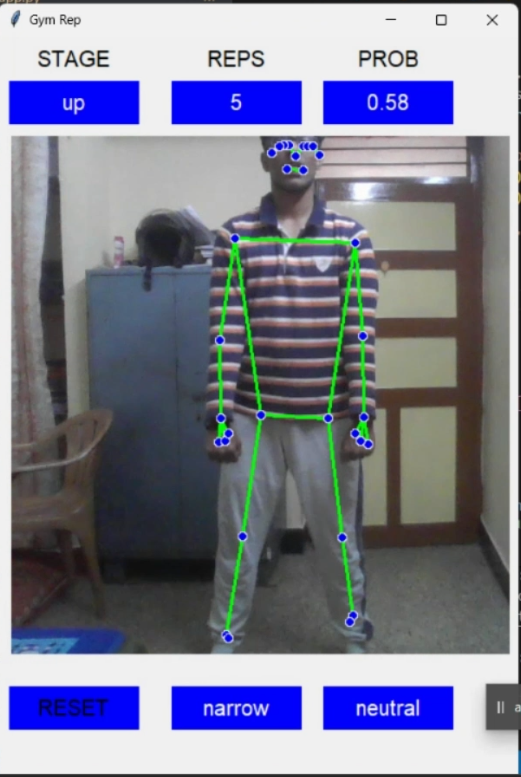
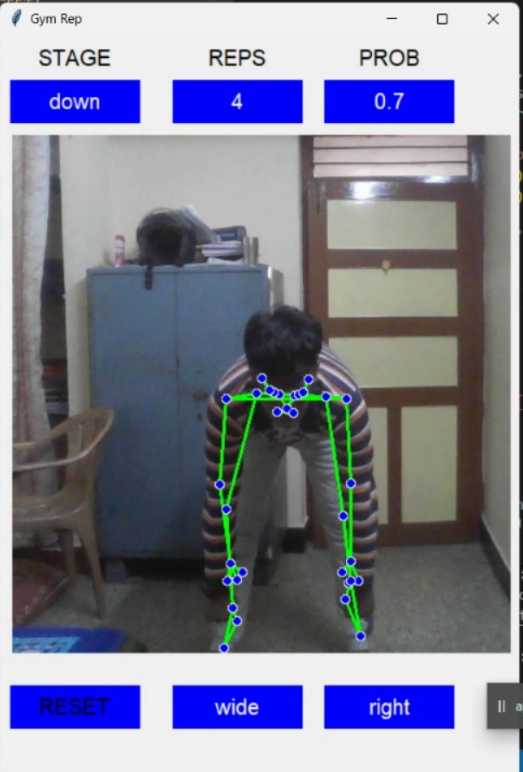
Deploy the trained model to run on a device or platform, such as a python or web-based application, to allow users to perform deadlift exercise recognition in real-time.



**Fig 5.7 Model Deployment (code)**

# CHAPTER -6 RESULTS AND DISCUSSION

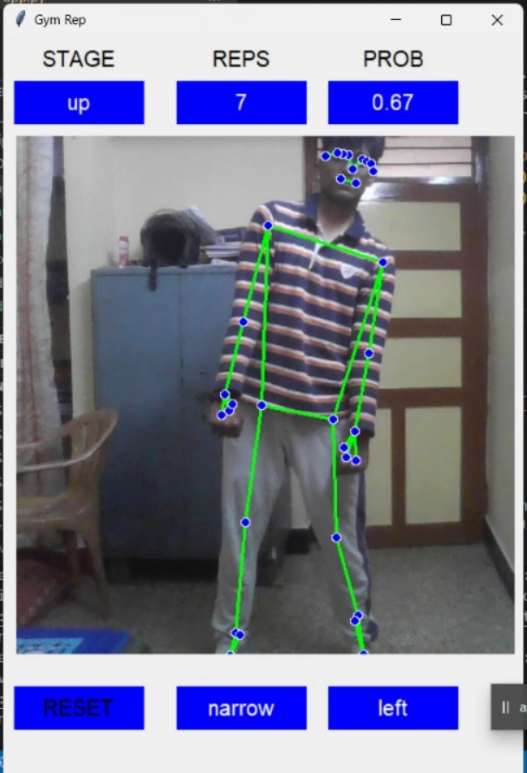
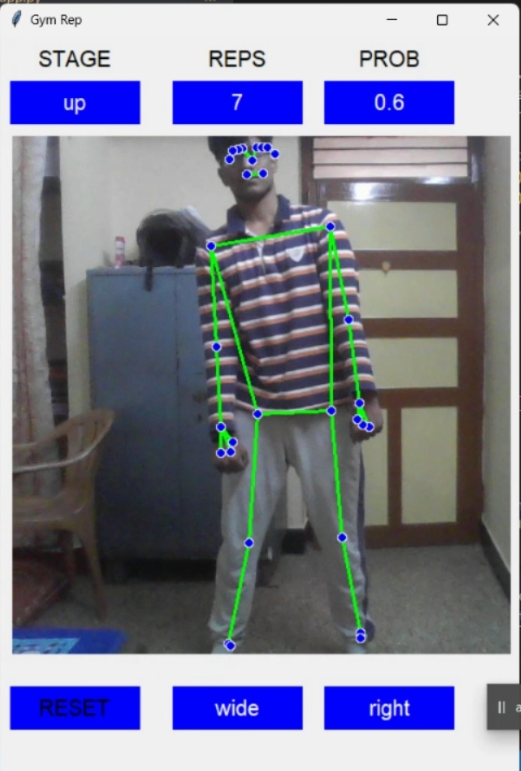
### Deadlift - Concentric and Eccentric phases

**** 

**Fig 6.1 – Concentric and Eccentric phases**

Fig 6.1 shows the up-and-down movement analysis, the goal is typically to understand the mechanics of the movement and identify any areas of inefficiency or risk of injury.

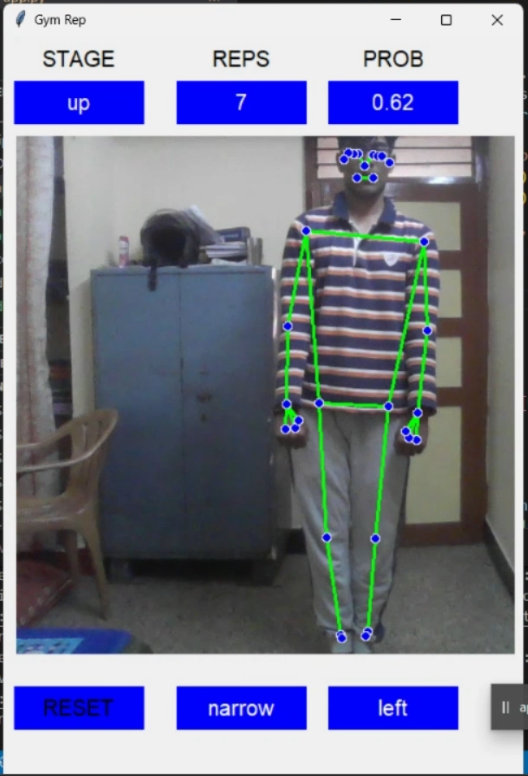
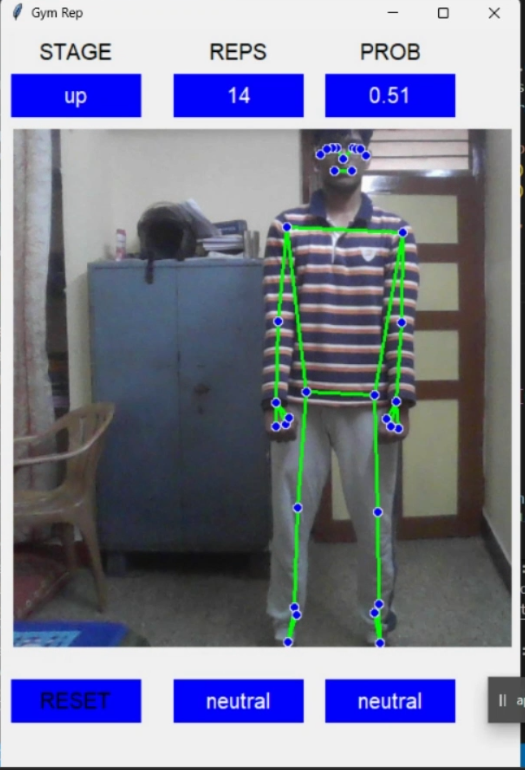
### Posture Detection: Left grip and Right grip

**Fig 6.2 – Left grip and Right grip**

Fig 6.2 shows a situation where the weight being lifted is unevenly distributed between the left and right sides of the body

### Posture Detection - Hip positioning

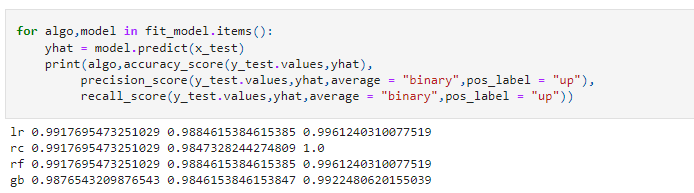
  

**Fig 6.3 – Hip positioning**

Fig 6.3 shows the narrow , neutral and wide positions of hip during deadlift.

### Model Metrics

Apparently, Random Forest Classifier dominates over the Gradient boosting classifier with an accuracy of 78% . Ridge Classifier and Logistic Regression are simpler algorithms that can be faster to train and easier to interpret, but may not perform as well on complex datasets.



**Fig 6.4 –Model Training & Results**

Fig 6.4 shows the model's performance using appropriate metrics, such as accuracy, precision, recall, and F1-score, on the test set.

# CHAPTER -7 CONCLUSION

In conclusion, machine learning algorithms, such as classifiers, can be used to count repetitions and detect posture in a deadlift workout. These algorithms can analyze the data collected from sensors or cameras and determine the number of repetitions performed and the posture of the lifter during each repetition.

The accuracy and precision of these algorithms will depend on factors such as the quality of the data, the type of algorithm used, and the training process. For example, algorithms such as Random Forest Classifier and Gradient Boosting Classifier have shown high levels of accuracy in classification tasks and may perform well in deadlift posture detection.

By using machine learning algorithms to monitor and track deadlift workouts, individuals can receive valuable feedback on their form and technique, allowing them to make improvements and reduce the risk of injury. Additionally, this technology can also provide trainers and coaches with useful insights into their clients' training, allowing them to make informed decisions about programming and coaching.

Overall, the use of machine learning algorithms for counting repetitions and posture detection in deadlift workouts has the potential to revolutionize the way individuals and trainers approach strength training and performance tracking.

## REFERENCES

1. https://developer.twitter.com/en/portal/products
2. https:[//www.analy](http://www.analyticsvidhya.com/blog/2016/08/beginners-guide-to-)t[ics](http://www.analyticsvidhya.com/blog/2016/08/beginners-guide-to-)v[idhya.com/blog/2016/08/beginners-guide-to-](http://www.analyticsvidhya.com/blog/2016/08/beginners-guide-to-) topic
3. https://campus.datacamp.com/courses/analysing-social-media-data-in-python/
4. https://github.com/networkx/networkx
5. https://docs.streamlit.io/
6. https://colah.github.io/posts/2015-08-Understanding-LSTMs/
7. https://towardsdatascience.com/fake-news-classifier-e061b339ad6c
8. https://spark.apache.org/docs/latest/graphx-programming-guide.html

# APPENDICES WORKLOG

|  |  |  |
| --- | --- | --- |
| DAY | DATE | TASK |
| Day 1 | 01-11-2021 | Learning basics of NLP |
| Day 2 | 02-11-2021 | Basic pre-processing of text data |
| Day 3 | 03-11-2021 | lemmatization |
| Day 4 | 04-11-2021 | stemming |
| Day 5 | 05-11-2021 | Intuition of bag of words, TFIDF |
| Day 6 | 06-11-2021 | word2vec |
| Day 7 | 07-11-2021 | Intuition behind bow |
| Day 8 | 08-11-2021 | Intuition behind TFIDF |
| Day 9 | 09-11-2021 | Intuition behind word2vec |
| Day 10 | 10-11-2021 | Intuition behind RNN |
| Day 11 | 11-11-2021 | Coding RNN |
| Day 12 | 12-11-2021 | Shift of RNN to LSTM |
| Day 13 | 13-11-2021 | Coding RNN |
| Day 14 | 14-11-2021 | Coding LSTM, GRU |
| Day 15 | 15-11-2021 | Detailed analysis about encoder decoder architecture |
| Day 16 | 16-11-2021 | Reading about image captioning |
| Day 17 | 17-11-2021 | Coding image captioning |
| Day 18 | 18-11-2021 | Reading about attention mechanism |
| Day 19 | 19-11-2021 | Coding about attention mechanism |

|  |  |  |
| --- | --- | --- |
| Day 20 | 20-11-2021 | Learning about transformers |
| Day 21 | 21-11-2021 | Coding transformers |
| Day 22 | 22-11-2021 | Gaining intuition about BERT |
| Day 23 | 23-11-2021 | CODING BERT |
| Day 24 | 24-11-2021 | Data Acquisition for covid vaccine side effects |
| Day 25 | 25-11-2021 | Data cleaning with text pre-processing |
| Day 26 | 26-11-2021 | data modelling with ML |
| Day 27 | 27-11-2021 | Using pycaret for choosing the best model |
| Day 28 | 28-11-2021 | SVM is chosen because it had an accuracy of  92% on test set |
| Day 29 | 29-11-2021 | Data modelling with deep learning |
| Day 30 | 30-11-2021 | Got RNN accuracy of 86% on test set |
| Day 31 | 01-12-2021 | Got LSTM ACCURACY OF 95% on test set |
| Day 32 | 02-12-2021 | got GRU accuracy of 90% |
| Day 33 | 03-12-2021 | Intuition behind RNN |
| Day 34 | 04-12-2021 | Coding RNN |
| Day 35 | 05-12-2021 | shift of RNN to LSTM |
| Day 36 | 06-12-2021 | Coding RNN |
| Day 37 | 07-12-2021 | Preparing scripts for deploying Ml model |
| Day 38 | 08-12-2021 | Creating a virtual Jupyter notebook on  created EC2 instance |
| Day 39 | 09-12-2021 | Facing error(Connection error) |
| Day 40 | 10-12-2021 | Facing error(Connection error) |
| Day 41 | 11-12-2021 | facing error(Connection error) |

|  |  |  |
| --- | --- | --- |
| Day 42 | 12-12-2021 | Error got solved |
| Day 43 | 13-12-2021 | Deployed the ml model on AWs Ec2 |
| Day 44 | 14-12-2021 | implementing Sentiment analysis |
| Day 45 | 15-12-2021 | Finished the tweet recommender system |
| Day 46 | 16-12-2021 | gaining insights about LDA |
| Day 47 | 17-12-2021 | gaining insights about sentiment classification |
| Day 48 | 18-12-2021 | Implementing LDA |
| Day 49 | 19-12-2021 | implementing Sentiment analysis |
| Day 50 | 20-12-2021 | Finished the tweet recommender system |
| Day 51 | 21-12-2021 | Finished the tweet recommender system |
| Day 52 | 22-12-2021 | Constructing the social graph |
| Day 53 | 23-12-2021 | unsupervised learning with LDA |
| Day 54 | 24-12-2021 | deploying the tweet analyser in heroku |
| Day 55 | 25-12-2021 | Construct the architecture of conversational Ai agent |
| Day 56 | 26-12-2021 | Transition of traditional architecture to end to end deep  learning architecture |
| Day 57 | 27-12-2021 | Conversational AI agent |
| Day 58 | 28-12-2021 | Traditional ASR models |
| Day 59 | 29-12-2021 | Gaining insights about Nemo Quartz net |
| Day 60 | 30-12-2021 | implementing Quartz net |
| Day 61 | 31-12-2021 | Dynamic input using pyaudio |
| Day 62 | 01-01-2022 | Implementing google search with query given |
| Day 63 | 02-01-2022 | Reading about selenium for scrapping the data |

|  |  |  |
| --- | --- | --- |
| Day 64 | 03-01-2022 | Selenium used to scrap data from the links |
| Day 65 | 04-01-2022 | Gaining insights about Text Summarization(T5) |
| Day 66 | 05-01-2022 | Conversational AI agent part2 |
| Day 67 | 06-01-2022 | Natural language understanding |
| Day 68 | 07-01-2022 | NLU higher deep architectures |
| Day 69 | 08-01-2022 | Selenium used to scrap data from the links |
| Day 70 | 09-01-2022 | Natural language understanding |
| Day 71 | 10-01-2022 | Learning about Transformers |
| Day 72 | 11-01-2022 | Studying about Text to speech |
| Day 73 | 12-01-2022 | Nemo TTS |
| Day 74 | 13-01-2022 | Learning about Transformers |
| Day 75 | 14-01-2022 | Learning about BERT |
| Day 76 | 15-01-2022 | End to end Conversational AI agent |
| Day 77 | 16-01-2022 | End to end Conversational AI agent |
| Day 78 | 17-01-2022 | Learning about Transformers |
| Day 79 | 18-01-2022 | Learning about BERT |
| Day 80 | 19-01-2022 | End to end Conversational AI agent |
| Day 81 | 20-01-2022 | End to end Conversational AI agent |
| Day 82 | 21-01-2022 | Learning about Transformers |
| Day 83 | 22-01-2022 | Learning about BERT |
| Day 84 | 23-01-2022 | End to end Conversational AI agent |
| Day 85 | 24-01-2022 | Learning about BERT |
| Day 86 | 25-01-2022 | Learning about Transformers |