ARTIFICIALLY INTELLIGENT HOME TRAINING ASSISTANT

Author

Syed Ahmed Hasan Ovais 2016-UET-NML-CS-25

Supervisor

Dr. Junaid Akhtar

Department of Computer Science

Namal Institute, Mianwali.

ARTIFICIALLY INTELLIGENT HOME TRAINING ASSISTANT

Authors

Syed Ahmed Hasan Ovais 2016-UET-NML-CS-25

A report submitted in partial fulfillment of the requirement for the degree of B.Sc. Computer Science.

Project Supervisor

Dr. Junaid Akhtar

External Examiner	________________
Project Co Supervisor	Dr. Bacha Rehman
External Supervisor	Ahmer Saleem

Department of Computer Science

Faculty of Computer Science

Namal Institute, Mianwali.

Table of Contents

List of Figures

List of Tables

Abstract

Regular exercise is one of the most significant factors in keeping up a good state of health. Previously, various solutions have been proposed to help individuals when working out. While the vast majority of those solutions center around utilizing wearable computing devices for tracking cardio activities, for example, running and cycling. However it does not change the fact that a wearable is generally connected to one piece of the body, tracks just that area, and is lacking for catching a wide range of workout exercises, particularly when different limbs are included. Cameras, on the other side, can completely track a client's body, however experience some noise, we exploit cell phones to support leisure activities with an emphasis on resistance training. We describe how off-the-shelf cell phones without extra outer sensors can be utilized to capture resistance training data and to give reliable training feedback.

In this report I present a unique camera-based solution for consequently distinguishing and recognizing 5 distinct workout exercises in unconstrained conditions. The user will be notified with the current performing exercise and repetition count from smartphone’s acceleration stream(video).

ACKNOWLEDGEMENTS

The achievement and ultimate result of our Final Year Project required a considerable measure of the direction and help from numerous individuals, including our advisor Dr. Junaid Akhtar, the co advisor Dr. Bacha Rehman and external advisor Ahmer Saleem, who helped us from the selection of the project to the progress in every little detail of our project. We are extremely privileged to have got this all along to the completion of the project. All that is done was just because of such superb supervision and help and I would especially say thanks to them.

UNDERTAKING

I hereby certify that "ARTIFICIALLY INTELLIGENT HOME TRAINING ASSISTANT" is my own project, neither has it been copied from somewhere else nor the work has been presented somewhere else for assessment.

Syed Ahmed Hasan Ovais 2016-UET-NML-CS-25

KeyWord

GLM: Generalized Linear Model

FP-Growth: Frequent Pattern Growth Algorithm

DBSCAN: Density-based spatial clustering of applications with noise

CNN: Convolution Neural Network

ANN: Artificial Neural Network

IMU: Inertial Monitoring Unit

MVC: Model View Controller Architectural Model

RNN: Recurrent Neural Network

RCNN: Recurrent Convolution Neural Networked

DFA: Deterministic Finite Automaton

Chapter 1 Introduction

1.1 Overview

Regular physical workout decreases the threat of obesity, diabetes, hypertension, cardiovascular hazard factors, and plays a significant role in weight reduction.[7] As indicated by professionals physical activity and exercise should be endorsed and prescribed to every man or woman with diabetes as a major aspect in the management of hypoglycemic control and overall health.[1]

To keep basic fitness and build strength, the Centers for Disease Control and Prevention (CDC) endorses grown-ups to do resistance workout at least two times per week.[1]And youngsters and adolescents need to accumulate an average of at the least 60 minutes each day and up to several hours as a moderate-intensity physical exercise.[3] Well regardless of the advantages of everyday exercise, the huge majority struggle to preserve up a consistent workout routine. This disappointment is frequently seen because of the absence of motivation and feedback, and due to the feeling of embarrassment while asking for help.[4]

1.2 Problem Statement

Despite numerous benefits and advisory clues, people lack the interest an don't perform physical exercise and get abandoned in 35 separate pathological and clinical situations. Many of the 35 conditions are subdivided under major categories, such as loss of operative capacities with chronological aging; metabolic syndrome, obesity, insulin resistance, prediabetes/type 2 diabetes (T2D), nonalcoholic liver disease, cardiovascular disease s (CVDs).[5] One approach to handle a lack of inspiration or motivation is a thorough performance analysis of the previous exercise and continuous feedback through personalized trainer [6]

1 Centers for Disease Control and Prevention. Physical activity recommendations for adults: cdc.gov/physicalactivity/everyone/guidelines/adults.html

1.3 Background, Objectives and Significance of the study

1.3.1 Lack of Motivation for exercise

Motivation is a crucial thing in maintaining sustained exercise, which in term associates to critical health results. Lack of motivation can widely be explained as, people may not be sufficiently interested in exercise, or value its outcomes enough to make it a priority in their lives [8]. Many individuals experience competing demands on their time from educational, career, and family obligations, possibly at the expense of time and resources that could be invested in exercising regularly.[9]

Embarrassment is another reason for not exercising that seems to be increasingly common. A nationwide survey by the International Health, Racquet, and Sportsclub Association in Boston found that about one third of the 1,700 respondents said they were intimidated to do work out at a gym, and most of them were female and overweight.[21]

1.3.2 Performance enhancement through Exercise Tracking

The proficiency to glimpse personalized data enhances cognition and facilitates consideration of exercise regimens. [10] However, capturing and tracking a regimen is challenging. Manual tracking is most accurate, but this is tedious for end users. Thus, various marketable and intellectual efforts have focused on automatically tracking and quantifying manual activity, the most pervasive being step count caught by a worn device (e.g., FitBit, Apple Watch, [11][12]). Nowadays, user devices can track some cardio and strength-training exercises using spacial applications[2, 3] . These applications generally depend on a wearable's inertial measurement unit (IMU) to monitor e.g., arm motion as users conduct different activities.

2 Gymaholic: http://www.gymaholic.me

3 Gymatic: https://itunes.apple.com/us/app/vimofit-auto-exercise-tracker/

1.3.3 Objectives

To this end, we propose an Artificially Intelligent Home Training Assistant, computer visions based mobile app that uses off the shelf mobile cameras to automate exercise tracking and provide performance analytics, such as repetition count, without any user or environment-specific workout or intervention. Instead of requiring each user to wear a sensor on their body, this app is an external single-point autonomous and low-cost solution, i.e., a smartphone camera placed in front of the person that can track 5 different exercises.

1.3.4 Target Audience

This outcome is particularly for whose novice peoples who want to do exercise at residence and keep track of their exercise and also for those who feel disconcert while enacting movement at the gym or by imploring help from others for inscribing exercise elements.

Chapter 2 Literature Review and Background Theory

AI is certainly one of the most compelling and incredible advancements in this day and age. AI is an apparatus for transforming data into information. Information is pointless except if we examine it and discover the patterns hidden inside. The hidden patterns and information about an issue can be utilized to predict a future event and even anticipate in complex decision making. Traditionally, software engineering combined human-created rules with data to create answers to a problem. Instead, machine learning uses data and answers to discover the rules behind a problem.13]

2.1 What is Machine Learning?

Machine learning is a part of research on artificial intelligence systems, trying to give information to computers through raw data, visual perceptions, and interacting with the world. Obtaining information permits computers to effectively generalize rules for the prediction of future cases.[14] In the previous decade, AI has given us self-driving vehicles, speech recognition, successful web search, etc.[15] Numerous analysts also think it is the most ideal approach to gain progress towards developing human-level intelligence. Machine learning is classified into different types, they're normally clustered by either learning style or by similarities in data

Learning Style	**Similarities in Data**
Supervised Learning	Classification
Unsupervised Learning	Regression
Semi-Supervised Learning	Clustering
Reinforcement Learning	

2.2 Supervised Learning

In the context of machine learning, supervised learning is the construction of algorithms that are able to produce general patterns and hypotheses by using previous training instances to predict the fate of future instances.[16] It is similar to the concept of a teacher in a class where the here the teacher is the data set and the student is the algorithm.

2.2.1 How does supervised learning work?

In supervised learning, the algorithm is expected to figure out how the input or features and output as labels are mapped together and related. The goal is to find an exact enough mapping function that when in future any unseen data is given, the function can anticipate the output which closes to the desired output. This function is achieved by the series of the iterative process, each time the algorithm makes a prediction, it is compared and the actual output and feedback is nourished to the algorithm, this process keep to going unless a the desired performance is not achieved[17]

2.2.2 Supervised learning algorithms

Common supervised machine learning algorithms are:

Classification	**Regression**
Support Vector Machine	Linear Regression,GLM
Naive Bayes Classification	Decision Trees
Random forest for classification	Random forest for regression
Nearest Neighbor	Logistic Regression

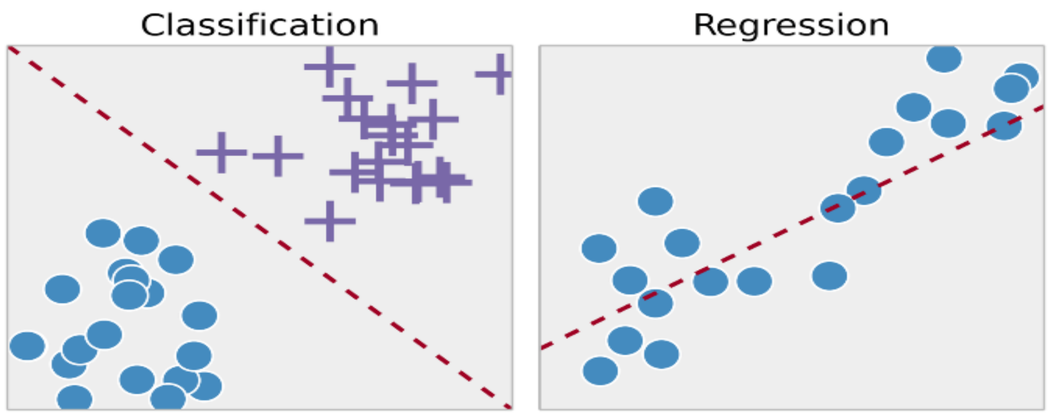


Fig 1: Comparison between classification and regression problems a) binary classifier b) linear regression model[20]

2.3 Unsupervised learning

In unsupervised learning, the algorithm is given the data without the corresponding output as the training data sets. Instead, algorithms are used to learn more and more interesting patterns about that data and use similar pattern matching techniques with unseen data and predict the output. [17]The popular application of unsupervised learning is clustering or predicting rules which describe input data in the best possible form etc.

2.3.1 Unsupervised learning algorithms

Unsupervised learning problems further grouped into clustering and association problems.

Clustering	**Association**
K-means Clustering	Apriori
C-mean Algorithm	FP- Growth
DBSCAN Algorithm	Hidden Markov Model
Principal Components Analysis(PCA)	Market Basket Analysis

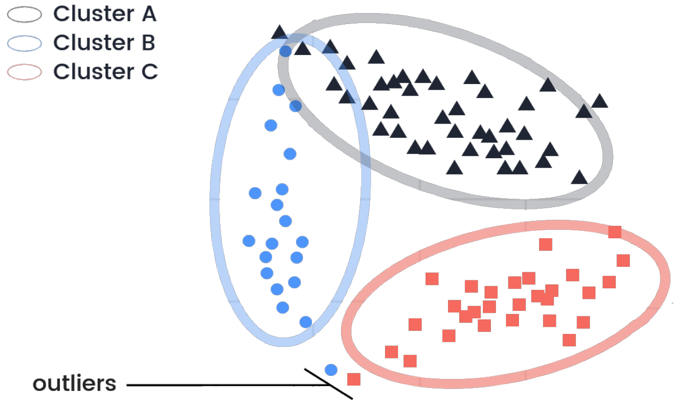


Fig 2: Graphical visualization of C-mean clustering algorithm[20]

2.3.2 Unsupervised learning applications

There are plenty of unsupervised learning application but the worth mentioning are Customer Segmentation for Marketing, Image Compression and Clustering of gene expressions[18].

2.4 Semi-supervised Learning

This type of learning technique is somewhere between supervised and unsupervised learners, we have a combination of both labeled and unlabeled input or train data set. In this first, we use unsupervised techniques to cluster similar data into separate groups and then use supervised learning techniques to predict the unlabeled data using label the complete data set. Then a better supervised learning model is created.[19]

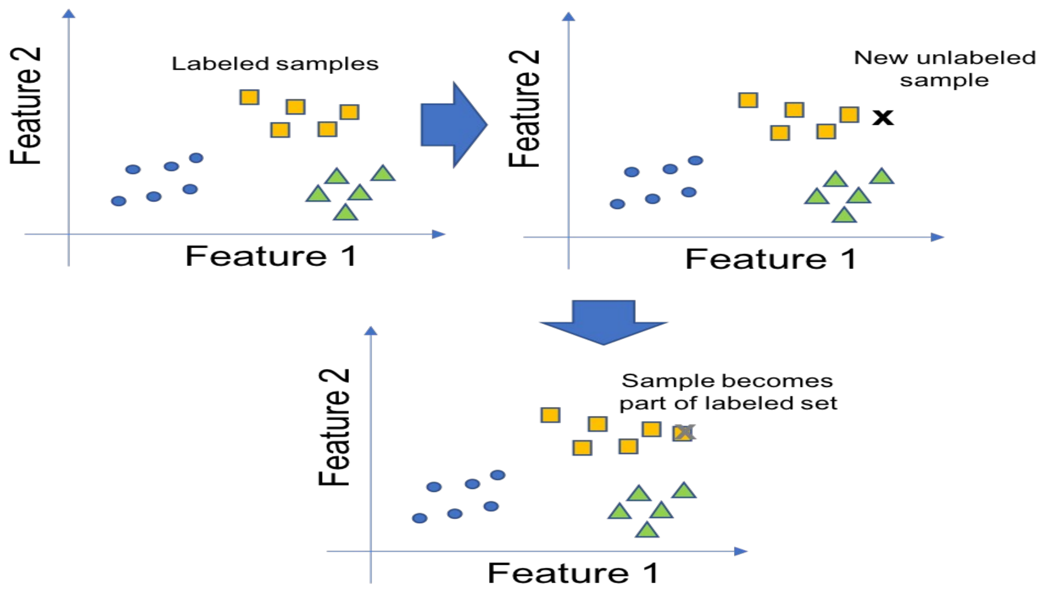


Fig 3: An unlabeled point is matched against labeled data to become part of the labeled data set.[17]

2.5 Reinforcement Learning

In reinforcement learning, no formal data is fed into the system for training, the algorithm learns from experiences lets assume a chessboard where you make a move if its corrects then you'll get a reward otherwise punishment in this way this is also known as learning from mistakes soon a time comes when he always make a right choice in a certain scenario. There are many applications of reinforcement learning like driverless self-autonomous cars, inventory management, robotics, etc. [17]

2.6 Perceptrons

Frank Rosenblatt an American psychologist gives the concept of perceptron an artificial neuron inspired by the Walter Pitts and Warren McCulloch research and won a noble prize. The concept behind the working of the perceptron is very simple, it takes several inputs and gives the single output:

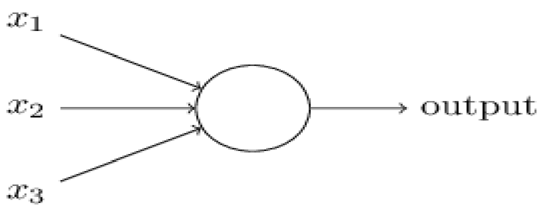


Fig 4: preceptron prepresentation[39]

so a perceptron could have one or more inputs depending upon the input vector. In the above figure, there are only three inputs x1, x2, x3. Frank proposed a simple solution to compute the output, he introduces the concept of the weighted sum concept and introduces the weights, real numbers expressing the importance of each input as compared to the desired output. This output is then fed into the threshold function commonly known as step function so the output would be either 0 or 1. the value of the threshold varies project to project.

$$\text{output} = \begin{cases} 0 & \text{if } \sum_j w_j x_j \leq \text{ threshold} \\ 1 & \text{if } \sum_j w_j x_j > \text{ threshold} \end{cases}$$

Fig 5: basic mathematical model[39]

So by changing the weights and threshold we can various models of decision-making artificial neuron's.[39

2.7 Complex perceptron Network

Human decision making more complex than the single perceptron as there are millions of neurons in our brain and few of then got activated and rest remain silent and gives the output decision, so as we are convinced that perceptron can work as the artificial neuron's so we can comply that a complex network of perceptrons could make quite subtle decisions a solve many real-life problems.

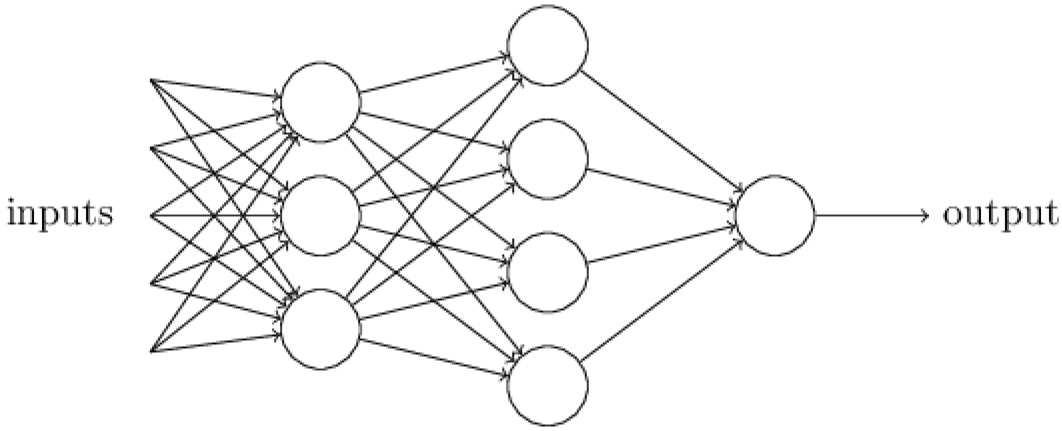


Fig 6: Complex preceptron network demostration[39]

From the above figure, we'll try to explain the concept behind the more and more complex perceptrons. The first column is called the input layer of the network set to make three simple decisions depending upon the summation of the weighted sum of inputs. Whereas in the second layer the more complex decisions are making as they depending upon the decisions of the previous layer and weight of the certain perceptions on decision making. So as the layer increase the decision making becomes more complex and more abstract as compared to the previous layer and similarly many layer network and be made to make to solve complex decision depending upon each other.[39]

2.8 Artificial Neural Networks

The basic structure of an ANN can be modeled as a complex network of perceptrons just as above. In ANN the multidimensional input is flattened and fed into the input layer which will later feed into the first then second hidden layer. The hidden layers will then make decisions from the previous layer and weigh up how a stochastic change within itself detriments or improves the final output. So in simple words, a small change in the weights and bias can somehow change the desired output. So these changes in the parameter weights and bias should also be regulated so that the network gets closer to the actual output. So the predicted results from the network approximate all the training input with minimum error. To device the mechanism that we can achieve the goal and regulate whether the changes in the weights are in the favor or not we define cost function. The perfect the value of weights and bias is adjusted in such a way that the cost function is approximately zero, $C(w,b)\approx 0$.

2.9 Convolution Neural Networks

Convolutional Neural Networks (CNN) are variants of Multi-Layer Perceptron (MLPs) designed to process two-dimensional (2-D) image (CNNs) are analogous to traditional ANNs in that they are comprised of neurons that self-optimize through learning. Each neuron will still receive input and operate (such as a scalar product followed by a non-linear function) - the basis of countless ANNs. From the input raw image vectors to the final output of the class score, the entire network will still express a single perceptive score function (the weight). The last layer will contain loss functions associated with the classes, and all of the regular tips and tricks developed for traditional ANNs still apply.

2.10 Difference Between CNN and ANN

The only notable difference between CNNs and traditional ANNs is that CNNs are primarily used in the field of pattern recognition within images. This allows us to encode image specific features into the architecture, making the network more suited for image-focused tasks - whilst further reducing the parameters required to set up the model.

2.11 Architecture of CNN

CNN’s are comprised of three types of layers. These are convolutional layers, pooling layers, and fully-connected layers. When these layers are stacked, a CNN architecture has been formed. A simplified CNN architecture for MNIST classification is illustrated in Figure

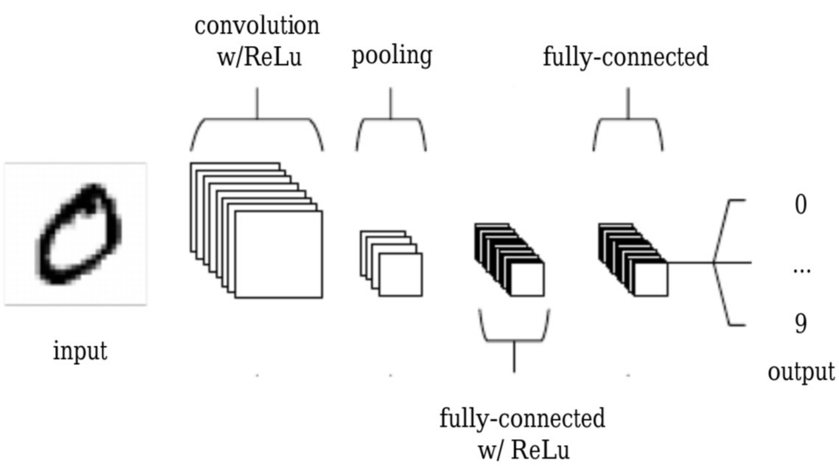


Fig 7: Working demonstration of a CNN model for Mnist data set

2.11.1 CNN Working

The working functionality of CNN can easy to understand if we divide it into four parts

2.11.2 Input layer

The image as an array of pixel values is fed into the input layer of the network.

2.11.3 Convolution layer

A kernel of a defined matrix is convole over the image array in simple word scalar product of the convolution matrix and the region of the image. Then the matrix is move over the image depending upon the defined stride and then again the scalar product is taken.

2.11.4 Activation Function

These linear multiplication values are passed through the activation function which passes values of only those pixel those pass the certain threshold. Most common of all are Sigmoid or Relu.

2.11.5 Pooling Layer

The pooling layer will then simply performdown sampling along with the spatial dimensionality of the given input, further reducing the number of parameters within that activation.

2.11.6 Dense layer or Fully Connected Layer

This layer works similariy to the ANN as all the output of the previous layer is connected to every single neuron and passed through the activation function.

2.11.7 Output Layer

The output layer's neurons are equal to the number of classes in our problem as only one neuron is active at a time.

Chapter 3 Machine Learning and Fitness

3.1 Applications in Health and Fitness

Artificial Intelligence (AI) in combination with Machine Learning (ML) and deep learning, is now taking the healthcare industry by storm. All three have emerged as important practical tools that can support health care organizations in optimizing their service provision, improving the quality of care offered, decrease the level of associated risk, and generating higher revenue.

Google managed to develop an ML algorithm to support the detection of cancerous tumors via mammograms[4] where else Stanford University researchers use Deep learning techniques for the identification of skin cancer[5]. Besides that ML is also used for drug discovery and manufacturing.

There are numerous AI fintech companies who are helping an individual to maintain his fitness. eg. Vida Health[6] have prepared machine learning application which helps individual finding the suitable coach or therapist based on their health goals, desired coaching style, availability, and location. There is also some intelligent application that uses machine learning for guidance on exercise, sleeps and eating habits to help keep the user stay on track with their health and fitness goals. Like Boltt Health app, FitWell. Some applications also use AI for the optimal nutritionist diet prediction as there are many factors apart from the food which contributes such as Sleep, workout routines, daily routine, genetics, and age.

In last, there are some applications which use AI for the statistical analyses of the workout performance using wearable gears, just as tech-house google and Fitbit teamed up to make health analysis app using Google analytic and Fitbit devices.

4 https://www.healthcareitnews.com/news/google-ai-platform-aids-oncologists-breast-cancer-screenings
5 https://news.stanford.edu/2017/01/25/artificial-intelligence-used-identify-skin-cancer/
6 https://www.vida.com/

3.2 Machine Learning and our project

Previous approaches for exercise recognition and tracking include using wearables, instrumenting equipment, and using computer vision. In this section, we discuss past work related to each of these key approaches.

3.2.1 Use of Wearables

IMUs track the user's exercise actions and compare them against the optimal posture and actions to guide then in improving their posture and exercise movement [26]. This Guidance depends only on tracing doper movements, but our motives contain identifying exercises as adequately. Hidden Markov model that can determine to classify precisely 9 exercises with 90% precision using numerous IMUs data [27][30]. Furthermore, myHealthAssistant [2][25] operated a substantial Bayesian classifier count repetition in the exercise and also identify 13 different exercises with 92% accuracy. All these systems use noise-free datasets and do not have any mechanism to noise avoidance and as we also know in real-word datasets always contain noise.

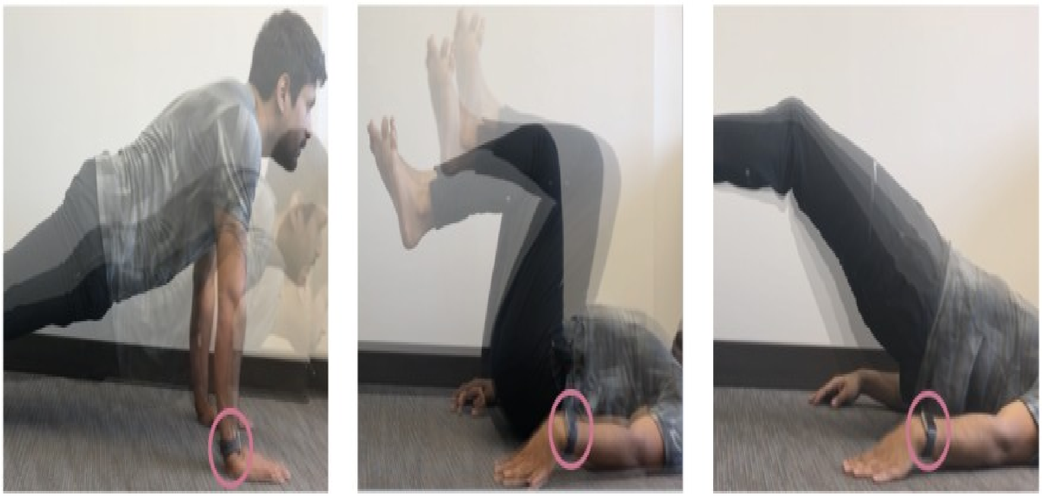


Fig 8: A wristworn IMU (circled in the photos) is not ideally positioned to monitor many exercises.[22]

In the past due to limited computational power, systems were only able to work in a constrained environment, but as the technology grows people start building frameworks that could work under real-world environments. To differentiate between exercises and count the number of exercises performed by the person, Crema et al. used a technique Linear Discriminant Analysis (LDA)on a series of data set gathered by a wearable IMU on 9 different exercises, reaching an average accuracy in the exercise detection around 85%.[28]. Recofit[29] is the most relevant research that uses inertial sensors to automatically track the exercise and also count the repetition it with an average accuracy above 90%, this system is built on the work previously done by Muelbauher et al.’s[31]. They use all resistance training exercises in their data set because of the repetitive nature of these exercises. To detect exercise performing segment and identify exercise from the stream of continuous inertial data, they use features based correlation techniques. Recofit[29] tries to withstand some problems related to sensor placement using dimensionality reduction but still not feasible to track workouts that do not implicate the limb where the sensor is chafed. Furthermore, as these methods compel devoted worn sensors per individual, this reduces the system scalability as compared to the singles point inferring outcome.

3.2.2 Sensors / Instruments on Equipment

Extra sensors can be attached to the workout equipment for tracking the exercise. In this view Velloso et al. placed IMU on both the equipment and on the user [32], [33], and design a system that was used to train a novice user. The data is collected from the experts using the combination of both IMUs and Kinect for the exercise movement. Then in the same setting, a novice user performs exercise after he is finished the system compares both the values and generate workout progress report. Ding et al. came up with a new idea he attached RFID tags with each equipment and the help of a back scatter signal’s strength tries to recognize different free-weight exercises. The research was experimented on 15 volunteers and suggest that it applies to a variety of different free-weight exercises.[34] It was able to recognize 10 different exercised with an accuracy of 90%.

Nowadays there are many products available for the users which provide external sensors that can be attached to the user's free-weight equipment and provide repetition count exercise classification and also recommend posture and weight suitable for you such as Bazifit[35].

3.2.3 Using Computer Vision

With the increase in the quality of cameras, day by day computer vision has also considered an important role in machine learning due to its unique stand-alone solutions. So research has also tried CV technique in rehabilitation and found quite reasonable results such as with help of depth camera sensor recognition of in-home physiotherapy exercises[36].of tracking exercises for the tele-rehabilitation[37]. So in all these types of tracking of exercise, the 3D moment of the user during exercise is important so that's why the use depth camera. For the past years, Microsoft has partnered with you to move to create an augmented mirror that users do physical movement analysis[38]. Tempo’s1 Microsoft Kinect-Esque motion sensors scan you 30 times per second and notify you with the help of 42’’ long screen if your form is wrong.

3.2.4 Conclusion

The solution provided with the help of IMU is considered optimal in some specified exercises and not feasible for the most, this is due to the inappropriate positioning in that exercise. Maurer et al led an exercise classification experiment on seven different workout exercises through putting acceleration sensors on the forearm, wrist, thigh, ankle, and compare results while placing these sensors on the wrist, belt, necklace, trouser pocket, shirt pocket, and bags. Discovered that recognizing ascending motion, for example, climbing stairs is more precise when the IMU is attached to the bag than when placed to a person's shirt[19]. Similarly, we found out that the data is inadequate when is attached wearable to the part of the body which is inactive during exercise(e.g., leg squeezes, push-ups).

An alternative is to instrument the exercise machine rather than the user, but that is too intrusive and also makes free-weight and body weight exercises harder to track. This presents a need for a method to robustly identify and track a wide range of exercises that a user might perform while maintaining the seamlessness offered by wearable devices.

Chapter 4 Requirement Analysis

Requirements Analysis is the process of defining the expectations of the users for an application that is to be built or modified. After the detail consultation with supervisor and the project time span we have focused on the following requirements.

4.1 Functional Requirement

This section will describe the most general and significant features of the project that were identified uniquely for the project and on which the project was scoped.

4.1.1 Video input stream

The system should be able to get real time video or select video from the gallery and give analysis on the video in real time.

4.1.2 Server

There should be a centralized resource or service which gets the input video and where are model is deployed to perform interference on the video and give feedback to the user in real time.

4.1.3 User Visualization

There should be some visualization interface (web, App) for the user interactions where they can see the model analysis on the video.

4.1.4 Real time Evaluation

The system should be fast enough to do analysis on the video frame by frame so that third person can also see analysis of his friend during exercise or him self can see the analysis of previously recorded video selected from gallery.

4.1.5 Exercise Analysis Specification

1. **Detect person**

   The system should be able to distinguish between person performing exercise and the background.

2. **Exercise Classification**

   The system should be able to distinguish between 5 different exercises (Abdominal Crunches, Bench Press, Push ups, Side Plank, Squats)

3. **Repetition count in Exercise**

   As for the proof of concepts we are only considering repetition count on Bench Press once we are done we'll extend same approach on all exercises.

4.2 Non-Functional Requirement

4.2.1 Performance Requirements

The model should be responsive and seek processing timely so that the should not be any type of delay during video processing. An alert should be generated if occurred with some problem.

4.2.2 Security Requirements

Being a product with a critical data production and storage, it is very necessary to look into security of the data on server side as well as in the user side.

4.2.3 Software Quality Attributes

Following attributes are only applied when the application is publicaly available to public audience.

1. **Availability**

   The product should be available online and could be accessible to all.

2. **Maintainability**

   The development cycle of this product was carried in such a way that the changes or updates could easily be made at any running stage. All of the updates would be centralized making the software maintenance convenient and easy.

4.3 Hardware Requirement

The hardware requirements for the projects are listed below:

1. A good quality video recording camera.
2. A moderate computational power processor
3. A display screen for showing visual analysis on video

Now a days smart phones are commonly used devices and fulfill all our hardware requirements, hence proved to be favorable solution for the specified problem.

4.4 Usecase Diagram

Use case diagram is basically used for high level understanding of the application and its uses. For deep understanding, additional diagrams and documentations can be used to provide complete technical and functional use of the system. More over in use case diagram the interacting users are known as actors.

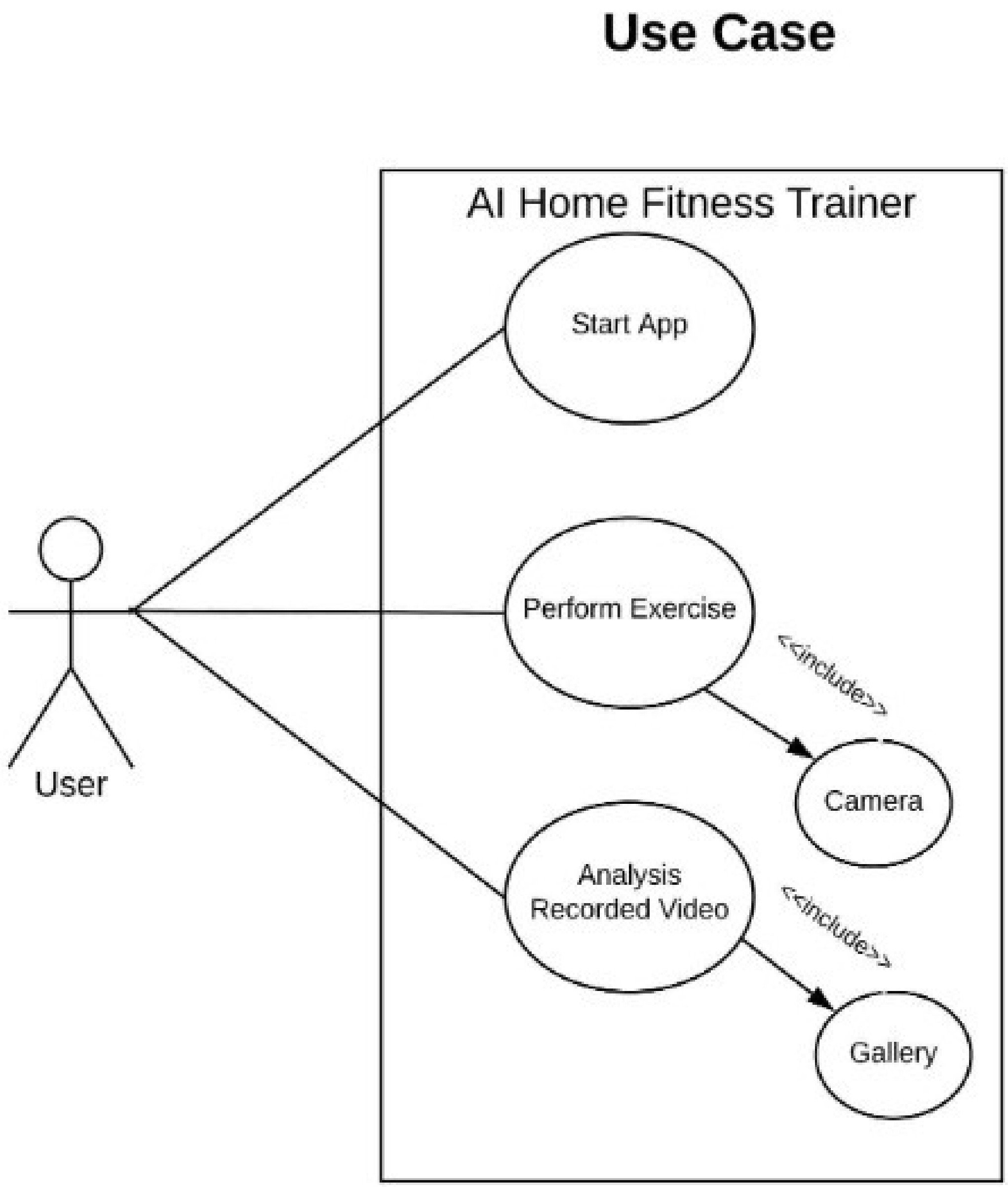


Fig 9: Project use case diagram

Chapter 5 Methodology

The development of this project involves extensive use of different techniques from both Computer Vision and Artificial Intelligence. Two solution are proposed and implemented for the problem of exercise detection and the repetition count. One use simple sequential model to detect exercise and the other one use two machine learning models 1) posenet pre-build tensorflow model for the 17 body joints for pose estimation then after doing some prepossessing pass through 2 1D CNN sequential model for the exercise detection.

5.1 Sequential Model

The goal of our project is to classifying the different exercise when people along with the exercise equipment is clearly visible. The following pipeline is followed for classification: Data Collection, Data Preprocessing,Training of model and Testing Model and results. Will now describe our pipeline.

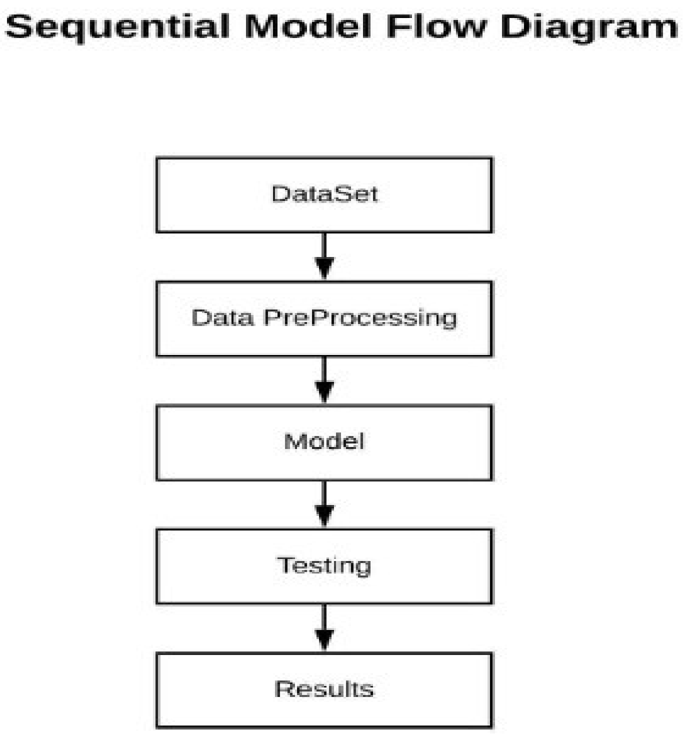


Fig 10: First proposed solution workflow

5.1.1 Data Collection

Our Data set consist of 5 different exercises performing by 7 different people at Namal College Gym. These exercise are Abdominal Crunches, Bench Press, Push Ups, Side Plank and Squats. We use Samsung C7 phone camera at the resolution of 1280 × 720 to record 30 frames per second(fps) video. After dropping periods when the exercise was not performing, we had following data set collected.

Exercise	Video length(sec)	frames
Abdominal Crunches	40	1206
Bench Press	57	1707
Push ups	36	1101
Side Plank	23	692
Squats	48	1436

Fig 11: Illustration of video to frames

5.1.2 Data PreProcessing

We use different Open CV technique for the date preprocessing first of all we resize all the frames to 128 × 228 keeping dimensionality scale constant and still human can classify through naked eye. And labeled each Image with there exercise. Keeping in view that there is a negligible change between 3 consecutive frames so we down size the data and pick one frame after every two frame. So aur data set become 2048 images in total.

5.1.2.1 Dataset split

We apportion the data into training and test sets, with an 80-20 split on each exercise so with the train data contain in total 2048 Images whereas the test data consist of 410 Images of all 5 exercises.

5.1.3 Training Model

We made sequential deep learning model to efficiently and accurately classify exercise.

5.1.3.1 Sequential Model

We use tensor flow library to build an sequential model. It is a stack of layers, each layer has weights that correspond to the layer that follows it. It is called sequential as it has a single path for forward and back-propagation. There are different type to layers like Conv2d, MaxPooling2d, Dropout, Flatten, Dense, Activation etc

1. **Conv2d:** This layer is to convolve an convolution kernel on the input to produce a tensor of outputs conv2d is for spatial data where as conv1d in for temporal data. There are many input parameters in this layer such as padding, kernel and if its the first layer then we have to define the input size of the data set.

2. **Activation Function:** the activation function is used to give non linearity in the model we here use ReLU (Rectified Linear Unit) function. It’s cheap to compute as there is no complicated math. The model can therefore take less time to train or run. Mathematically, it is defined as $y = \max(0, x)$

3. **MaxPooling2d:** This layer is used reducing its dimensionality and enhance the features contained in the sub-regions binned. The parameters are sub-regions matrix size

4. **Dropout:** This layer used for preventing over fitting and regularization of data. With the addition of this layer randomly selected neurons are ignored during training and so do not update the weights during back propagation.

5. **Flatten:** This layer plays a simple role just reshape the input to single row of equal number of elements in the input matrix.

6. **Dense:** A simple name of this layer is fully connected layer all the output of previous layers are connected to the input of all the neurons in the dense layer specially use in the ending of the model to converge the solution.

5.1.3.2 Architecture of model

```
model = Sequential()
model.add(Conv2D(32, (3, 3), padding='same',
                 input_shape=(128, 228, 3)))
model.add(Activation('relu'))
model.add(Conv2D(32, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Conv2D(64, (3, 3), padding='same'))
model.add(Activation('relu'))
model.add(Conv2D(64, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(512))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(5))
model.add(Activation('softmax'))
```

Fig 12: Architecture Model of First solution

5.1.3.3 Hyper parameters

1. **Learning rate:** The learning rate of the model is set to 0.001.
2. **Epochs:** The model was run to 65 epochs
3. **Batch Size:** The batch size for the weight update is set to 32.
4. **Loss Function:** Categorical cross-entropy is used to calculate the average difference between the predicted and actual probability distributions for all problem classes. It is minimize on the completion of each batch and the best score is 0.

$$-\frac{1}{N}\sum_{i=1}^{N}\log p_{model}\left[y_i \in C_{y_i}\right]$$

Fig 13: Categorical cross-entropy loss function

5. **Adam Optimizer:** It's a back-propagation algorithm in which learning rate is maintained for each network weight (parameter) and separately adapted as

learning unfolds. Where as in Stochastic gradient descent learning rate is maintained for weight updates through out the training.

5.1.4 Testing

We use confusion matrix for summarizing performance of the classification problem. And calculate accuracy, precision and recall as if you have unequal number of sample of each class in testing data then the accuracy would be misleading.

5.1.4.1 Confusion Matrix

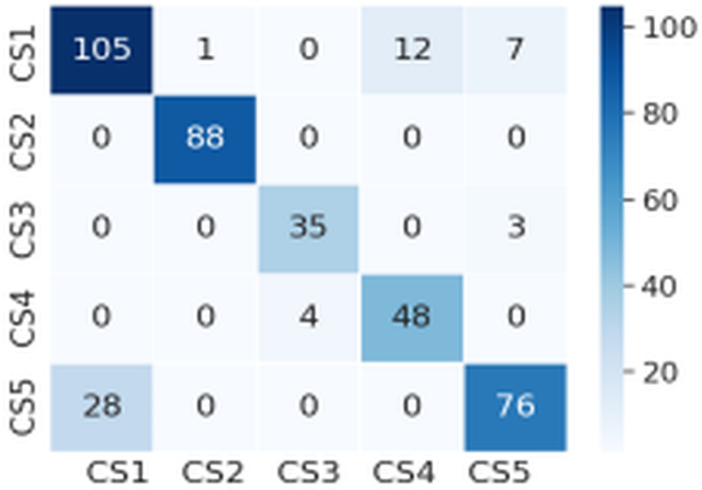


Fig 14: Confusion Matrix of first approach

5.1.4.2 Accuracy

It is simply the ratio of true positive and number of test samples it is useful if the testing data set has equal number of samples in each class.

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{True Positives} + \text{False Positives} + \text{False Negatives} + \text{True Negatives}}$$

Fig 15: Formula for Calculating Accuracy

How ever my model accuracy is **86.48%**

5.1.4.3 Precision

Precision is ratio of correctly predicted to the sum of both corrected predicted and false predicted true samples in test data.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

Fig 16: Formula for Calculating Precision
How ever my model precision is **87.18%**

5.1.4.4 Recall

Its the ratio of all true predicted and summation of true sample predicted true and false sample predicted false.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

Fig 17: Formula for Calculating Recall
How ever my model Recall is **88.29%**

5.1.4.5 F1 Score

The F score is defined as the weighted harmonic mean of the test's precision and recall. Simply its is the measure of test accuracy.

$$\text{F1 Score} = 2 \cdot \frac{\text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}}$$

Fig 18: Formula for Calculating F1 Score
How ever my model F1 Score is **87.73%**

5.1.5 Video Testing

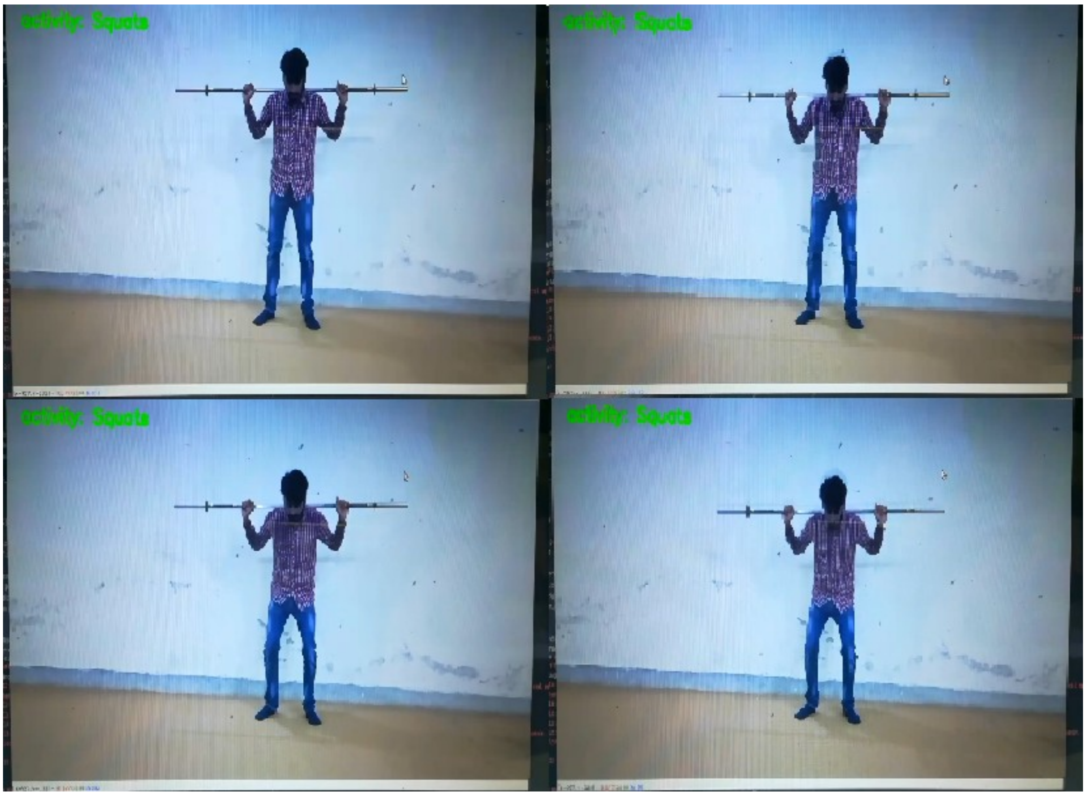


Fig 19: Testing First Model on video to detect Squarts exercise

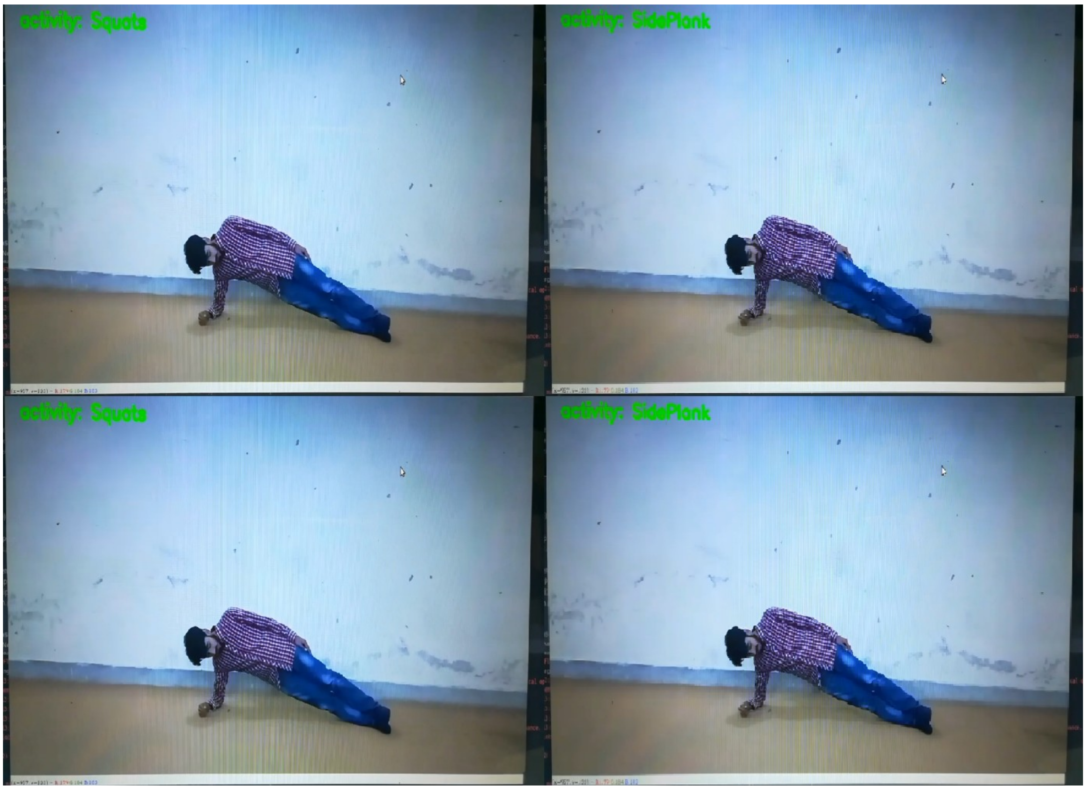


Fig 21: Testing First Model on video to detect Side Plank Exercise

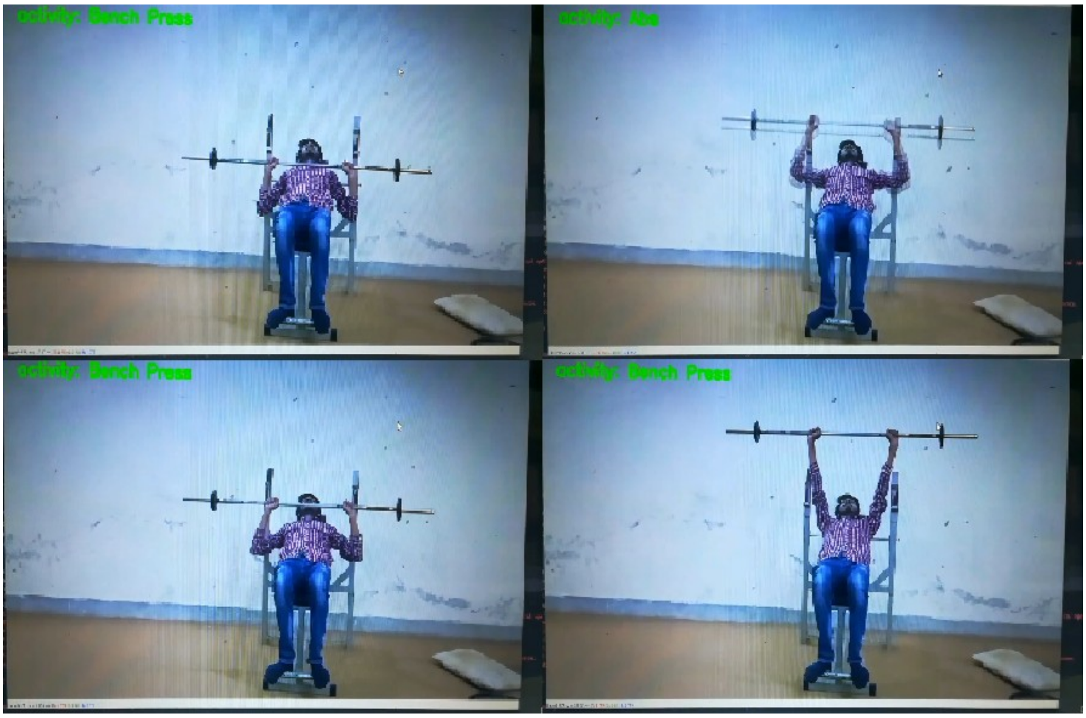


Fig 22: Testing First Model on video to detect BenchPress Exercise

Fig 23: Testing First Model on video to detect Abdominal Crunches Exercise

5.2 Extending Image Classification model to repetition count

We try to extend same approach fro the repetition count and found good results. We divide our existed Bench press data set to 3 classes.

1. **Up class:** Having all the images when arm of the user is stretched, i.e. the instrument (weights) is at the maximum height.
2. **Down class:** Having all the images when arm of the user is squeezed, i.e. the instrument (weights) is at the minimum height close to chest.
3. **Middle class:** Having all the images when arm is in between maximum and minimum height.
4. **Background class:** A simple class having white background images as the no one is performing.

And build a sequential model with 8 class and test on the video for the testing of results.

5.2.1 Architectural Model

```
model = Sequential()
model.add(Conv2D(32, (3, 3), padding='same',
                 input_shape=(128, 228, 3)))
model.add(Activation('relu'))
model.add(Conv2D(32, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Conv2D(64, (3, 3), padding='same'))
model.add(Activation('relu'))
model.add(Conv2D(64, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())

model.add(Dense(512))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(8))
model.add(Activation('softmax'))
```

Fig 24: Architecture of model after extending First approach

5.2.2 Video Testing

Fig 25: Testing First approach extended model on video to detect Bench press exercise and repetition count (count=0 right corner)

Fig 26: Testing First approach extended model on video to detect Bench press exercise and repetition count (count=1 right corner)

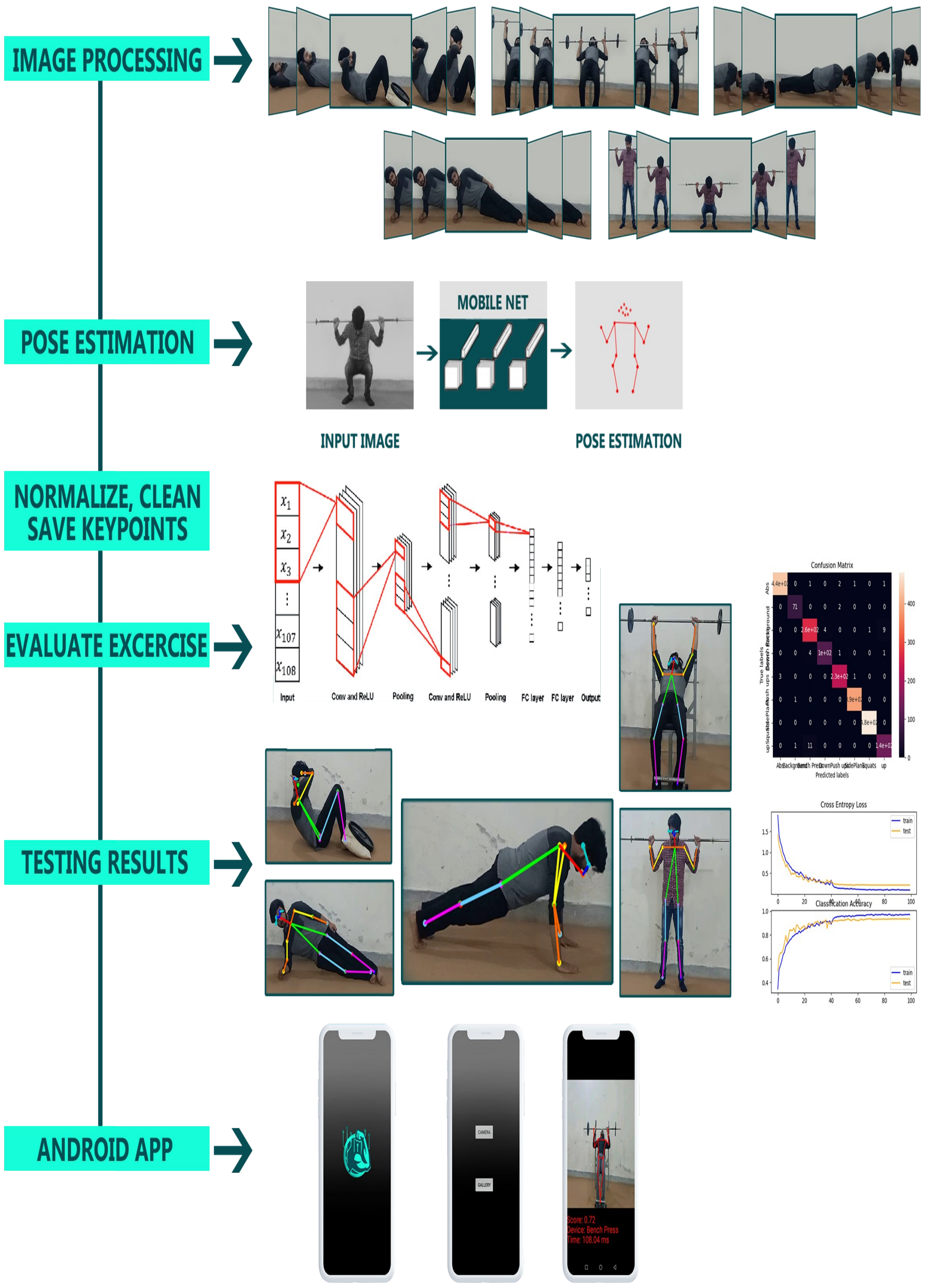
Fig 27: Testing First approach extended model on video to detect Bench press exercise and repetition count (count=2 right corner)

5.3 First Approach Sequential Model Limitation

1. **Environmental Constrains:** The model was working fine in same lighting, same distance from the camera and the user, same background setting and as we all know that in real time its not possible.
2. **Spacial Constrains:** Mobile phone has a limited memory and in this setting model size become too large.

5.4 Second Approach Using Pose Estimation

Now a day there are many solution preexist that can track the human skeleton pose from RGB image using deep learning such as OpenPose [7, 35], V NECT [20] and Tensor flow PoseNet [11]. we also try to build our system using this approach following is the work flow for this proposed solution.



5.4.1 Image Processing

We increase our data set by adding some background images and some new data and further more we divide the bench press to thee different classes now the the total data set break down is as following:

Exercise	Video length(sec)	frames
Abdominal Crunches	40 + 46	2584
Bench Press	57+7	1933
Push ups	36+ 3	1184
Side Plank	23+82	3805
Squats	48+27	2253
Background	11	342

5.4.2 Data PreProcessing

We use different Open CV technique for the date preprocessing first of all we resize all the frames to 257 × 257 keeping in view the model input image size. And labeled each Image with there exercise. Keeping in view that there is a negligible change between 3 consecutive frames so we down size the data and pick one frame after every two frame. So our data set become 3919 images in total.

5.4.2.1 Dataset split

We apportion the data into training and test sets, with an 80-20 split on each exercise so with the train data contain in total 3362 Images whereas the test data consist of 672 Images of all 5 exercises.

5.4.2.2 Dividing bench press in further two classes

Keeping in view that we also count repetition in the bench press we further make two classes. 1) up-class 2) down-class and 3) middle-class just as we previously did in the extension of first approach. Now our final data set for the training become.

Exercise	Images
Abdominal Crunches	718
Background	95
Bench Press	278
Bench Press(Down class)	110
Bench Press(Up class)	149
Push ups	329
Side Plank	1057
Squats	626

5.4.3 Pose Estimation

Its simply refers as a computer vision technique trough which we can able to detect human body joints in a visible image or a video stream. There are two visions of posenet available one is for single human body joints detection where as other is multi human body joints in a single image. We here use single human as our application is for single user exercise detection and repetition count. Pose estimation process is as following:

1. **input Image:** An RGB image of size 257 × 257 is fed into the convolution neural network.

2. **Output Tesor:** The pose net model outputs a tensor having 4 elements, pose, pose confidence score, keypoint positions and keypoint confidence score.

    1. **Pose:**Detect the pose object in the image that can contain the keypoints

    2. **Pose confidence Score:** it contains the confidence score of each pose detected it varies from 0 to 1, so we can neglect the pose when the model don't gives strong confidence score against that pose.

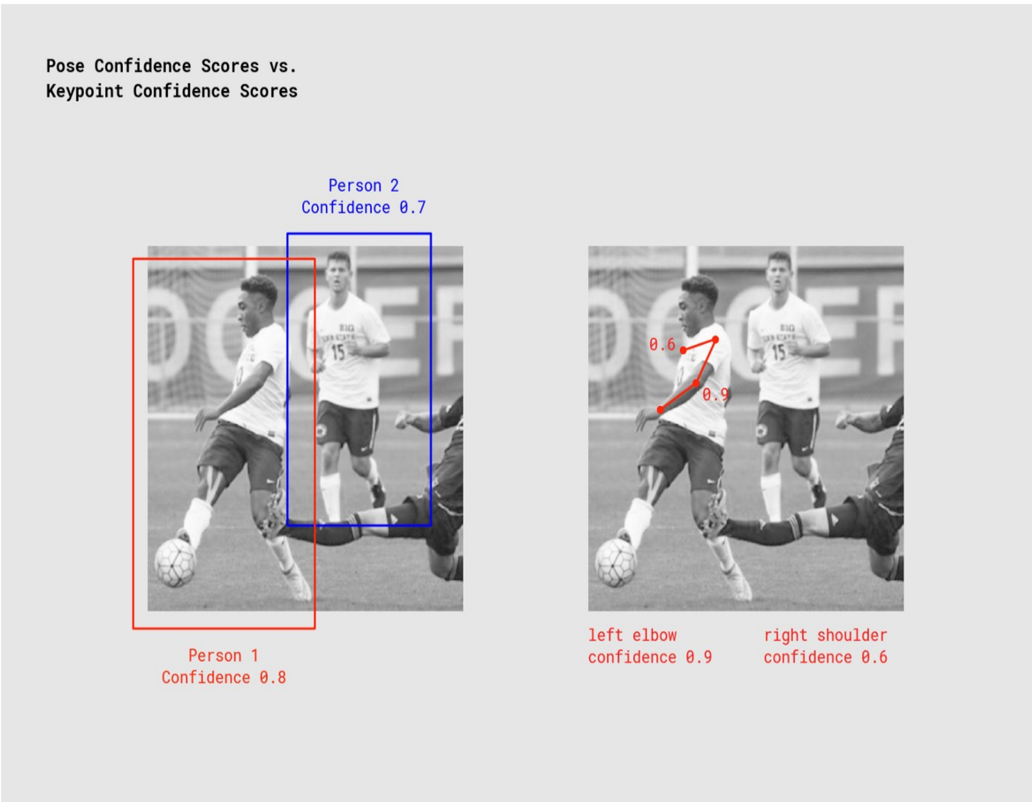


Fig 28: Shows two differnt pose detected and confidence score of each pose[24]

3. **Keypoints position:** Its detects the 17 different body key joints like (left shoulder, right foot, nose etc) position in the image.

4. **Keypoints confidence Score:** It contains the confidence score of each body joint detected in the image, it is also varies from 0 to 1, so we can neglect the body joint when the model don't gives strong confidence score against that joint.

Fig 29: Identifying 17 body joints with there confidence score[24]

5.4.3.1 Passing Training Images through PoseNet

After passing all the training images through posenet we store position of all the body keypoints detected. As for optimal results its adviced to use keypoints having confidence score above 0.5 but as in my data set it was hard to detect body joints due the large distance between camera and the human so we consider all the keypoints detected with the confidence score above 0.

5.4.4 Normalize, clean and prepare Datset

5.4.4.1 Clean body joints

Where there is no body joint is detected in the image we discard that image as it will effect our model learning.

5.4.4.2 Normalization

For the normalization of the body key points we use two techniques

5.4.4.2.1 Translational Invariance Handling

This means if we translate the input of the CNN , in our case human performing any where in the frame, our CNN should be able to classify exercise. In order to handle this we calculate the euclidean distance from nose (considering pivot in human body) to all the key body joints detected separated and store then in a excel sheet along with the image label.

NOSE	LEFT_EYE	RIGHT_EYE	LEFT_EAR	RIGHT_EA	LEFT_SHO	RIGHT_SH	LEFT_ELB	RIGHT_EL	LEFT_WRIS	RIGHT_WR	LEFT_HIP	RIGHT_HIP	LEFT_KNEE	RIGHT_KNI	LEFT_ANKL	RIGHT_ANKLE	Lable
0	63.63175	79	32	32.24903	0	42.52058	163.6154	83.81527	178.059	129.0736	98.0816	192.3148	15	0	28	0	Abs
0	64.00781	0	32.24903	82.8734	58.24088	28	227.8991	282.9311	339.8897	204.9244	0	0	0	0	16	0	Abs
0	41.7732	111	125.3994	73.10951	139	159.3895	151.7135	241.5802	333.7319	143.7533	236.6622	345.1377	16	16	28	28	Bench Pres
0	112.0089	184.3177	167	31	96.02083	56	137.0036	231.3655	361.4651	112.5389	204.9219	345.1377	32.24903	0	32.24903	57.28001	Bench Pres
0	64.00781	137.6118	121.4619	63.13478	155.5699	115.2476	136.9708	226.0465	364.3295	127.8828	220.6581	361.0111	16	31.38471	0	27	Down
0	73.10951	125.3994	104.2017	73.10951	137.0036	127.6323	151.7135	241.5802	333.7319	143.7533	236.6622	330.1893	16	28	28	28	Down
0	95.38344	95.38344	150.0467	104.2017	0	0	250	339.7058	504.8218	250.5115	337.2907	477.0461	16	0	42.52058	0	Push ups
0	73.10951	84.5281	113.8991	84.29116	84.5281	0	223.5733	334.383	473.2705	254.3796	334.383	473.2705	16	0	31	0	Push ups
0	68.02941	112.5389	158.4929	84.5281	139	225.1444	191.6377	311.713	419.1145	204.6192	320.1125	435.3321	28	16	28	32.24903	SidePlank
0	54.70832	82.8734	128.0976	84.5281	139	223.5039	159.3895	279.5085	419.1145	200.5841	293.4621	413.7656	28	16	31.76476	32.24903	SidePlank
0	63.13478	113.8991	115.2476	64.00781	146.219	121.4619	158.3193	225.8008	302.0497	158.4929	226.0465	302.2334	31.38471	16	31.38471	32.24903	Squats
0	83.63014	150.0467	127.6323	84.29116	167.8362	152.6106	220.6581	348.3691	457.3194	220.7827	345.1377	457.4407	16	32.24903	27	28	Squats
0	64.00781	104.2017	167.1706	64.00781	84.5281	167.1706	151.7135	241.5802	333.7319	143.7533	236.6622	345.1377	0	16	28	28	up
0	54.70832	89.5377	137.6118	72.34639	84.34453	145.4545	143.7533	241.5802	380.1473	143.5618	236.546	376.9682	0	27	32.24903	31.38471	up

Fig 30: The sample of two images per exercise body keykoints after Translational Invariance Handling

5.4.4.2.2 Scale Invariance Handeling

This mean that the object should be same if increase the scale or we decrease the scale the CNN model should be able to detect the object. In our case the camera distance from the user is not fixed so if it comes closer then the user scale larger and if it goes back then the user scale lesser so to handle this problem we calculate the euclidean distance between left shoulder and the left hip. Then divide all the key point distances with half of that distance. And store all data points with the exercise label.

$$\text{Key point}(i) = \frac{\text{EuclidenDis}(\text{Keypoint}(\text{Nose}), \text{KeyPoint}(i))}{\frac{\text{EuclidenDis}(\text{Keypoint}(\text{left shoulder}) \text{KeyPoint}(\text{left hip}))}{2}}$$

$$\text{Key point}(i) = \frac{2 \times \text{EuclidenDis}(\text{Keypoint}(\text{Nose}), \text{KeyPoint}(i))}{\text{EuclidenDis}(\text{KeyPoint}(\text{left Shoulder}), \text{KeyPoint}(\text{left hip}))}$$

Fig 31: Formula for handling Scale and Translational Invariance

NOSE	LEFT_EYE	RIGHT_EYE	LEFT_EAR	RIGHT_EA	LEFT_SHO	RIGHT_SH	LEFT_ELBO	RIGHT_EL	LEFT_WRIS	RIGHT_WR	LEFT_HIP	RIGHT_HIP	LEFT_KNEE	RIGHT_KN	LEFT_ANKL	RIGHT_ANKLE	Lable
0	0.284406	0.170644	0.580075	0.16086	1.217315	0.917209	1.934814	1.954779	3.023293	1.297119	3.103552	3.434146	4.892125	1.478911	5.828844	5.892096	Abs
0	0.226414	0.054915	0.280011	0.173641	0.85777	0.591433	1.667435	1.70324	0.233004	2.486381	2.747941	2.855574	1.390335	2.241461	3.487864	3.396321	Abs
0	0.131906	0.166829	0.212725	0.263811	0.85078	0.876973	1.478335	1.398519	0.750848	2.176327	2.732797	2.754329	4.956022	4.925051	8.318084	8.318674	Bench Press
0	0.285031	0.180248	0.324994	0.513847	0.97288	1.149049	2.156729	2.26062	1.326301	1.682718	2.845406	2.767265	5.040059	4.986137	8.859683	8.863125	Bench Press
0	0.099975	0.543963	0.099975	0.447117	0.720977	0.733317	1.880038	1.86345	1.898013	2.124164	2.631553	2.595605	4.021551	3.999642	6.036082	6.049496	Down
0	0.124158	0.277686	0.043903	0.570739	0.652662	0.682955	1.721743	1.867808	1.842871	2.131446	2.564856	2.557304	3.93195	3.980683	5.926902	5.930985	Down
0	0.075426	0.075426	1.527229	1.464474	1.359783	1.433097	0.075426	1.431136	0.43981	0.754262	3.20893	3.305853	4.216473	4.230578	0.226278	0.377131	Push ups
0	0.235667	0.235667	0.372667	0.372667	0.5	0.601	1.166667	1.343667	1.833333	1.840833	2.166667	2.173	3.3375	3.370667	4.859167	4.879167	Push ups
0	0.189627	8.337252	0.341851	8.58903	0.61809	0.437091	0.616289	0.335214	4.628662	0.189627	2.523277	2.382147	1.463497	1.444155	4.904049	4.82175	SidePlank
0	1.59857	2.602324	4.478999	2.174441	11.98427	13.09026	9.383021	19.78981	5.764075	4.66059	13.26935	15.82949	24.47757	22.15693	38.10795	38.66202	SidePlank

Fig 32: The sample of two images per exercise body keykoints after Translational Invariance and Scale Invariance Handling

5.4.5 Evaluating Exercise

After passing all the images through PoseNet and recording all the body key points euclidean distances from pivot Nose along with the exercise label, and then applying Normalization techniques we have prepared data set for the supervised 1Dimention Convolution Neural Network.

5.4.5.1 Architecture of Model

```
model_m = Sequential()
model_m.add(Reshape((17, 1), input_shape=(17,)))
model_m.add(Conv1D(32, 2, activation='relu', input_shape=(17,1)))
model_m.add(Conv1D(64, 2, activation='relu'))
model_m.add(MaxPooling1D(1))
model_m.add(Conv1D(128, 2, activation='relu'))
model_m.add(Conv1D(256, 1, activation='relu'))
model_m.add(GlobalAveragePooling1D())
model_m.add(Dropout(0.5))
model_m.add(Dense(8, activation='softmax'))
# print(model_m.summary())
```

Fig 33: 1D CNN Model Architecture

1. **Conv1D Layer:** This act similarly as I have explained in my previous sequential model the only difference is that here the kernel is of like 1d array and convolve horizontally on the input data.

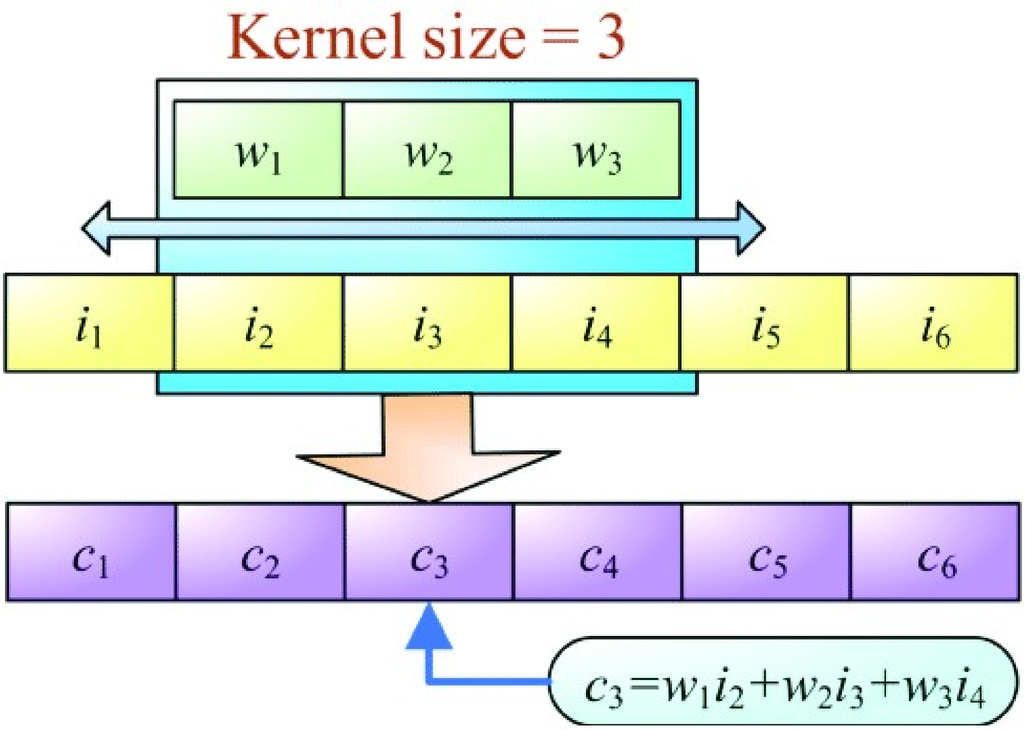
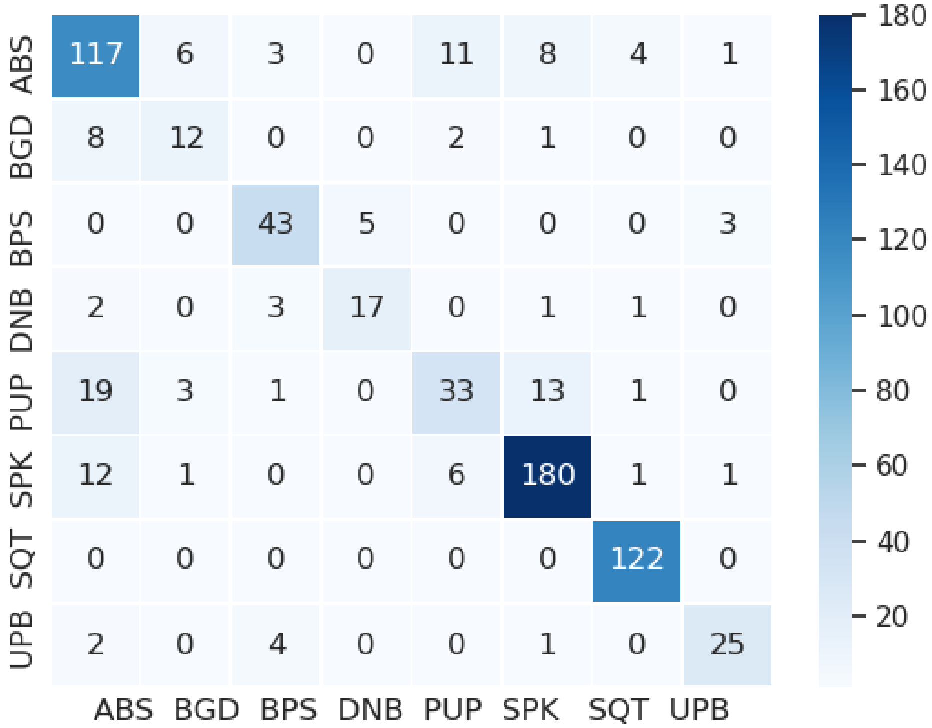
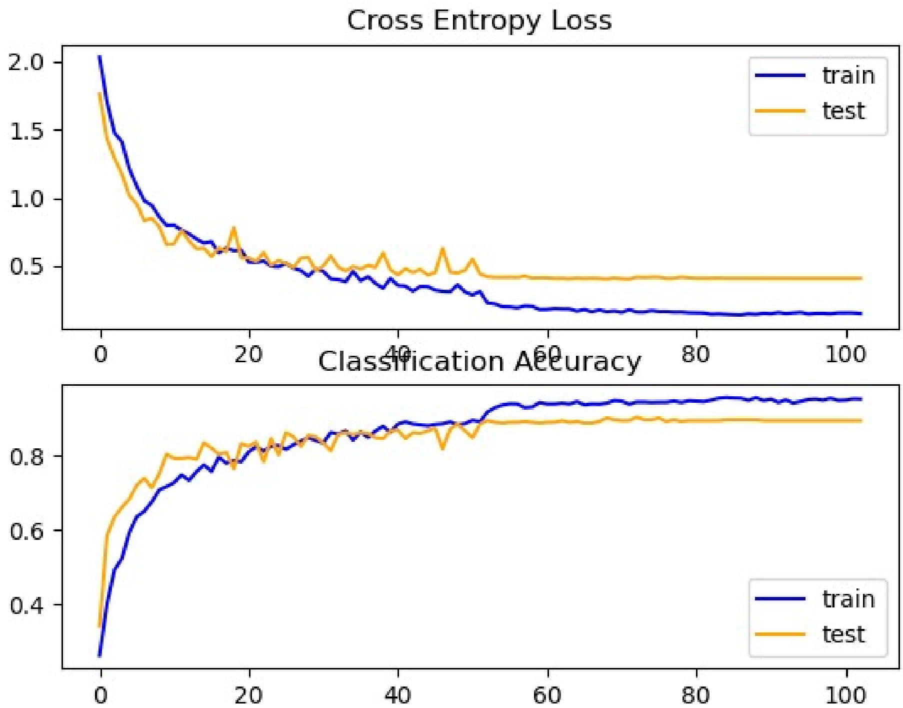


Fig 34: 1D convolution operation simulation[23]

5.4.6 Testing and Results

5.4.6.1 Confusion Matrix





5.4.6.2 Accuracy, Precision, Recall and F1 Score

Testing Benchmarks	Results(percentage)
Accuracy	94.46
Precision	94.41
Recall	93.53
F1Score	93.99

5.4.7 Video Testing

On same environment


activity: Bench Press
1.0


activity: Push ups
0


activity: Push ups
0


activity: Push ups
0


activity: Squats
0


activity: Squats
0


activity: Squats
0


activity: SidePlank
0

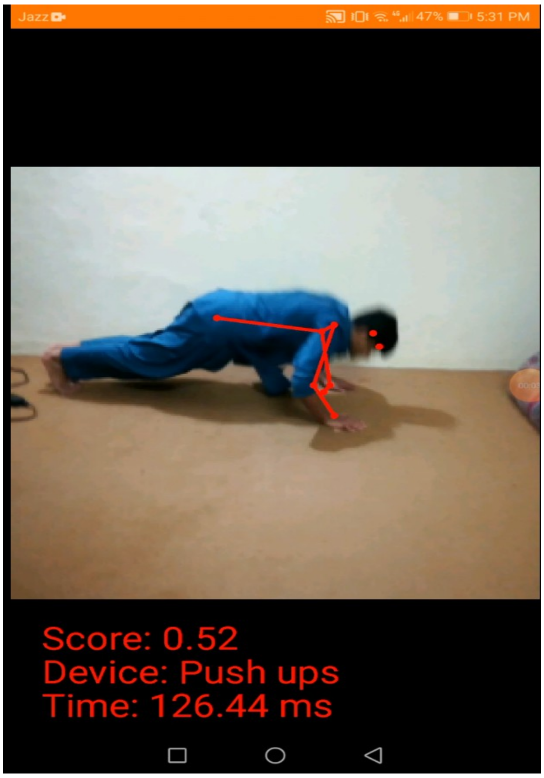

activity: SidePlank
0

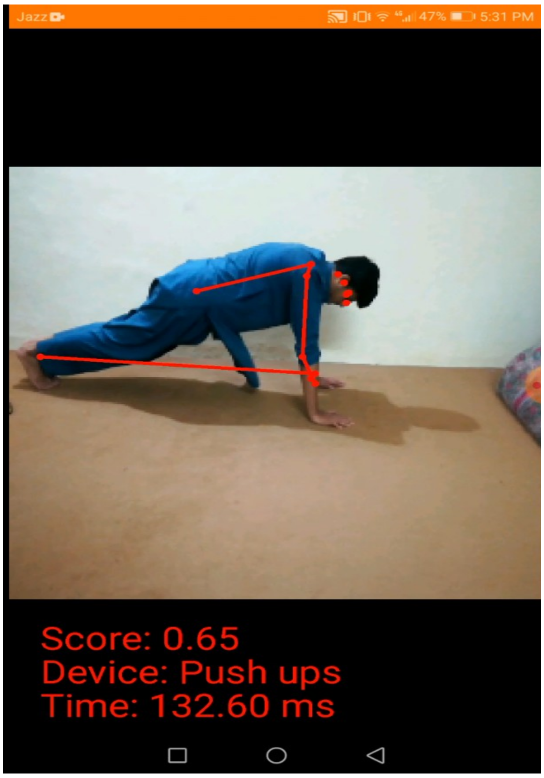

activity: SidePlank
0

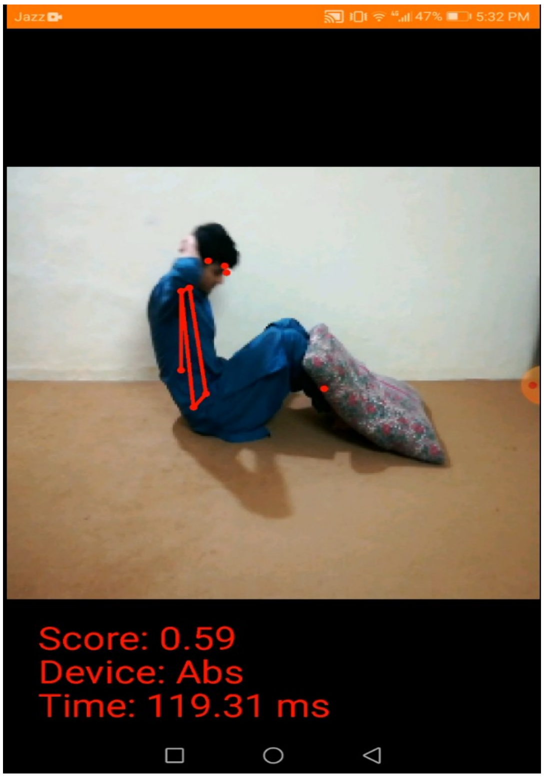

activity: Abs
0

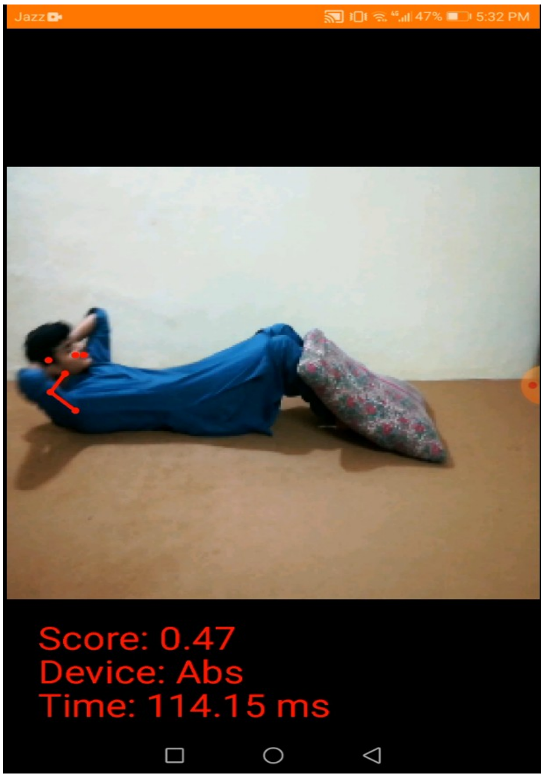

activity: Abs
0

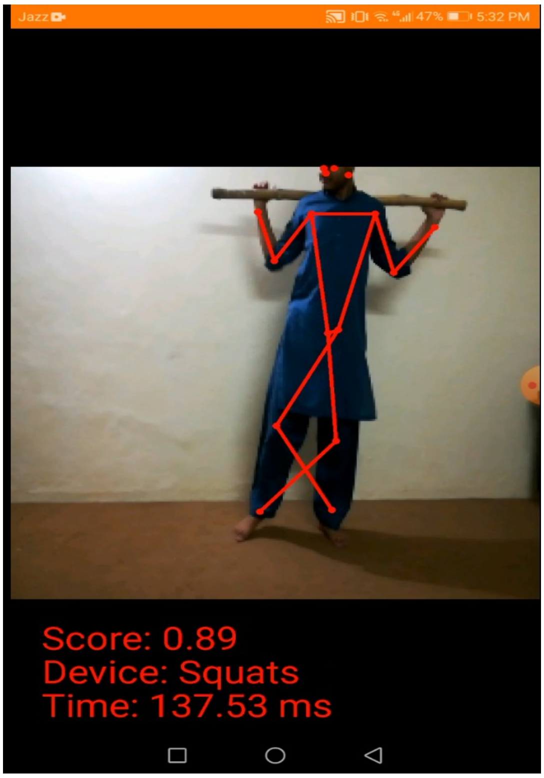
on real environment

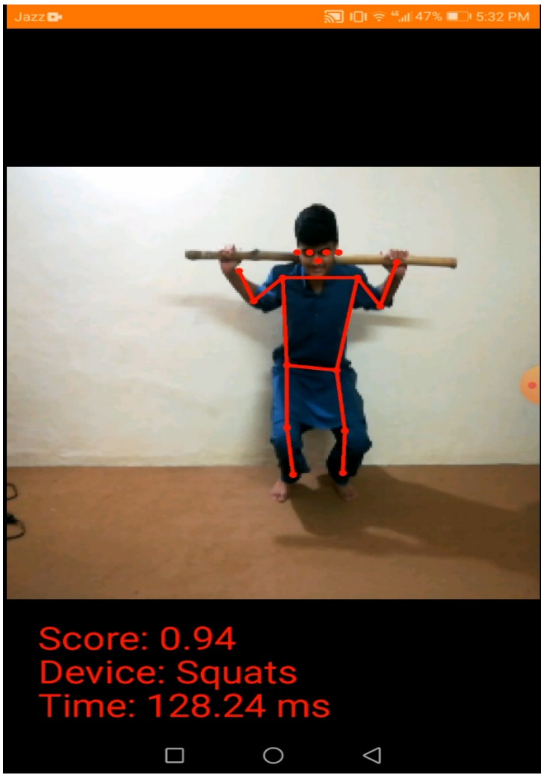


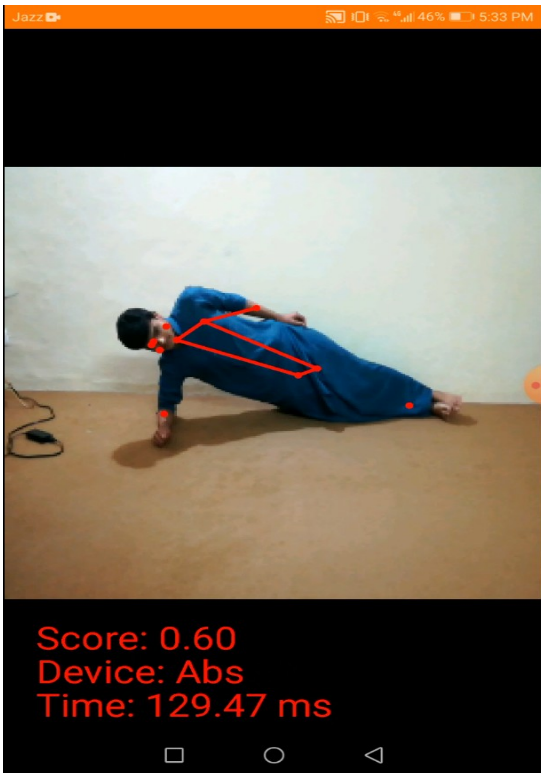

Jazz
47%
5:31 PM
Score: 0.65
Device: Push ups
Time: 132.60 ms


Jazz
47%
5:32 PM
Score: 0.59
Device: Abs
Time: 119.31 ms


Jazz
47% 5:32 PM
Score: 0.47
Device: Abs
Time: 114.15 ms


Jazz
47% 5:32 PM
Score: 0.89
Device: Squats
Time: 137.53 ms


Jazz
47% 5:32 PM
Score: 0.94
Device: Squats
Time: 128.24 ms

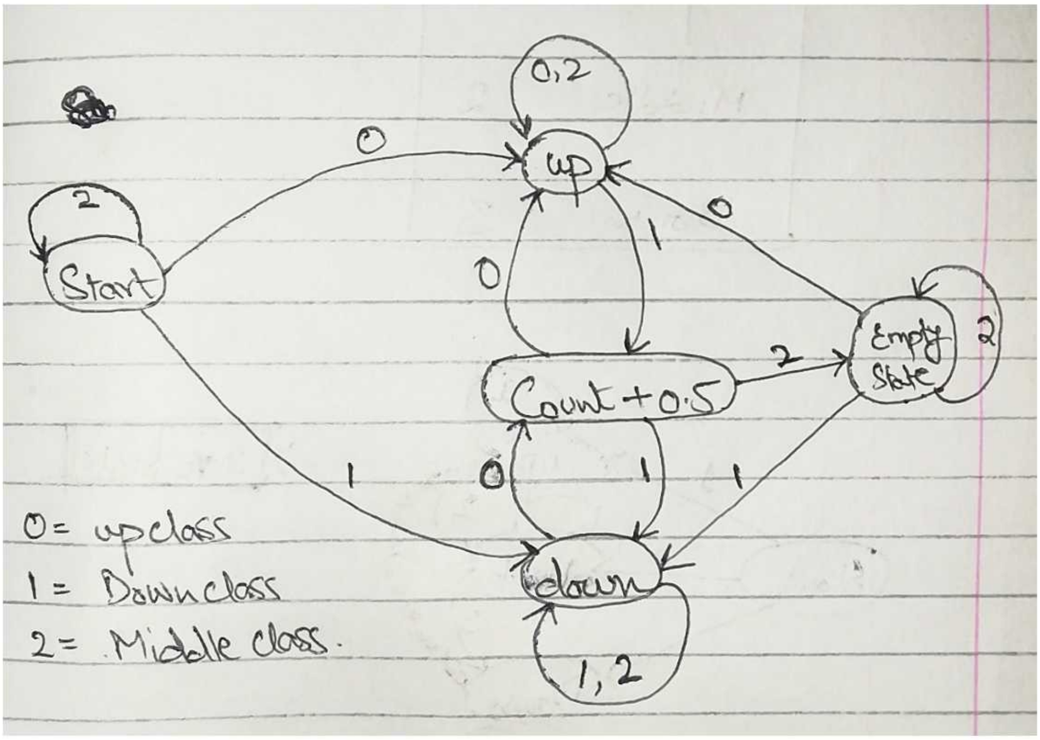

Jazz
46% 5:33 PM
Score: 0.60
Device: Abs
Time: 129.47 ms

5.5 Repetition Count from Image Classification Models

There are different techniques used for identifying repetitive sequence of frames in a video, one of them is by using complex machine learning storage algorithms like RNN, RCNN etc. but then our model will become computationally more complex and so its needs strong machine to handle them, if we still manages to implement these algorithms in mobile then there are large chances that it will further slow down the video per frame processing speed and induces large delays between two consecutive frames.

5.5.1 Implement DFA state machine for repetition count

So we implement a deterministic finite state machine its simple and have negligible computation as compare to RNN, RCNN.



Basically when system first detects any up class or any down class it stores that state, since there is a consecutive frames in video so there are high chances that the next frame will also be same state so its remains in that state. But when all the frames of that class are over then the next class which in our case is middle class will be detected, because during bench press no person can directly move from up position to down without passing middle class and wise versa, but as our repetition will count half when the person moves from up class to down class passing middle class or wise versa. Since next down or up class still haven't come so we’ll sill remain in that state, but as soon as we detect first frame of next class down class (when starting from up class) we move to count + 0.5 as now our half of the cycle is completed, so the process will keep on going unless the whole exercise is completed and the end we will get the total score of repetition of bench press exercise. This approach is implement on one exercise one but we have observed that it can be implemented on all the exercises.

5.6 Using Pose Estimation technique advantages/Limitations

5.6.1 Advantages

This technique can also help us in human pose correction during exercise by comparing it with trainer pose.

5.6.2 Limitations

This technique rely on the skeleton of user, which is unreliable approach when

1. user is far from the camera.
2. or if there is a occlusion comes in between camera and the user.

So this technique mainly depends upon the accuracy of pose estimation model and the precision of the camera.

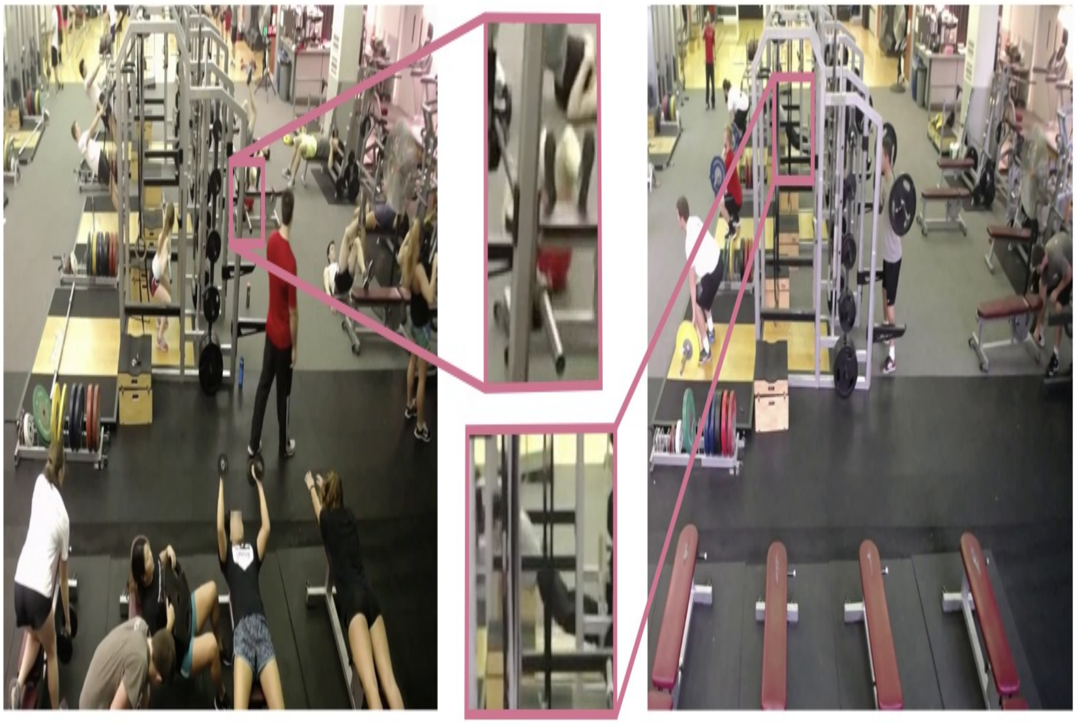


Fig 35: In gym settings, user pose can be challenging to determine due to significant occlusion[22]

Chapter 6 System Analysis

The system is designed using Three-Tier Architecture where there are three modules, such as, Mobile Application Module, PoseNet module, Sequential Module. The purpose of using Three-Tier Architecture was to bring modularity in the application to make the maintenance independent and easy.

6.1 System Three-Tier Architecture

The three layers of the system architecture are independent and follows the MVC (Model View Controller) in tier architecture as well as inside the tier layers separately. Following is the brief discussion on how the 3 tiers have been built and that upon what technologies.

6.1.1 Mobile Application Module

The Three-Tier Architecture, mobile application module (Frontend) of the system has been built on Android technology. The frontend also follows MVC Framework in which Each one of these section is worked to deal with particular development parts of an application. Here, the model is known as the storage of application related data. The view is the visual portrayal of the model. It speaks to the UI segments and is in charge of representation of information and the controller gets a demand from view, process the questions with the assistance of model and sends the outcomes back to the view.

6.1.2 PoseNet Module

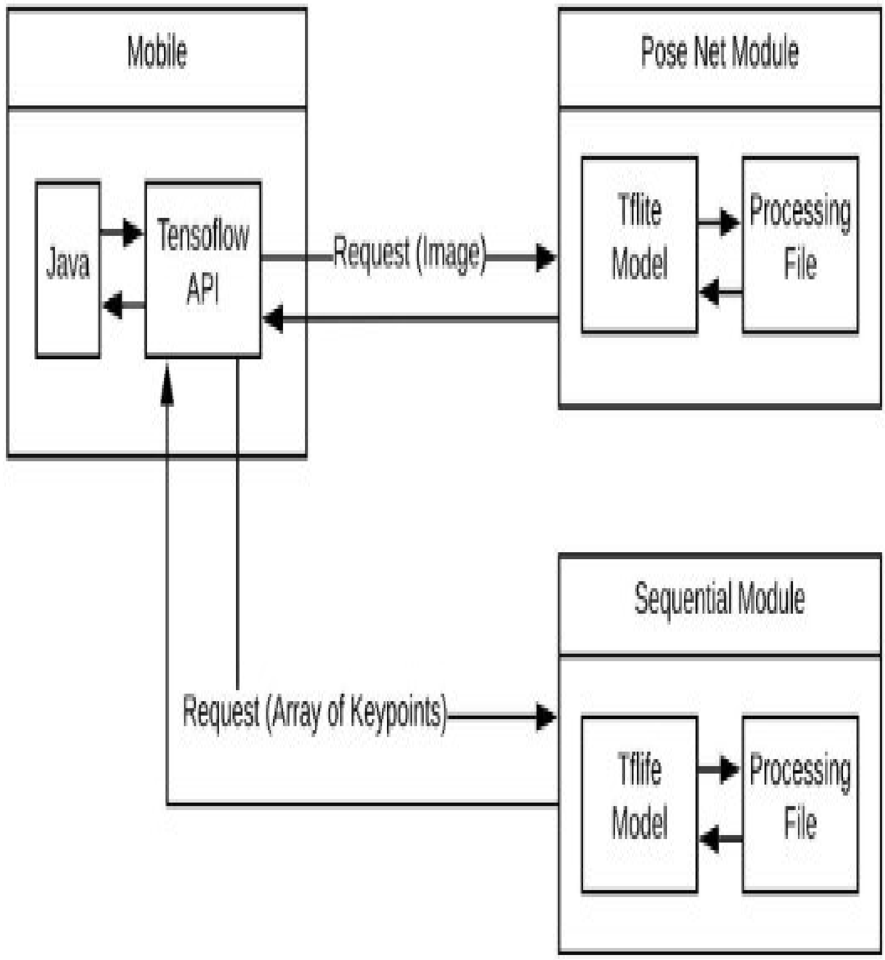
The Three-Tier Architecture, PoseNet Module has been build on tensorflow python framework. This module also uses MVC Framework because of which, this layer has modularity in its logic. But the as for now there is no view class as we do not want to display separately this module. This module is responsiable to find different body joints if the person is detected.

6.1.3 Sequential module

The Three-Tier Architecture, Sequential Module has been build on tensorflow python framework using MVC. This module is responsible to exercise detection. The module use 1D CNN to perform the task and give feedback to the application so that it can show results to the user.

6.1.4 System Architectural Diagram

The Android mobile application use java language where as tensor flow model



in .tflie is written in c language so there is a mediator tensor flow interpreter api that act as a mediator and make two compatible.

Chapter 7 Android Application

A mobile application or an app is a software application that runs on a mobile devices such as smart phones, tablets etc. Due to the excess of these devices these applications are very easily reachable for public.

7.1 Implementation of Mobile Module

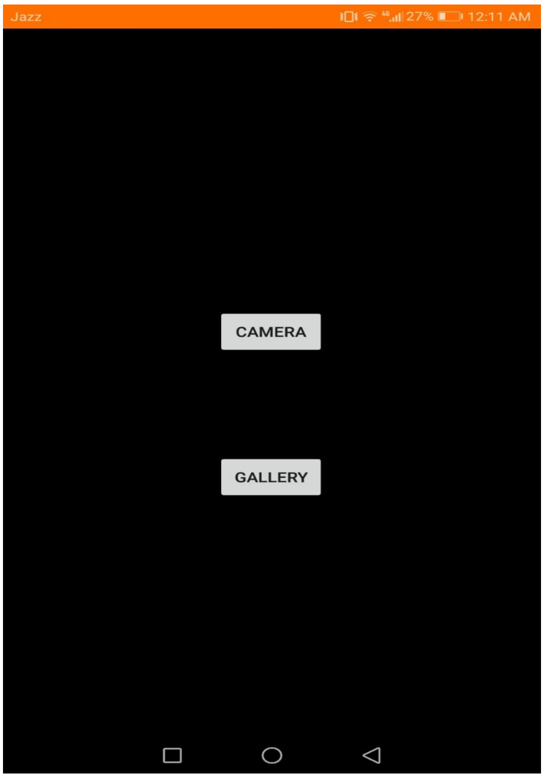
7.1.1 Language

Currently Android mobile application can be done using two languages one is kotlin and other is Java. I preferred to choose kotlin language because it combines both object-oriented and functional programming features and it uses LLMV compiler technology to compile Kotlin codes into binaries for CPU architectures and operating systems which makes its faster then java application to compile.

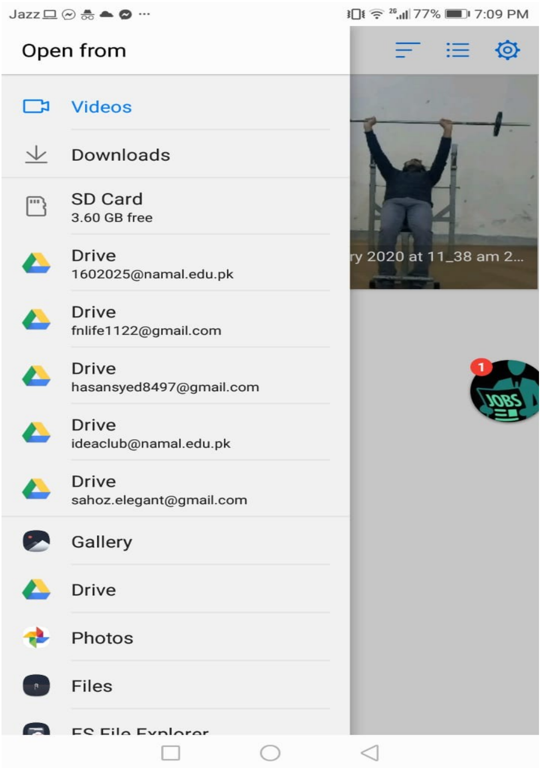
7.1.2 Activities, Fragments and Xmls

There are only 1 activity and 3 fragments and majority of the code is done in fragments so that the application can be scaled in the future

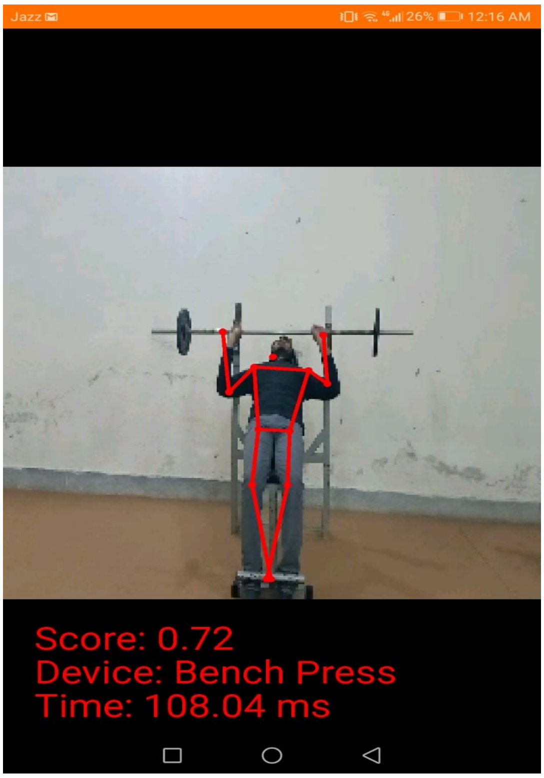
1. **Activity:**
    1. **CameraActivity:** This is the main activity from where the whole application starts. Its only initiate the start fragment.
2. **Fragments:**
    1. **StartPage:** This is the application home page where there are two buttons one for analyzing exercise in real-time where as other.



2. **VideoPlayer:** When the user choose to analize previously recorded video and press Gallery button then this fragment is initialized, it uses opencv to read video frame by frame and then after processing display the results.

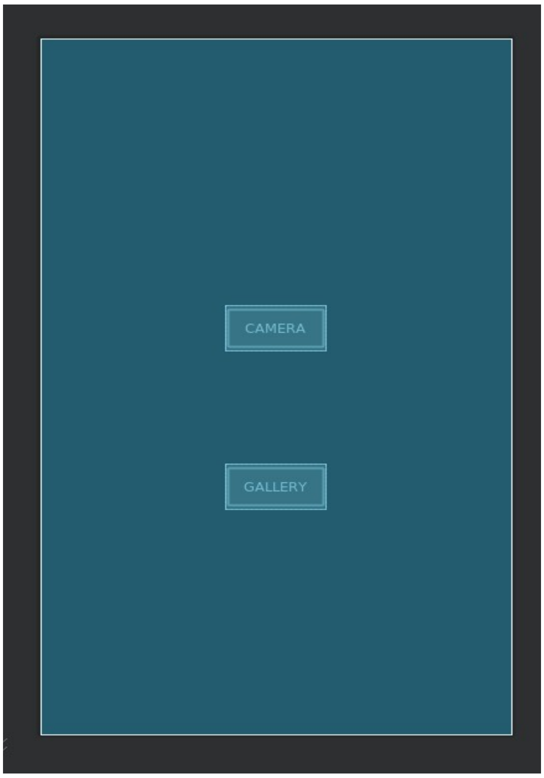


3. **PoseNetActivity:** When the user choose to analize real-time video and press Camera button then this fragment is initialized, it opens camera for 25ms and capture the image and process it. And display results with drowning body skeleton on image.

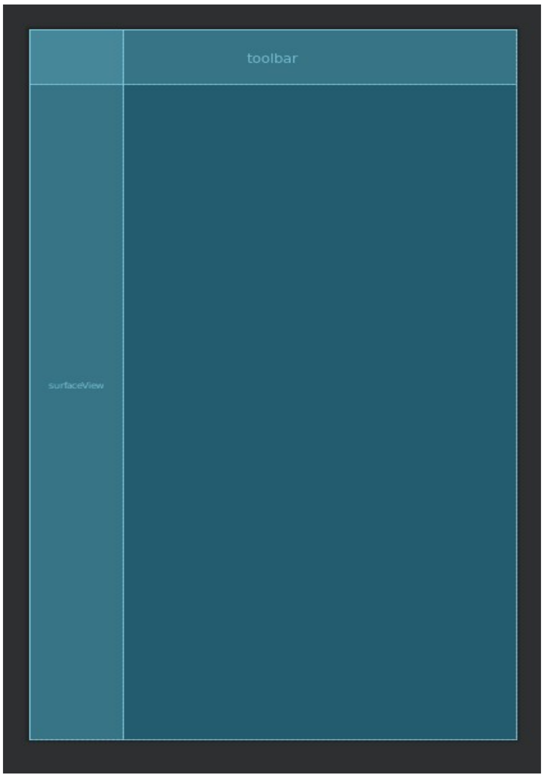


3. **XMLFiles**

1. **fragment_start_page:** This is the xml (View) file for the home page



2. **tfe_pn_activity_posenet:** This is the xml(view) for showing person performing exercise and drawing skeleton keyjoints on the image and displaying results in the bottom. Both camera and the gallery fragment use the view to display there results.



7.1.3 Other files:

There are three more files other then activity, fragment and xmls these are the model and control files named bellow

1. ImageUtils
2. Constants
3. Confirmation Dialog

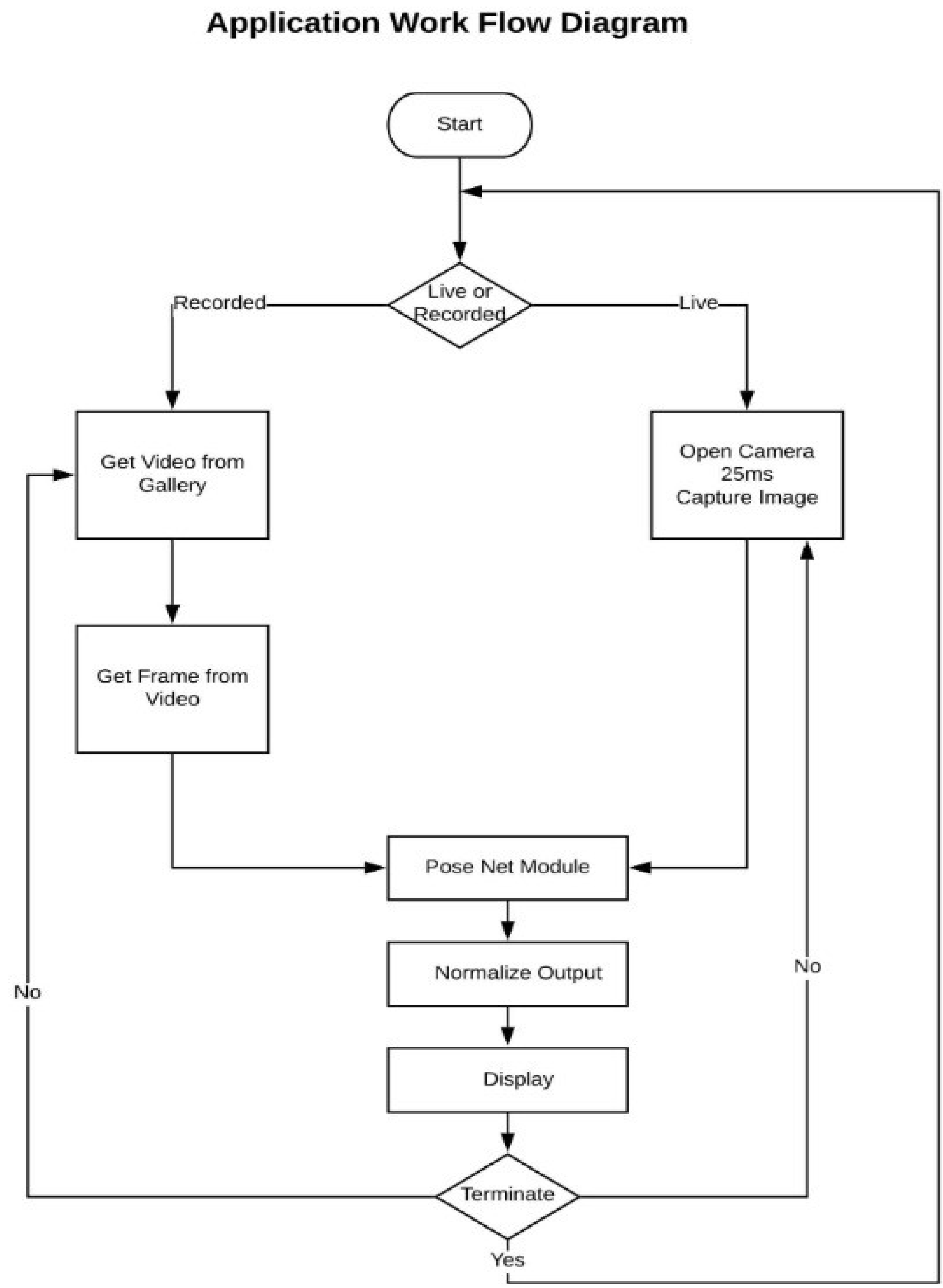
7.2 Implementation of PoseNet Module

This module also follow MVC modular approach there are two files in this module on is the .tflite model implementation and the other is the model file which stores pose, pose confidence score, keypoints, keypoints confidence score. there is no view file in it but support is available for the future.

7.3 Implementation of Sequential Module

This module also follow MVC modular approach similar to the above to modules, there are two files in this module on is the .tflite model file in which 1D CNN classification model is implemented and the other is the model file there is no view file in it but support is available for the future.

7.4 Application Work Flow Diagram



Chapter 8 Conclusion

8.1 Problems

The data set used was not good for the posenet for identification of body key joints as it was taken from a large distance. And due to COVID-19, I was unable to recreate the whole data set. I tried other pose estimation model (Open-pose having18 body key joints) for better results on my existing data set but these model need fast processing power which an ordinary mobile phone does not have so it stat crashes when I tried to run it on Android.

8.2 Future Extension

The idea can be extended for the Gym owners, they can use the CCTV surveillance camera data to train the model for the tracking all the gym-goers separately and log there respective workouts exercises in the gym app. The analyzed results can be then used for the nutrition for the proper creation of the diet plan for the person.

Exercise would be detected using the pose estimation on 17 different points so we can also use it to find whether the person performs the exercise correctly or not.

8.3 Summary

Fitness App is ansupervised learning machine learning project where we use twodifferent approaches to achieve our objectives. First one was to use1)simple CNN Image classification sequential model and some datastructure and algorithmic techniques for the exercise identificationand the repetition count, as for the proof of the concept we areconsidering 5 different exercises. Another approach was to 2)buildsequential model over the pre-build tensor flow posenet poseestimation model.

Both of the approachwere able to count repetition and identify exercises but the firstmodel was environment constrained but the second approach inenvironment independent but there was pose estimation model accuracy,limitation in the second approach.

Chapter 9 References:

[1] Sheri R Colberg, Ronald J Sigal, Jane E Yardley, Michael C Riddell, David W Dunstan, Paddy C Dempsey, Edward S Horton, Kristin Castorino, and Deborah F Tate. 2016. Physical activity/exercise and diabetes: a position statement of the American Diabetes Association. Diabetes Care 39, 11 (2016), 2065–2079.

[2]Christian Seeger, Alejandro Buchmann, and Kristof Van Laerhoven. 2011. Adaptive gym exercise counting for myHealthAssistant: poster abstract. In Proceedings of the 6th International Conference on Body Area Networks (BodyNets ’11). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Brussels, BEL, 126–127.

[3]Janssen, I., LeBlanc, A.G. Systematic review of the health benefits of physical activity and fitness in school-aged children and youth. Int J Behav Nutr Phys Act 7, 40 (2010). https://doi.org/10.1186/1479-5868-7-40

[4]Martyn Standage, Joan L Duda, and Nikos Ntoumanis. 2003. A model of contextual motivation in physical education: Using constructs from self-determination and achievement goal theories to predict physical activity intentions. Journal of educational psychology 95, 1(2003), 97.

[5]Booth, Frank & Roberts, Christian & Laye, Matthew. (2012). Lack of Exercise Is a Major Cause of Chronic Diseases. Comprehensive Physiology. 2. 1143-1211. 10.1002/cphy.c110025.

[6]Bravata, Dena & Smith-Spangler, Crystal & Sundaram, Vandana & Gienger, Allison & Lin, Nancy & Lewis, Robyn & Stave, Christopher & Olkin, Ingram & Sirard, John. (2007). Using Pedometers to Increase Physical Activity and Improve Health: A Systematic Review. JAMA : the journal of the American Medical Association. 298. 2296-304. 10.1001/jama.298.19.2296.

[7]Warburton D, Nicol CW, Bredin S: Health benefits of physical activity: The evidence. Canadian Medical Association Journal. 2006, 174: 801-809. 10.1503/cmaj.051351.

[8]Ryan RM, Williams GC, Patrick H, Deci EL. Self-determination theory and physical activity: The dynamics of motivation in development and wellness. Hellenic Journal of Psychology. 2009;6:107–124.

[9]Teixeira PJ, Carraça EV, Markland D, Silva MN, Ryan RM. Exercise, physical activity, and self-determination theory: a systematic review. Int J Behav Nutr Phys Act. 2012;9:78. Published 2012 Jun 22. doi:10.1186/1479-5868-9-78

[10]Rosemery O Nelson and Steven C Hayes. 1981. Theoretical explanations for reactivity in self-monitoring. Behavior Modification 5, 1

(1981), 3–14.

[11]Christian Seeger, Alejandro Buchmann, and Kristof Van Laerhoven. 2011. Adaptive gym exercise counting for myHealthAssistant. In

Proceedings of the 6th International Conference on Body Area Networks. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 126–127.

[12]Christian Seeger, Alejandro Buchmann, and Kristof Van Laerhoven. 2011. myHealthAssistant: a phone-based body sensor network

that captures the wearer’s exercises throughout the day. In Proceedings of the 6th International Conference on Body Area Networks. ICST

(Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 1–7.

[13]Francois Chollet. 2017. Deep Learning with Python (1st. ed.). Manning Publications Co., USA.

[14]Prof. Debahuti Mishra Dr Valentina Emilia Balas Dr. Pradeep Kumar Mallick, Dr. Sandeep Kumar Satapathy and Dr. Shruti Mishra. Machine learning for intelligent decision making in biomedical data. Informatics in Medicine Unlocked.

[15]Portions of this paper draw on Darrell M. West, The Future of Work: Robots, AI, and Automation, Brookings Institution Press, 2018.

[16]A. Singh, N. Thakur and A. Sharma, "A review of supervised machine learning algorithms," 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, 2016, pp. 1310-1315.

[17]F. Musumeci, C. Rottondi, A. Nag, I. Macaluso, D. Zibar, M. Ruffini, and M. Tornatore. An overview on application of machine learning techniques in optical networks. IEEE Communications Surveys Tutorials, 21(2):1383– 1408, Secondquarter 2019.

[18]Md Rezaul Karim, Oya Beyan, Achille Zappa, Ivan G Costa, Dietrich Rebholz-Schuhmann, Michael Cochez, Stefan Decker, Deep learning-based clustering approaches for bioinformatics, Briefings in Bioinformatics, , bbz170, https://doi.org/10.1093/bib/bbz170

[19]Diederik P. Kingma, Danilo J. Rezende, Shakir Mohamed, and Max Welling. 2014. Semi-supervised learning with deep generative models. In Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2 (NIPS’14). MIT Press, Cambridge, MA, USA, 3581–3589. [20]https://blog.seebo.com/supervised-vs-unsupervised-machine-learning/

[21]http://www.nbcnews.com/id/15598063/ns/health-fitness/t/another-hurdle-exercise-embarrassment/#.XvhO9nUzY5k

[22]Rushil Khurana, Karan Ahuja, Zac Yu, Jennifer Mankoff, Chris Harrison, and Mayank Goel. 2018. GymCam: Detecting, Recognizing and Tracking Simultaneous Exercises in Unconstrained Scenes. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 2, 4, Article 185 (December 2018), 17 pages. https://doi.org/10.1145/3287063

[23]Kuo, Ping-Huan & Huang, Chiou-Jye. (2018). A Green Energy Application in Energy Management Systems by an Artificial Intelligence-Based Solar Radiation Forecasting Model. Energies. 11. 819. 10.3390/en11040819.

[24]https://medium.com/tensorflow/real-time-human-pose-estimation-in-the-browser-with-tensorflow-js-7dd0bc881cd5#:~:text=PoseNet%20can%20be%20used%20to,persons%20in%20an%20image%2Fvideo.

[25]Christian Seeger, Alejandro Buchmann, and Kristof Van Laerhoven. 2011. MyHealthAssistant: a phone-based body sensor network that captures the wearer’s exercises throughout the day. In Proceedings of the 6th International Conference on

Body Area Networks (BodyNets ’11). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Brussels, BEL, 1–7.

[26]Šeketa G., Džaja D., Žulj S., Celić L., Lacković I., Magjarević R. (2015) Real-Time Evaluation of Repetitive Physical Exercise Using Orientation Estimation from Inertial and Magnetic Sensors. In: Jobbágy Á. (eds) First European Biomedical Engineering Conference for Young Investigators. IFMBE Proceedings, vol 50. Springer, Singapore

[27]Chan CB, Ryan DA, Tudor-Locke C. Health benefits of a pedometer-based physical activity intervention in sedentary workers. Preventive Medicine. 2004 Dec;39(6):1215-1222. DOI: 10.1016/j.ypmed.2004.04.053.

[28]C. Crema, A. Depari, A. Flammini, E. Sisinni, T. Haslwanter and S. Salzmann, "IMU-based solution for automatic detection and classification of exercises in the fitness scenario," 2017 IEEE Sensors Applications Symposium (SAS), Glassboro, NJ, 2017, pp. 1-6, doi: 10.1109/SAS.2017.7894068.

[29]Dan Morris, T. Scott Saponas, Andrew Guillory, and Ilya Kelner. 2014. RecoFit: using a wearable sensor to find, recognize, and count repetitive exercises. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI ’14). Association for Computing Machinery, New York, NY, USA, 3225–3234. DOI:https://doi.org/10.1145/2556288.2557116

[30]Keng-Hao Chang, Mike Y. Chen, and John Canny. 2007. Tracking free-weight exercises. In Proceedings of the 9th international conference on Ubiquitous computing (UbiComp ’07). Springer-Verlag, Berlin, Heidelberg, 19–37.

[31]Michael Muehlbauer, Gernot Bahle, and Paul Lukowicz. 2011. What Can an Arm Holster Worn Smart Phone Do for Activity Recognition? In Proceedings of the 2011 15th Annual International Symposium on Wearable Computers (ISWC ’11). IEEE Computer Society, USA, 79–82.

[32]Eduardo Velloso, Andreas Bulling, and Hans Gellersen. 2013. MotionMA: motion modelling and analysis by demonstration. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI ’13). Association for Computing Machinery, New York, NY, USA, 1309–1318. DOI:https://doi.org/10.1145/2470654.2466171

[33]Eduardo Velloso, Andreas Bulling, Hans Gellersen, Wallace Ugulino, and Hugo Fuks. 2013. Qualitative activity recognition of weight lifting exercises. In Proceedings of the 4th Augmented Human International Conference (AH ’13). Association for Computing Machinery, New York, NY, USA, 116–123. DOI:https://doi.org/10.1145/2459236.2459256

[34]Han Ding, Longfei Shangguan, Zheng Yang, Jinsong Han, Zimu Zhou, Panlong Yang, Wei Xi, and Jizhong Zhao. 2015. Femo: A platform for free-weight exercise monitoring with rfids. In Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems. ACM, 141–154.

[35]Abdelkareem Bedri, Richard Li, Malcolm Haynes, Raj Prateek Kosaraju, Ishaan Grover, Temiloluwa Prioleau, Min Yan Beh, Mayank Goel, Thad Starner, and Gregory Abowd. 2017. EarBit: Using Wearable Sensors to Detect Eating Episodes in Unconstrained Environments. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 1, 3 (2017), 37.

[36]Ilktan Ar and Yusuf Sinan Akgul. 2014. A computerized recognition system for the home-based physiotherapy exercises using an RGBD

camera. IEEE Transactions on Neural Systems and Rehabilitation Engineering 22, 6 (2014), 1160–1171.

[37]Antón, David & Goni, Alfredo & Illarramendi, Arantza. (2015). Exercise Recognition for Kinect-based Telerehabilitation. Methods of Information in Medicine. 54. 145-155. 10.3414/ME13-01-0109.

[38]Anderson, Fraser & Grossman, Tovi & Matejka, Justin & Fitzmaurice, George. (2013). YouMove: Enhancing movement training with an augmented reality mirror. UIST 2013 - Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology. 311-320. 10.1145/2501988.2502045.

[39]Nielsen, M. A. (2018). Neural Networks and Deep Learning [misc]. Determination Press