# POSTURE CORRECTOR

A PROJECT REPORT

**21CSC305P –MACHINE LEARNING**

**(2021 Regulation)**

**III Year/ V Semester**

**Academic Year: 2024 -2025**

#### Submitted by

SAKSHAM MANN [RA2211003011213]

DEVISH MITTAL [RA2211003011227]

KARTIK MITTAL [RA2211003011230]

REETAM KOLE[RA2211003011231]

#### Under the Guidance of

## Dr. ASWATHY K CHERIAN

Assistant Professor

Department of Computing Technologies

### *Assistant Professor*

### *in partial fulfillment of the requirements* *for the degree of*

## BACHELOR OF TECHNOLOGY

## in

## COMPUTER SCIENCE AND ENGINEERING

## 

## SCHOOL OF COMPUTING

## COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

## KATTANKULATHUR- 603 203

### NOVEMBER 2024



# SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

# KATTANKULATHUR – 603 203

## BONAFIDE CERTIFICATE

Certified that **21CSC305P -** **MACHINE LEARNING project** reporttitled “Human Posture Analysis” is the bonafide work of “ **SAKSHAM MANN[RA2211003011213],aaDEVISHaMITTAL[RA2211003011227],KARTIK MITTAL[RA2211003011230], REETAM KOLE[RA2211003011231]”** who carried out the task of completing the project within the allotted time.

|  |  |
| --- | --- |
| **SIGNATURE**  Dr. Aswathy K Cherian  **Course Faculty**  **Assistant Professor**  Department of Computing Technologies  SRM Institute of Science and Technology  Kattankulathur | **SIGNATURE**  Dr. Niranjana G  **Head of the Department**  **Professor and Head**  Department of Computing Technologies  SRM Institute of Science and Technology Kattankulathur |

**ABSTRACT**

Fitness is becoming an important part of human life as it brings many benefits to personal health. However, exercises can also be ineffective and become dangerous if performed incorrectly by the performer. Proper form is important in any physical activity, but it is especially important in sports or workouts. Correct form can not only reduce your risk of injury, but also allow you to move efficiently, increase your performance, and use your full range of motion. In my project, I use machine learning to provide detailed analysis and recommendations for improving the form of exercise performers. In addition, Deep Learning and Computer Vision are being intensively researched and improved every day. In particular, Google's development of MediaPipe, an open-source framework for building world-class machine learning solutions that provide basic machine learning models for common tasks such as hand tracking, posture recognition, This project, "Human Posture Detection" is based on the detection of postures and is used to analyze, detect and classify fitness exercises. The final experimental results show that the algorithm proposed in this work can effectively identify correct and incorrect shapes performed in an exercise.

**TABLE OF CONTENTS**

[**ABSTRACT**](#_bookmark1) **1**

[**LIST OF FIGURES**](#_bookmark2) **3**

**LIST OF TABLES 4**

[**ABBREVIATIONS**](#_bookmark3) **5**

[**1 INTRODUCTION**](#_bookmark4) **6-7**

* 1. Problem description 6
  2. Purpose of Study 6
  3. General Approach 7
  4. Software Requirements 7

[**2 LITERATURE SURVEY**](#_bookmark20) **8-9**

2.1 Pose estimation techniques 8

2.2 Related Work 8

2.3 Utilization of previous research 9

**3 METHODOLOGY OF POSTURE CORRECTOR 10-23**

3.1 Exercise selection 10

3.2 Error determination 10

3.3 Data collection 11-12

3.4 Simple error detection 12-13

3.5 Data process for error detection 13-15

3.6 Model Training 15-16

3.7 Error detection process in depth 16-23

**4 RESULTS AND DISCUSSIONS 24--33**

4.1 Model Evaluation 24-30

4.2 Project Output 31-33

**5 CONCLUSION AND FUTURE ENHANCEMENT 34-35**

**REFERENCES 36**

**APPENDIX 39**

[**LIST OF FIGURES**](#_bookmark2)

|  |  |  |
| --- | --- | --- |
| Figure No | Title of the Figure | Page No |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | Dataset of pushup  Data processing  Example of landmarks for pushup exercise  Class balance of Bicep Curl's dataset  Class balance of Basic Plank's dataset  Class balance of Squat's dataset  Class balance of Pushup dataset  Web server - Video analyzing process  Bicep curl error - F1 curve  Bicep curl error - ROC curve  Plank error - F1 curve  Plank error - ROC curve  Squat stage - F1 score curve  Squat stage - ROC curve  Pushup error - F1 score curve  Pushup error - ROC curve  Bicep Curl error detection  Plank error detection  Squat error detection  Pushup error detection  User Interface using gradio | 10  10  14  17  19  21  23  24  27  27  28  29  30  30  31  32  32  33  33  34  34 |

[**LIST OF FIGURES**](#_bookmark2)

|  |  |  |
| --- | --- | --- |
| Figure No | Title of the Figure | Page No |
| 1  2  3  4  5  6  7  8 | Model training experiments for Bicep Curl’s error  Model training experiments for Plank’s error detection  Model training experiments for Squat's stage  Model training experiments for Pushup's stage  Bicep Curl error - Evaluation results  Plank error - Evaluation results  Squat stage - Evaluation results  Pushup stage - Evaluation results | 18  19  21  23  26  28  29  31 |

[**ABBREVIATIONS**](#_bookmark3)

1. **AI (Artificial Intelligence)**
2. **ML (Machine Learning)**
3. **DL (Deep Learning)**
4. **CV (Computer Vision)**
5. **MPL (Media Pipe)**
6. **HPE (Human Pose Estimation)**
7. **Gradio (User-Interface)**

**CHAPTER 1**

**INTRODUCTION**

* 1. **PROBLEM DESCRIPTION**

With advances in artificial intelligence and computing power, computer vision technology has made significant strides toward integration into our daily lives. Computer vision, a branch of computer science, focuses on developing systems that can process, analyze, and interpret visual data—such as images and videos—much like humans do. At its core, computer vision teaches machines to understand images at the pixel level, enabling them to retrieve, process, and interpret visual information through specialized algorithms.

Physical activity is now widely recognized as essential for maintaining health. Regular exercise can improve brain function, aid in weight management, reduce the risk of diseases, strengthen bones and muscles, and enhance the ability to perform daily tasks. Regardless of age, gender, or fitness level, everyone can benefit from being active. However, when it comes to exercise, proper form is more crucial than the number of repetitions. The way an exercise is executed can determine whether it is effective or harmful. Even experienced individuals benefit from feedback on their form, as perfecting it not only improves performance but also conserves energy and reduces the risk of injury.

For beginners or those working out alone, improper form is a common issue due to a lack of knowledge or instruction. This can lead to strained muscles or injuries, and without proper guidance, these individuals may not achieve their desired results, instead being sidelined by frustrating injuries.

* 1. **PURPOSE OF THE STUDY**

The objective of this study is to design and develop four distinct machine learning models, each tailored to one of the most commonly performed home exercises. The purpose of these models is to accurately detect any improper or incorrect movements while an individual is performing the corresponding exercise. By focusing on these widely practiced exercises, the system aims to provide real-time or post-exercise feedback on form accuracy, ensuring that users can perform their workouts safely and effectively.Each model is trained to recognize the specific biomechanics associated with its respective exercise, allowing it to identify subtle deviations from proper form. These deviations, if left unchecked, can lead to inefficiencies in the exercise and increase the risk of injury. The models will serve as virtual personal trainers, offering corrective insights and helping users adjust their posture or technique. This approach not only aids beginners in learning correct exercise execution but also assists experienced individuals in refining their form, ultimately enhancing the overall quality and safety of their workouts. By integrating machine learning with exercise analysis, this study aims to bridge the gap between at-home fitness routines and professional training, making expert-level guidance more accessible to a broader audience.

**1.3 GENERAL APPROACH**

* Research on which popular exercises which commonly improperly perform.
* Research on which technology to choose that is suitable to solve the problem.
* Collect and process data of the chosen exercises.
* Train and evaluate model for each exercise.
* Utilize the trained models

**1.4 SOFTWARE REQUIREMENTS**

1. Operating System:

* Windows 10/11, Linux (Ubuntu), or macOS

2. Programming Languages:

* Python 3.7+

3. Machine Learning & Deep Learning Libraries:

* TensorFlow, PyTorch, Keras, Scikit-learn

4. Computer Vision Tools:

* MediaPipe, OpenCV

5. Data Manipulation & Processing Libraries:

* NumPy, Pandas, Matplotlib, Seaborn

6. Development Environment:

* Jupyter Notebook, Google Colab, Visual Studio Code

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 POSE ESTIMATION TECHNIQUES**

1. Pose Estimation using Deep Learning and Computer Vision: Several studies have explored the use of deep learning techniques such as Convolutional Neural Networks (CNNs) combined with Computer Vision for human pose estimation. These studies highlight how models like OpenPose and MediaPipe have advanced in detecting human keypoints and body landmarks for various applications, including sports and fitness.
2. MediaPipe for Human Pose Detection: MediaPipe by Google offers a robust solution for real-time pose estimation and tracking. Multiple papers detail its use in fitness applications where correct form is essential. The flexibility of MediaPipe to integrate with custom machine learning models makes it ideal for fitness posture detection systems.
3. Machine Learning for Exercise Classification: Research has been conducted on classifying human activities, including exercises, using machine learning algorithms such as Decision Trees, Support Vector Machines (SVM), and deep learning models. These studies indicate that accurate classification of exercises is possible with sufficient labeled data and advanced pose estimation.
4. Applications of AI in Fitness: AI-based fitness tools are increasingly popular, leveraging CV and ML to provide users with insights into their form and performance. Studies explore various approaches to integrating AI into fitness apps and systems to monitor workouts, detect errors in posture, and suggest corrections.

**2.2 RELATED WORKS**

In March 2018, a study “Pose Trainer: Correcting Exercise Posture using Pose Estimation” is published by Steven Chen and Richard Yang (both from Standford University) [5] with the intention to use machine learning in fitness exercise for helping prevent injuries and im-prove the quality of people’s workouts with just a computer and a webcam. The report of the study introduces Pose Trainer, an end-to-end computer vision application that uses pose estimation, visual geometry, and machine learning to provide personalized feedback on fitness exercise form. The study worked with 4 different exercises (bicep curl, front raise, shoulder shrug and shoulder press) recording training videos for each, and use both geometric heuristic algorithms to provide personalized feed-back on specific exercise improvements, as well as machine learning algorithms to automatically determine posture correctness using only labeled input videos

**2.3 UTILIZATION OF PREVIOUS RESEARCH**

In this project, we leverage insights from previous research to enhance the accuracy and efficiency of exercise posture detection and classification. Studies on pose estimation techniques, such as OpenPose and MediaPipe, provide us with a strong foundation for detecting human body landmarks. By integrating MediaPipe into our system, we can accurately extract key points from exercise videos, benefiting from its proven performance in human pose estimation.

Furthermore, previous work on Machine Learning Models for Exercise Classification informs our choice of algorithms. Research has shown that deep learning models, such as CNNs and LSTMs, are highly effective in classifying human activities. We apply these methods to classify exercises as correct or incorrect, optimizing them with insights from prior studies on posture correction and feedback systems.

Lastly, the Applications of AI in Fitness provide a roadmap for integrating real-world usability into our system. By learning from prior feedback mechanisms, we enhance user experience by offering detailed corrective suggestions, building on proven strategies from earlier research. This combination of past research ensures that our project is both robust and practical for users seeking to improve their exercise form.

**CHAPTER 3**

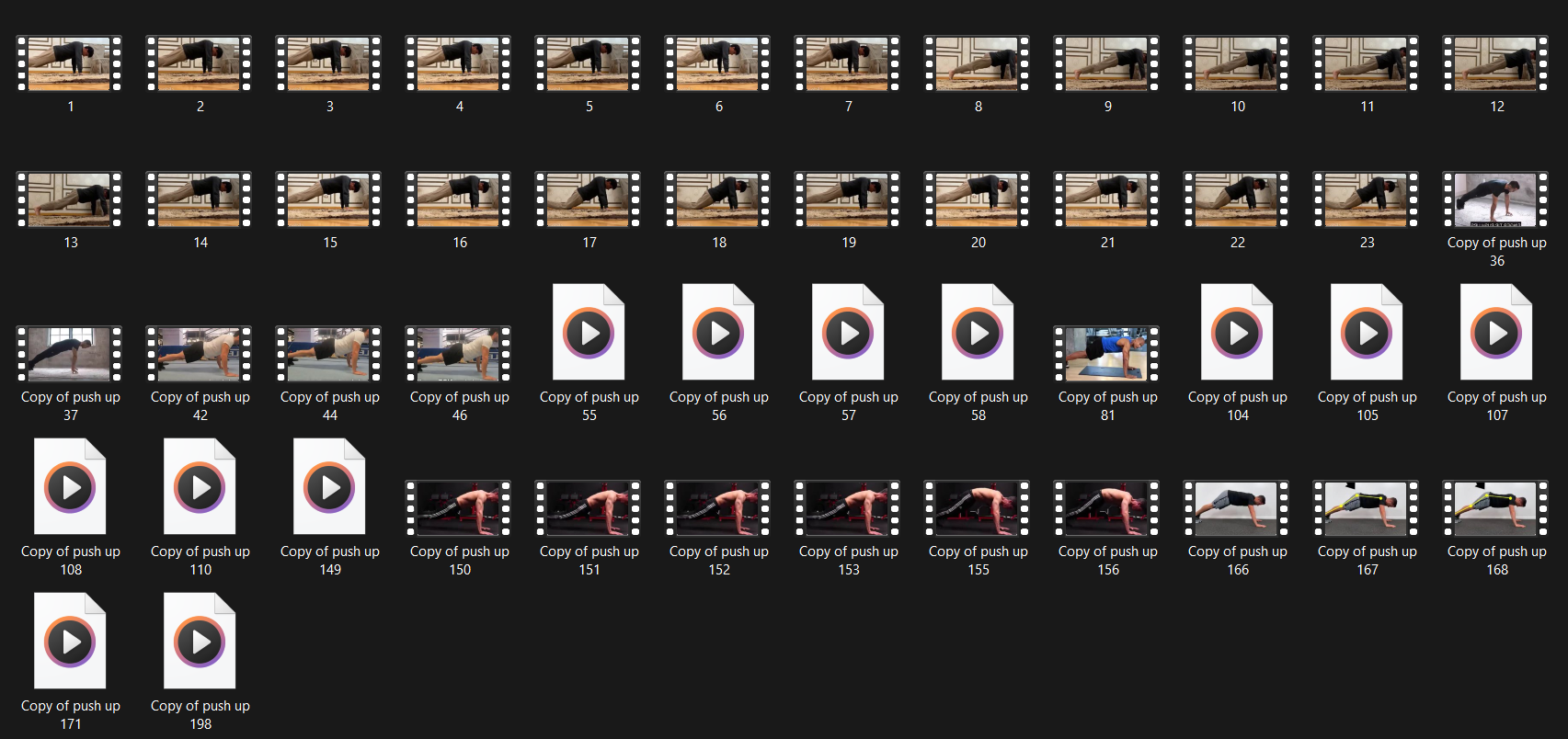
**METHODOLOGY OF POSTURE CORRECTOR**

**3.1 EXERCISES SELECTION**

First and foremost, the initial step of this thesis is to decide which exercises to use to train the machine learning model for incorrect pose correction. Since the time for this work is limited, only 4 exercises are selected. Each exercise must meet the following criteria:

* An exercise is popular among people who exercise at home or who exercise mostly alone. Therefore, there is a high probability that the exercise will be performed improperly.
* An exercise must contain at least 2 common mistakes that affect the inefficiency of the exercise.

Based on the above criteria, we decided on 4 exercises: biceps curl, basic plank, basic squat and pushup.



*Figure 1: Dataset of pushup*

**3.2 ERROR DETERMINATION**

After selecting four exercises for analysis, the next step involves identifying at least two common errors associated with each exercise and developing effective detection strategies. A detailed exploration of the methods for detecting these errors will be presented on page 23. The identified common errors for each exercise are as follows:

* Bicep Curl:
  + Loose upper arm positioning
  + Weak peak contraction
  + Leaning back during the standing posture
* Basic Plank:
  + Sagging of the lower back
  + Elevated high back
* Basic Squat:
  + Foot placement that is either too narrow or too wide
  + Knee positioning that is excessively tight or too wide
* Push-Up:
  + Elevated high back
  + Elbow Angles
  + Sagging of the lower back

By systematically addressing these errors, the project aims to provide users with comprehensive feedback on their exercise form, facilitating improved performance and injury prevention.

**3.3 DATA COLLECTING**

**3.3.1 Self-collected data**

* Bicep Curl: Videos were recorded to capture various angles and performances of individuals performing bicep curls, focusing on different common errors identified, such as loose upper arm positioning and weak peak contraction.
* Basic Plank: Video recordings of individuals in plank positions were made, emphasizing variations in lower back sagging and elevated high back posture.
* Basic Squat: Data was gathered through videos of squat exercises, paying attention to foot placement and knee positioning, which are critical for proper form.
* Push-Up: Recordings of push-up exercises were collected to analyze knee angles and the positioning of the knees in relation to the toes during the descent.

**3.3.2 Public Dataset from Kaggle**

**1. Bicep Curl**

For the bicep curl exercise, we utilized self-collected data to create a dataset that highlights common form errors. We recorded videos featuring various individuals performing bicep curls, focusing on specific errors such as loose upper arm positioning and weak peak contraction. A total of 20 video files were curated, showcasing both proper execution of the bicep curl and common mistakes.

**2. Basic Plank**

For the plank exercise, we sourced a dataset from Kaggle, a well-known online community platform for data scientists and machine learning enthusiasts. The dataset consists of various yoga poses, including well-known ones such as downward dog, goddess pose, tree pose, plank pose, and warrior pose. We specifically focused on the folder containing images of individuals performing the plank pose correctly. After reviewing the 266 image files available in that folder, we handpicked 30 images that accurately represent the proper form for the basic plank, discarding any images that did not meet this criterion.

**3. Basic Squat**

For the basic squat exercise, we employed self-collected data to capture the intricacies of squat form. We recorded several videos of individuals performing squats, paying close attention to common errors such as improper foot placement and incorrect knee positioning. A total of 20 video files were gathered, showcasing both proper squat technique and frequent mistakes.

**4. Push-Up**

For the push-up exercise, we also collected data through video recordings, focusing on capturing a range of individuals performing push-ups. We concentrated on identifying common errors, including incorrect knee angles and knee positioning relative to the toes during the descent. After reviewing the footage, we extracted 20 video files that illustrate both proper push-up form and common mistakes.

By systematically gathering and curating these datasets for each exercise, we aim to create a comprehensive training set that accurately reflects proper and improper exercise forms, facilitating effective model training and evaluation.

**3.4 SIMPLE ERROR DETECTION**

For some simple errors (for example, the feet placement error in squat), the detection method is either measuring the distance/angle between different joints during the exercise with the coordinate outputs from MediaPipe Pose.

**a. Distance Calculation**

Assume there are 2 points with the following coordinates: Point 1 (𝑥1, 𝑦1) and Point 2 (𝑥2, 𝑦2), below is the formula to calculate the distance between 2 points.

**

**b. Angle Calculation**

Assume there are 3 points with the following coordinates: Point 1 (𝑥1, 𝑦1), Point 2 (𝑥2, 𝑦2) and Point 3 (𝑥3, 𝑦3), below is the formula to calculate the angle created by 3 points**.**

𝑎𝑛𝑔𝑙𝑒\_𝑖𝑛\_𝑟𝑎𝑑𝑖𝑎𝑛 = 𝑎𝑟𝑐𝑡𝑎𝑛2 (𝑦3 − 𝑦2, 𝑥3 − 𝑥2) − 𝑎𝑟𝑐𝑡𝑎𝑛2(𝑦1 − 𝑦2, 𝑥1 − 𝑥2)

𝑎𝑛𝑔𝑙𝑒\_𝑖𝑛\_𝑑𝑒𝑔𝑟𝑒𝑒 = (𝑎𝑛𝑔𝑙𝑒\_𝑖𝑛\_𝑟𝑎𝑑 ∗ 180)/Π

**3.5 DATA PROCESS FOR ERROR DETECTION**

**3.5.1 Data processing**

The general approach of processing data from collected videos for training model is illustrated in Figure 2: Data processing below.

A diagram of a video

Description automatically generated

*Figure 2: Data processing*

**3.5.2 Detecting important landmarks**

For each exercise there are different poses/body positions, so it is important to identify the body parts (shoulder, hip, ...) that contribute to the exercise. The important landmarks identified for each exercise are used to extract the position of the body parts during the exercise with MediaPipe Pose.

A black rectangular object with white lines

Description automatically generated

*Figure 3: Example of landmarks for pushup exercise*

In addition, the properties of each landmark, such as x, y, z, and visibility, are reduced to 4 headers for a csv file. For example, with the NOSE landmarks, the 4 headers would be as follows: NOSE\_x, NOSE\_y, NOSE\_z and NOSE\_v (see page 15 - MediaPipe Pose Output for more detail). Therefore, for each exercise, an empty csv file will be initialized. The first header is the “label” which contain a class for each datapoint, the rest of file’s headers are all the important headers with their properties flatten

**3.5.3 . Extract data from video to file using OpenCV and MediaPipe**

With every exercise, there are videos which are separated into different classes. With each video, every frame that matches a certain form of an exercise would be processed as follows:

1. Process every frame using OpenCV then utilize MediaPipe Pose to produce a list of coordinate predictions of all keypoint locations, and their corresponding prediction confidence.
2. Configuration options for Medipipe Pose:

* MIN\_DETECTION\_CONFIDENCE: Minimum confidence value ([0.0, 1.0]) from the person-detection model for the detection to be considered successful. Default to 0.5.
* MIN\_TRACKING\_CONFIDENCE: Minimum confidence value ([0.0, 1.0]) from the landmark-tracking model for the pose landmarks to be considered tracked successfully, or otherwise person detection will be invoked automatically on the next input image. Setting it to a higher value can increase robustness of the solution, at the expense of a higher latency. Default to 0.5

A screen shot of a computer code

Description automatically generated

* For every frame with the list of predicted landmarks, the important landmarks for a correspond exercise would be extracted in append as a row to the csv file with the label column matches the class of the video.
* The collected data which is saved in a csv file would be split to train and evaluate model. 80% of the data will be used for model training and the remaining 20% will be used for model evaluation.

**3.6 MODEL TRAINING**

There are 2 methods used in this thesis for model training. For each exercise, the models trained for each method will be compared and the best model will be chosen.

* Classification with Scikit-learn.
* Building a Neural Network for classification with Keras.

**3.6.1 Classification with Scikit-learn**

The machine learning algorithms used for model training are: Decision Tree/Random Forest (RF), K-Nearest Neighbors (KNN), C-Support Vector (SVC), Logistic Regression classifier (LR) and Stochastic Gradient Descent classifier (SGDC). After the training session, each result from each algorithm will be evaluate with metrics such as: precision, recall, accuracy and F1 score. The algorithm which provides the best results will be selected. Below is the sample code for this section

A screen shot of a computer code

Description automatically generated

**3.6.2 Neural Network for classification with Keras**

With Keras, composing a neural network by creating layers and linking them together. In this project, one type of layer called a fully connected or Dense layer. In Keras, this is defined by the keras.layers.Dense class. A dense layer has a number of neurons, which is a parameter which can be chosen when the layer is created. When connecting the layer to its input and output layers every neuron in the dense layer gets an edge (i.e. connection) to all of the input neurons and all of the output neurons.

Occasionally, Dropout layers are utilized in the model’s architecture. The term “dropout” refers to dropping out the nodes (input and hidden layer) in a neural network. The nodes are dropped by a dropout probability of p. Dropout layer is used to combat the problem of overfitting. In overfitting, a unit may change in a way that fixes up the mistakes of the other units, using dropout, it prevents these units to fix up the mistake of other units, thus preventing co-adaptation, as in every iteration the presence of a unit is highly unreliable. Therefore, by randomly dropping a few units (nodes), it forces the layers to take more or less responsibility for the input by taking a probabilistic approach.

For each training session, there are 4 models’ architectures which would be experiment with. Depends on the evaluation metrics for each model, the model that yield the best results will be chosen.

* Neural network with 3 Dense layers.
* Neural network with 5 Dense layers.
* Neural network with 5 Dense layers and 2 Dropout layers.
* Neural network with 7 Dense layers.

**3.7 ERROR DETECTION PROCESS IN DEPTH**

**3.7.1 Biceps Curl**

**3.7.1.1 Basic technique, errors description and important landmarks**

The biceps curl is a highly recognizable weight-training exercise that works the muscles of the upper arm, and to a lesser extent, those of the lower arm. The exercise has to be performed while standing, involve around the arms moving up and down while holding a dumbbell or barbell.

There are 3 popular errors of bicep curl that will be targeted in this thesis:

* Loose upper arm: when one arm moves up during exercise, the upper arm moves instead of standing still.
* Weak peak contraction: when an arm moves up, it does not go high enough, so the biceps do not contract to the best of effectiveness.
* Lean too far back: the upper body of the performer leans backwards and forwards during the exercise for the swing.

**3.7.1.2 Errors detection method**

**a. Loose upper arm**

Can be detected by calculating the angle between the elbow, shoulder and the shoulder’s projection on the ground. Through my research, if the angle is over 40 degrees, the movement will be classified as a “loose upper arm” error.

**b. Weak peak contraction**

Can be detected by calculating the angle between the wrist, elbow and shoulder when the performer’s arm is coming up. Through my research, if the angle is more than 60 degrees before the arm comes down, the movement will be classified as a “weak peak contraction” error.

**c. Lean too far back**

Due to its complexity, machine learning is used for this error. Videos of participants with and without this error are used as data for the training model. Figure 4 is a visual graph represent the number of frames gathered from the videos and their classes. There are a total of 15372 images, in which, there are 8238 samples (53.6%) belong to class correct posture (C) and 7134 (46.4%) samples belong to class lean back (L). Every datapoint in the dataset is consist of 37 columns which are formed based on its classed and the important landmarks.

A red and blue bar graph

Description automatically generated

*Figure 4: Class balance of Bicep Curl's dataset*

The results of the experiment in training mentioned model are shown in the table below.

A table with numbers and a few words

Description automatically generated with medium confidence

*Table 1. Model training experiments for Bicep Curl’s error*

**3.7.2 Basic Plank**

**3.7.2.1 Basic technique, errors description and important landmarks**

The plank, or planking, is an exercise that involves your core muscles, improving your strength, balance and endurance. To perform a plank, lying on the ground with the elbows in line with the shoulder and the feet shoulder width apart, Push body up bearing the weight on the forearms and feet, body is kept straight.

There are 2 popular errors of basic plank that will be targeted in this thesis:

* High lower back: when performing the exercise, the lower back is not kept straight, but is lifted too high.
* Low lower back: when performing the exercise, the lower back is not kept straight, but is brought down too low.

**3.7.2.2 Errors detection method**

**a. Low back**

This error is detected by calculating the hip drop relative to the shoulders and ankles. When the lower back sags, the hips dip below the aligned position, creating a shallow angle. The correct alignment is established by analyzing proper form push-up videos, collecting 450 data points to define the acceptable range. A lower angle suggests a sagging lower back, signaling improper form.

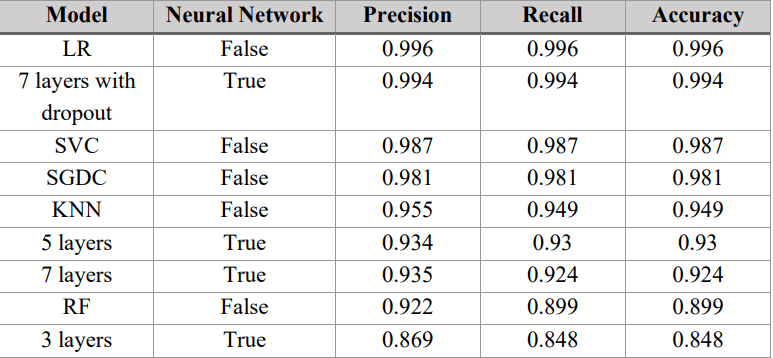
**b. High lower back**  
 This error can be detected by measuring the angle between the shoulder, hip, and ankle. If the lower back is too high, the hips will form a steep angle with the body, breaking the straight alignment. A threshold is determined based on the analysis of videos of properly performed push-ups, where 500 data points are collected. Any significant deviation from the standard angle range indicates a high lower back position.

A bar graph with different colored rectangles

Description automatically generated

*Figure 5: Class balance of Basic Plank's dataset*

The results of the experiment in training mentioned model are shown in the table below



*Table 2*: *Model training experiments for Plank’s error detection*

**3.7.3 Basic Squat**

**3.7.3.1 Basic technique, errors description and important landmarks**

A squat a strength exercise in which the trainee lowers their hips from a standing position and then stands backup.During the descent of a squat, the hip and knee joints flex while the ankle joint conversely the hip and knee joints extend and the ankle joint plantarflexes when standing up.

There are 2 popular errors of basic squat that will be targeted in this thesis:

* Feet placement: The placement of the feet is extremely important in the squat position. The 2 feet should be placed so that the width of 2 feet is approximately equal to the width of 2 shoulders.
* Knee placement: Knee placement is not only important, but can be dangerous if done incorrectly during heavy loading. During the "down" phase of the exercise, the knee should be wider open than the feet are wide

**3.7.3.2 Errors detection method**

**a. Feet placement**

Can be detected by calculating ratio between the distance of 2 feet and the distance of 2 shoulders. To precisely choose the correct ratio, videos of contributors perform proper form of a squat are analyzed. In that respect, 851 datapoints are gathered.

**b. Knee placement**

Can be detected by calculating ratio between the distance of 2 knee and 2 feet. Similar to the previous error, videos of contributors are analyzed to determine a correct threshold. Due to the dynamic movement of the knee during the exercise, the calculated ratio from the data will be separate into 3 stages: up, middle and down.

A red and blue rectangular bars

Description automatically generated

*Figure 6: Class balance of Squat's dataset*

The results of the experiment in training mentioned model are shown in the table below

*A table with numbers and a few words

Description automatically generated with medium confidence*

*Table 3: Model training experiments for Squat's stage*

**3.7.4 Pushup**

**3.7.4.1 Basic technique, errors description and important landmarks**

The push-up is performed by starting in a high plank position with hands placed slightly wider than shoulder-width apart and keeping the body in a straight line from head to heels. As you lower your chest towards the floor, bend your elbows and maintain core engagement, then push back up to the starting position.

There are 3 popular errors of pushup that will be targeted in this thesis:

* Hip placement: Proper hip alignment is crucial in a push-up. The hips should stay in line with the shoulders and heels throughout the movement. Sagging or raising the hips disrupts form and reduces core engagement, increasing the risk of back strain.
* Elbow positioning: Elbow placement is critical for maintaining shoulder health. During the downward phase, the elbows should stay close to the body at a 45-degree angle. Flaring them out too wide can strain the shoulders and compromise the efficiency of the push-up.
* Core engagement: Maintaining a strong core is essential to prevent the midsection from sagging or the back from arching. Failure to engage the core properly leads to a breakdown in form and increased risk of lower back injury.

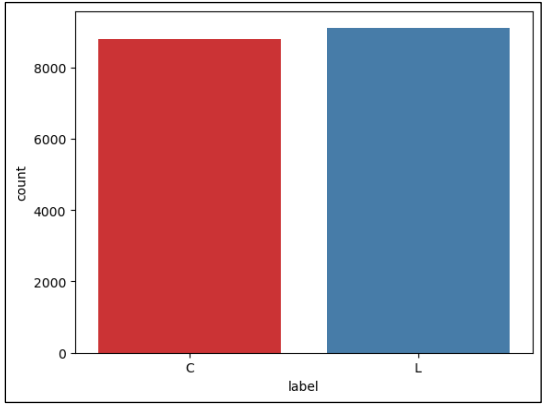
**3.7.4.2 Errors detection method**

**a. Low back**

This error is detected by calculating the hip drop relative to the shoulders and ankles. When the lower back sags, the hips dip below the aligned position, creating a shallow angle. The correct alignment is established by analyzing proper form push-up videos, collecting 450 data points to define the acceptable range. A lower angle suggests a sagging lower back, signaling improper form.

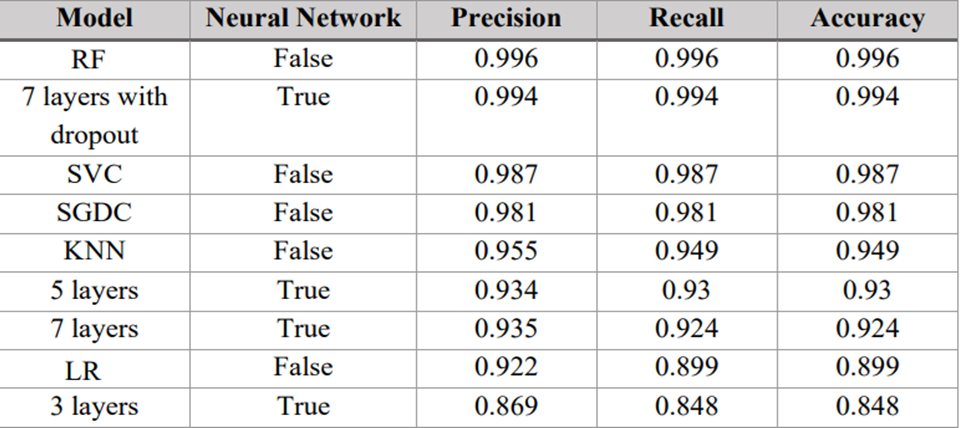
**b. Elbow positioning**  
 Elbow positioning is detected by calculating the angle between the shoulder, elbow, and wrist. Ideally, the elbow angle should remain at approximately 45 degrees from the torso. The correct range of angles is determined by analyzing videos of correct push-up form. The elbow angle is calculated during the downward, mid, and upward phases, and the movement across each phase is evaluated using 600 collected data points.

**c. High lower back**  
 This error can be detected by measuring the angle between the shoulder, hip, and ankle. If the lower back is too high, the hips will form a steep angle with the body, breaking the straight alignment. A threshold is determined based on the analysis of videos of properly performed push-ups, where 500 data points are collected. Any significant deviation from the standard angle range indicates a high lower back position.

****

*Figure 7: Class balance of Pushup dataset*

The results of the experiment in training mentioned model are shown in the table below

****

*Table 4: Model training experiments for Pushup's stage*

**3.8 USER INTERFACE USING GRADIO**

Gradio is a Python library that allows developers to create user-friendly interfaces for machine learning models or any Python function quickly. It enables users to interact with machine learning models via a web interface, where they can input data (like images, text, or audio) and receive outputs (like predictions, visualizations, etc.).

For each video, the process of analysis from the server is described here:

- Error detection with trained models: Depending on the type of exercise, the appropriate model is selected for error detection. Each error is recorded to return it to the client.

In summary, the procedure for training the model was presented in detail. At the same time, the construction of a web application was also presented to demonstrate the predictive capability of the model. The next section focuses on the process of evaluating and testing the quality of the mentioned model and web application.

**A diagram of a video file

Description automatically generated**

*Figure 8: Web server - Video analyzing process*

**CHAPTER 4**

**RESULTS AND DISCUSSIONS**

**4.1 MODEL EVALUATION**

This section contains an objective evaluation of the performance of the trained models, an explanation of some of the factors that influence the evaluation results, and an outline of the testing procedure

**4.1.1 Evaluation metrics**

Evaluation metrics are used to measure the quality of trained machine learning model. Evaluation of machine learning models or algorithms is essential for any project. There are many different types of evaluation metrics to test a model. The following are the metrics that were used to evaluate the trained models in this thesis:

- *Confusion matrix*: provides a detailed overview of the classification. For a better performing model True Positive (TP), True Negative (TN) must be high and False Negative (FN), False Positive (FP) should be low as possible.

o True Positive (TP): Number of correctly predicted positive samples.

o False Positive (FP): Number of negative samples incorrectly predicted as positive.

o True Negative (TN): Number of correctly predicted negative samples.

o False Negative (FN): Number of positive samples incorrectly labelled as negative.

- *Precision*: tells the ratio of the true positives to the total predicted positives.

A black text on a white background

Description automatically generated

- *Recall*: tells the proportion that the model is accurately classifying the true positives. It is also called Sensitivity.

A mathematical equation with black text

Description automatically generated

*- F1 score*: defined as the harmonic mean of precision and recall. The higher the precision and recall, the higher the F1-score.

A black text with black lines

Description automatically generated

*- F1 score curve:* a graph shows the tradeoff between F1 score and different threshold. By observing the changes of F1 score, the curve can help to find the optimal threshold for a model.

*- ROC curve (receiver operating characteristic curve):* a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters: True Positive Rate (TPR) and False Positive Rate (FPR).

o 𝑇𝑃𝑅 = 𝑇𝑃/(𝑇𝑃 + 𝐹𝑁)

o 𝐹𝑃𝑅 = 𝐹𝑃/(𝐹𝑃 + 𝑇𝑁)

*- AUC (area under the ROC curve):* measures the entire two-dimensional area underneath the entire ROC curve. AUC provides an aggregate measure of performance across all possible classification thresholds.

**4.1.2 Evaluation results**

**1. Bicep Curl**

According to the metrics from the models training experiments mentioned shown in Table I. Model training experiments for Bicep Curl’s error, the best model for this error is the K-Nearest Neighbors (KNN) model. As shown below, the model gives decent results.

A table with numbers and symbols

Description automatically generated with medium confidence

*Table 5: Bicep Curl error - Evaluation results*

The choice of confidence threshold affects the evaluation and detection results of the model, choosing the appropriate confidence threshold will give better result. Observing the F1 curve and ROC curve below can help with choosing the optimal confidence threshold. From the F1 curve chart, the lines between classes are closed to each other, and the threshold which yield the best F1 score is around 0.5.

A graph with blue and orange lines

Description automatically generated

*Figure 9: Bicep curl error - F1 curve*

*A graph of a positive result

Description automatically generated*

*Figure 10: Bicep curl error - ROC curve*

**2. Basic plank**

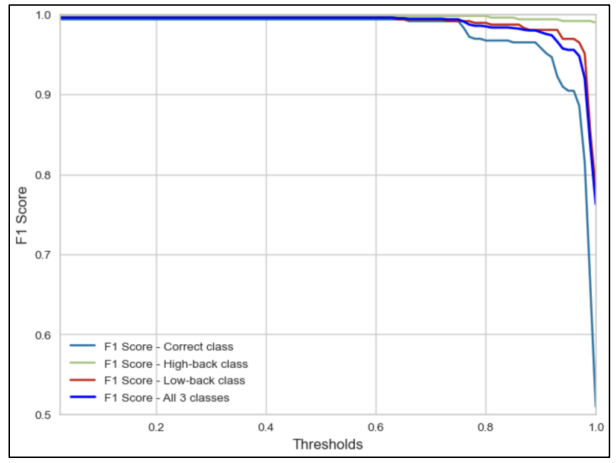
According to the metrics from the models training experiments mentioned shown in Table II. Model training experiments for Plank’s error detection, the best model for this error is the Logistic Regression (LR) model. As shown below, the model gives decent results.

A table with numbers and symbols

Description automatically generated

*Table 6: Plank error - Evaluation results*

Observing the F1 curve and ROC curve below can help with choosing the optimal confidence threshold. From the F1 curve chart, the lines between classes are closed to each other, and the threshold which yield the best F1 score is around 0.6.



*Figure 11: Plank error - F1 curve*

*A graph with a line graph

Description automatically generated with medium confidence*

*Figure 12: Plank error - ROC curve*

**3. Basic squat**

According to the metrics from the models training experiments mentioned shown in Table III. Model training experiments for Squat's stage, the best model for stage detection is the Logistic Regression (LR) model. As shown below, the model provides decent results.

A table with numbers and symbols

Description automatically generated

*Table 7: Squat stage - Evaluation results*

Observing the F1 curve and ROC curve below can help with choosing the optimal confidence threshold. From the F1 curve chart, the lines between classes are closed to each other, and the threshold which yield the best F1 score is around 0.5.

A graph of a number of classes

Description automatically generated

*Figure 13: Squat stage - F1 score curve*

*A graph with a line and a red line

Description automatically generated with medium confidence*

*Figure 14: Squat stage - ROC curve*

**4. Pushup**

According to the metrics from the models training experiments mentioned shown in Table IV. Model training experiments for Pushup error, the best model for this error is the Random Forest (RF) model. As shown below, the model provides decent results

A table with numbers and symbols

Description automatically generated

*Table 8: Pushup stage - Evaluation results*

Observing the F1 curve and ROC curve below can help with choosing the optimal confidence threshold. From the F1 curve chart, the lines between classes are closed to each other, and the threshold which yield the best F1 score is around 0.5.

A graph of a graph with different colored lines

Description automatically generated with medium confidence

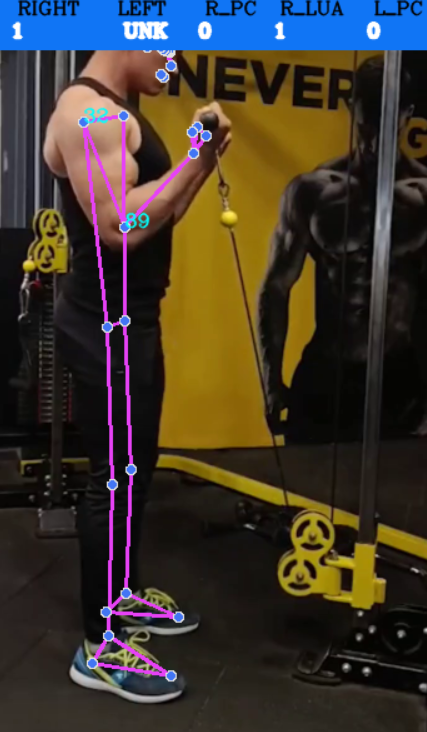
*Figure 15: Pushup error - F1 score curve*

*A graph with a red line

Description automatically generated*

*Figure 16: Pushup error - ROC curve*

**4.2 PROJECT OUTPUT**

****

*Figure 17: Bicep Curl error detection*

**A close-up of a red and purple line

Description automatically generated**

*Figure 18: Plank error detection*

**A person squatting in front of a curtain

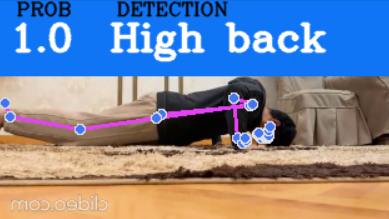
Description automatically generatedA person holding an object

Description automatically generated**

*Figure 19: Squat error detection*

**A person lying on the floor

Description automatically generated**

****

*Figure 20: Pushup error detection*

A screenshot of a computer

Description automatically generated

*Figure 21: User Interface using gradio*

**CHAPTER 5**

**CONCLUSION AND FUTURE ENHANCEMENT**

**5.1 CONCLUSION**

During the course of this research, considerable progress was made in gaining a deeper understanding of various machine learning theories, technologies, and tools, particularly in the field of computer vision and human posture detection. The project leveraged the advanced capabilities of the MediaPipe Pose framework, an open-source tool that enables precise human pose tracking, to develop machine learning models tailored to detect and classify posture errors during exercise routines. These models were specifically trained to analyze four of the most common exercises: biceps curl, basic plank, basic squat, and pushup. Each of these exercises presents unique biomechanical challenges, and the models were designed to identify and correct form-related errors that can negatively impact performance or lead to injury.

The results of this project exceeded initial expectations, showcasing the models’ capability to accurately detect incorrect movements and postures with a high degree of reliability. By breaking down the key elements of each exercise, such as body alignment, joint angles, and movement dynamics, the models were able to classify a wide range of errors, from improper knee and hip alignment during squats to incorrect elbow positioning and core engagement in push-ups. This level of detailed analysis highlights the potential of machine learning to provide insights into exercise performance, ultimately aiming to enhance safety and effectiveness during workouts.

In conclusion, this research not only fulfilled its objectives of detecting and classifying exercise errors but also laid the groundwork for the broader application of machine learning technologies in the fitness industry. The promising results of this project underscore the potential for machine learning-based solutions to be integrated into fitness apps, home workout systems, and rehabilitation programs, offering a more personalized, data-driven approach to physical training and wellness.

**5.2 FUTURE WORKS**

Due to constraints in time and resources, the current focus of the study was on identifying and correcting the most common and basic errors associated with each of the four exercises. While the results have been promising, there are several areas for future exploration and improvement to further refine and extend the system's capabilities:

* Dataset Expansion: To reduce bias and enhance model accuracy, the dataset should be extended to include videos from a broader range of participants. This will allow the models to generalize better and work effectively with individuals who vary in body type, fitness level, and exercise form.
* Additional Error Detection: Future research should explore other common errors that may occur during the four exercises (biceps curl, plank, squat, and push-up), as the current study focused primarily on the most frequent mistakes.
* Incorporating More Exercises: Expanding the scope of the project to include other popular exercises that are frequently performed incorrectly would make the system more comprehensive and applicable to a wider audience.
* Automatic Exercise Detection: Adding the ability to detect the start and end of an exercise session would greatly enhance the system’s real-world usability. This feature would allow for more automated and precise analysis, providing users with detailed feedback on each repetition without manual input.

**REFERENCES**

[1] Alexandra Merisoiu, “Lunges – Common Mistakes And Correct Technique”, 25/12/2015. Available: https://themerisoiutechnique.com/2015/12/25/lungescommon-mistakes-and-correct-technique/.

[2] Anilkumar, Ardra, Athulya KT, Sarath Sajan, and Sreeja KA. "Pose Estimated Yoga Monitoring System." Available at SSRN 3882498 (2021).

[3] Babara Gibson, “WHY FORM IS IMPORTANT IN WORKING OUT”, 29/07/2016. Available: https://www.fitness19.com/why-form-is-important-inworking-out/.

[4] Bojana Galic, “6 Biceps Curl Mistakes That Make This Exercise Way Less Effective”, 13/10/2019. Available: <https://www.livestrong.com/article/13722123-biceps-curl-mistakes>.

[5] Chen, Steven, and Richard R. Yang. "Pose Trainer: correcting exercise posture using pose estimation." arXiv preprint arXiv:2006.11718 (2020).

[6] Deven Rathore, “Building A Video Streaming App With Nuxt.js, Node And Express”, 13/04/2021. Available: https://www.smashingmagazine.com/2021/04/building-video-streaming-appnuxtjs-node-express/.

[7] Eric Bugera, “Are You Making These 5 Squat Mistakes? Here’s How To Fix Them”, 21/09/2021. Available: <https://barbend.com/squat-mistakes/>.

[8] GOOGLE LLC, “MediaPipe”. Avaiable: <https://mediapipe.dev/>.

[9] Ivan Grishchenko, Valentin Bazarevsky, Eduard Gabriel Bazavan, Na Li, Jason Mayes, “3D Pose Detection with MediaPipe BlazePose GHUM and TensorFlow.js”, 30/08/2021. Available:

https://blog.tensorflow.org/2021/08/3d-pose-detection-with-mediapipeblazepose-ghum-tfjs.html.

[10] Jakub Czakon, “F1 Score vs ROC AUC vs Accuracy vs PR AUC: Which Evaluation Metric Should You Choose?”, 14/11/2022. Available: <https://neptune.ai/blog/f1-score-accuracy-roc-auc-pr-auc>.

[11] Jason BrownLee, “How to Use ROC Curves and Precision-Recall Curves for Classification in Python”, 31/08/2018. Available: https://machinelearningmastery.com/roc-curves-and-precision-recall-curvesfor-classification-in-python.

[12] Josyula, Rohit, and Sarah Ostadabbas. "A Review on Human Pose Estimation." arXiv preprint arXiv:2110.06877 (2021).

[13] Kukil, “Building a Poor Body Posture Detection and Alert System using MediaPipe”, 08/03/2022. Available: https://learnopencv.com/building-a-bodyposture-analysis-system-using-mediapipe/.

[14] Lipton, Zachary C., Charles Elkan, and Balakrishnan Naryanaswamy. "Optimal thresholding of classifiers to maximize F1 measure." In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 225-239. Springer, Berlin, Heidelberg, 2014.

[15] Lugaresi, Camillo, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang et al. "MediaPipe: A framework for perceiving and processing reality." In Third Workshop on Computer Vision for AR/VR at IEEE Computer Vision and Pattern Recognition (CVPR), vol. 2019. 2019.

[16] Lugaresi, Camillo, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang et al. "MediaPipe: A framework for building perception pipelines." arXiv preprint arXiv:1906.08172 (2019).

[17] Siddharth M, “Performing multi-class Classification on FIFA Dataset Using Keras”, 14/07/2021. Available: https://www.analyticsvidhya.com/blog/2021/07/performing-multi-classclassification-on-fifa-dataset-using-keras/.

[18] Tiffany Ayuda, “3 Common Plank Mistakes (And How to Avoid Them)”, 22/10/2022. Available: <https://dailyburn.com/life/fitness/how-to-do-a-plank/>

**APPENDIX**

**1. Bicep Model**

**a)Code of error detection**

A screen shot of a computer

Description automatically generated

**b)Output**

**A person working out in a gym

Description automatically generated**

**2. Plank Model**

**a)Code of error detection**

A screen shot of a computer code

Description automatically generated

**b)Output**

**A person lying on a bed

Description automatically generated**

**3. Squat Model**

**a)Code of error detection**

A screen shot of a computer screen

Description automatically generated

**b)Output**

**A person sitting on a stage

Description automatically generated**

**4. Pushup Model**

**a)Code of error detection**

A screen shot of a computer program

Description automatically generated

**b)Output**

**A person doing a plank on a rug

Description automatically generated**