# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI – 590 018



**A project Phase I Report on**

# Physical Exercise Form Detection and Correction

Submitted in partial fulfillment of the requirements for the VII Semester of degree of **Bachelor of Engineering in Information Science and Engineering** of Visvesvaraya Technological University, Belagavi

##### Submitted By

**Nikita Pyati – 1RN19IS095 Rohan Savalgi – 1RN19IS121**

**Sachin Rao – 1RN19IS127 Vineet KS – 1RN19IS175**

**Under the Guidance of**

**Mr. Santosh Kumar**

##### Associate Professor Department of ISE



**Department of Information Science and Engineering**

# RNS Institute of Technology

## Dr. Vishnuvaradhan Road, Rajarajeshwari Nagar post, Channasandra, Bengaluru-560098

**2021-2022**

# RNS INSTITUTE OF TECHNOLOGY

## Dr. Vishnuvaradhan Road, Rajarajeshwari Nagar post, Channasandra, Bengaluru - 560098

**DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING**



# CERTIFICATE

Certified that the project work phase 1 entitled **Prevention of Wild Fire by Early Fire and Smoke Detection** has been successfully completed by **Nikita Pyati (1RN19IS095), Rohan Savalgi (1RN19IS121), Sachin K Rao (1RN19IS127) and Vineet K S (1RN19IS175)** bona fide students of **RNS Institute of Technology, Bengaluru** in partial fulfillment of the requirements for the award of degree in **Bachelor of Engineering in Information Science and Engineering** of **Visvesvaraya Technological University, Belagavi** during academic year **2022-2023**. The project phase 1 report has been approved as it satisfies the academic requirements in respect of project phase 1 work for the said degree.



|  |  |  |
| --- | --- | --- |
| **\_ \_\_\_** | **\_ \_ \_** | **\_ \_ \_** |
| **Mrs. Sunitha K** | **Dr. Suresh L** | **Dr. M K Venkatesha** |
| Project Guide | Professor and HoD | Principal |
| Assistant Professor Department of ISE | Department of ISE RNSIT | RNSIT |
| **Name of the Examiners** | **External Viva** | **Signature with Date** |
| **1. \_ \_** |  | **1. \_ \_** |
| **2**. **\_ \_** |  | **2. \_** |

# DECLARATION

##### We, NIKITA PYATI [1RN19IS095], ROHAN SAVALGI [1RN19IS121], SACHIN K RAO [1RN19IS127] and VINEET K S [1RN19IS175], students of VII Semester BE, in Information Science and Engineering, RNS Institute of Technology hereby declare that the Project work phase 1 entitled *PHYSICAL EXERCISE FORM CORRECTION USING VIDEO PROCESSING* has been carried out by us and submitted in partial fulfillment of the requirements for the *VII Semester degree of Bachelor of Engineering in Information Science and Engineering of Visvesvaraya Technological University, Belgaum* during academic year 2022-2023.

**Place:** Bengaluru **Nikita Pyati – 1RN19IS095**

##### Date:

**Rohan Savalgi – 1RN19IS121**

**Sachin Rao – 1RN19IS127**

**Vineet KS – 1RN19IS175**

# ABSTRACT

Physical exercise is an important part of the daily routine that helps to overcome the risk of diseases as well as improve quality of life. Physical exercise is essential to prevent and reduce the risk of many diseases and improve physical and mental health. People are quite busy in modern times and this trend is increasing day by day, therefore, this is a time to think about the appropriate technology for monitoring their activities and caring for their elderly. An intelligent interface between the human body and the monitoring mechanism could solve this type of problem

In recent years, many studies have proposed various methodologies to extract various physiological data and monitor physical exercise. Mainly, two approaches (contact sensor technology and contactless technology) are widely used in recent times to extract physiological features whether during physical exercise or resting. Both the approaches have pros/cons, however, the proposed implementation is parallelly used. In contact sensor techniques, the sensors are attached on the body and the result is interpreted using machine learning or statistical approaches. For non-contact/contact sensor techniques, various types of cameras are used to capture video/images and processed with various image processing techniques including Infrared camera and thermal camera. Whatever the image extraction technique, they must be followed by computer vision or machine learning and deep learning techniques to interpret the result.

# ACKNOWLEDGMENT

The fulfillment and rapture that go with the fruitful finishing of any assignment would be inadequate without the specifying the people who made it conceivable, whose steady direction and support delegated the endeavors with success.

We would like to profoundly thank **Management** of **RNS Institute of Technology** for providing such a healthy environment to carry out this project work.

We would like to express our thanks to our Principal **Dr. M K Venkatesha** for his support and inspired us towards the attainment of knowledge.

We wish to place on record my words of gratitude to **Dr. Suresh L,** Professor and Head of the Department, Information Science and Engineering, for being the enzyme and master mind behind my project work.

We would also like to thank our project coordinator **Mrs. Kusuma S,** Assistant Professor and **Dr. Prakasha S,** Associate Professor our project guide **Mr. Santosh Kumar,** Assistant Professor, Department of ISE, RNSIT, Bangalore, for their valuable suggestions and guidance.

We would like to thank all other teaching and non-teaching staff of Information Science & Engineering, RNSIT, Bangalore, who have directly or indirectly helped us to carry out the project work.

**Place:** Bengaluru

##### Date:

**ANIL [1RN18IS018] DARSHAN S K [1RN18IS036] DEEPAK C [1RN18IS037] GANGADHAR ADI [1RN18IS047]**

**TABLE OF CONTENTS**

**Declaration I**

[**Abstract II**](#_heading=h.gjdgxs)

[**Acknowledgment III**](#_heading=h.30j0zll)

[**Table of Contents IV**](#_heading=h.1fob9te)

**List of Figures V**

[**List of Abbreviations VI**](#_heading=h.3znysh7)

1. **INTRODUCTION 1**
2. **LITERATURE REVIEW 3**
3. **PROBLEM IDENTIFICATION 11**
4. **OBJECTIVES 12**
5. **METHODOLOGY 13**
6. **SYSTEM DESIGN 18**
7. **CONSLUSION 36**
8. **REFERENCES 37**

|  |  |  |
| --- | --- | --- |
|  | **LIST OF FIGURES** |  |
| **Fig No.** | **Description** | **Page No.** |
| Figure. 3.1 | Forest Fire and Smoke | 11 |
| Figure 5.1 | Structural chart of the proposed system. | 13 |
| Figure 5.2 | Typical CNN Architecture | 15 |
| Figure 5.3 | Layers in CNN | 15 |
| Figure 5.4 | Convolutional Layer | 16 |
| Figure 5.5 | Pooling Layer | 17 |
| Figure 5.6 | fully connected layer and Output Layer | 17 |
| Figure 6.1 | Project Modules | 19 |
| Figure 6.2 | System Architecture of the smoke and Fire detection | 22 |
| Figure 6.3 | Use case diagram of the proposed system | 23 |
| Figure 6.4 | Data flow diagram for pre processing | 25 |
| Figure 6.5 | Data flow diagram for Identification | 25 |
| Figure 6.6 | Data flow diagram for Feature Extraction | 25 |
| Figure 6.7 | Data flow diagram for Classification and Detection | 26 |
| Figure 6.8 | Activity diagram for smoke and fire detection | 27 |
| Figure 6.9 | Sequence diagram for smoke and fire detection | 27 |
| Figure 6.10 | Flowchart for the smoke and fire detection using deep learning | 28 |
| Figure 6.11 | Flowchart for data acquisition | 28 |
| Figure 6.12 | Flowchart for the preprocessing module | 29 |
| Figure 6.13 | Conversion from RGB to grayscale | 30 |
| Figure 6.14 | Noise filtering using Median Filter | 31 |
| Figure 6.15 | Thresholding using Basic global Thresholding | 31 |
| Figure 6.16 | Image Sharpening using High-Pass Filter | 32 |
| Figure 6.17 | Flowchart for Feature Extraction | 32 |
| Figure 6.18 | Gx and Gy in HOG | 33 |
| Figure 6.19 | Feature Extraction using HOG | 34 |
| Figure 6.20 | Flowchart for classification using CNN | 34 |

# List of Abbreviations

|  |  |
| --- | --- |
| CNN | Convolution Neural Network |
| FTTC | Fault-Tolerant Cooperative Control |
| HLD | High-level design |
| HOG | Histogram Orientation Gradient |
| LoRa | Long Range Wireless Data Telemetry |
| ML | Machine Learning |
| RGB | Red Green Blue |
| SMC | Sliding Model Control |
| UAV | Unmanned Aerial Vehicles |
| VGG | Visual Geometry Group |
| WSN | Wireless Sensor Network |

**CHAPTER 1**

# INTRODUCTION

## 1.1 Overview

Fitness is important in people’s lives. Good fitness habits can improve cardiopulmonary capacity, increase concentration, prevent obesity, and effectively reduce the risk of death. Home fitness does not require large equipment but uses dumbbells, yoga mats, and horizontal bars to complete fitness exercises and can effectively avoid contact with people, so it is deeply loved by people. People who work out at home use social media to obtain fitness knowledge, but learning ability is limited. Incomplete fitness is likely to lead to injury, and a cheap, timely, and accurate fitness detection system can reduce the risk of fitness injuries and can effectively improve people’s fitness awareness.

. Exercise can be helpful to personal health. However, it can also be potentially dangerous and inefficient if the exercise is performed in wrong posture by the trainees. Performing wrong posture is a very common issue for every gymnast, either beginner or even professional. This is because it is a big challenge for trainees to perform a series of complex movements in exercise. Posture issues can manifest into many different conditions that can put trainee health in danger, such as neck pain, back pain, shoulder pain or knee pain. Correct posture is able to minimize the strain on the human body by maintaining the balance between muscles and skeleton. This balanced musculoskeletal state is very important because it prevents further damage or any progressive deformation in exercise training and protects the supporting structures in the body. Therefore, performing correct posture is a very crucial aspect for proper arrangement of supporting structures, maintaining balance of the body and effective functioning of the body.

Doing exercises the right way is not an easy task. If the user wants serious results he/she has to learn the correct way of training. The wrong posture or technique is not just ineffective, it can also lead to serious injuries. Having a personal trainer reduces these risks, but it is expensive and gyms can become overcrowded.

In the past, many studies have engaged in the detection of fitness movements, among which the detection of fitness movements based on wearable devices, body nodes, and image deep learning has achieved better performance.

.

Project Phase 1

However, a wearable device cannot detect a variety of fitness movements, may hinder the exercise of the fitness user, and has a high cost. Both body-node-based and image-deep-learning-based methods have lower costs, but each has some drawbacks.

Physical exercise form detection using video processing is a technique that involves analyzing video footage of a person performing physical exercises in order to determine whether they are executing the movements correctly. This can be useful for a variety of purposes, such as providing feedback to athletes or fitness enthusiasts to help them improve their technique, or for use in fitness apps or other digital tools to provide guidance and motivation to users. By using computer vision techniques to analyze the video frame-by-frame and identify key points or landmarks on the person's body, the system can evaluate the exercise form and provide feedback to the user based on predefined rules or criteria. Physical exercise form detection using video processing can be a powerful tool for improving athletic performance and promoting healthy habits.

Dept. of ISE,RNSIT 2021-2022 2

**Chapter 2**

# LITERATURE SUVERY

##### Physical Exercise Form Correction Using Neural Networks(2020)

This paper presents a labeled dataset for training Artificial Intelligence models to help correct posture in static fitness exercises. It proposes a solution based on Convolutional Neural Networks and evaluates the performance under different environmental conditions (background, camera angle, distance etc.). The trained network is embedded in a mobile application. Our work reveals some challenges and solutions to overcome them. Apart from the improvements suggested by the previous section, further work could enhance the flexibility of the training process in order to make the solution more customizable by end-users and trainers. Instead of embedding the models inside the application, one can use a database and create a web service for users to upload their own images and train new models. Another challenge comes when considering rehabilitation patients that cannot perform an exercise correctly. Then the system would need to adapt through the recovery phases, at first being loose, and becoming more demanding in time, similar to how a human personal assistant would behave.

##### Posture Correction using Human Pose Estimation(2020)

OpenPose is a strong library used for human pose detection. A model with decent accuracy was developed that could detect the exercise being performed and give accurate suggestions based on the ML model with the help of geometry heuristics. Fig. 4.1 Angle between upper arm and forearm for “Good” posture Correct positioning involves training yourself to hold your body against gravity with the least strain and tension on supportive structures.In this project, an application is presented which provides feedback on human posture while performing exercises using pose detection, visual geometry, and machine learning. The output of this project is to locate the human body's key points from the video provided. ML algorithm is used for analyzing posture correctness and geometric algorithms for providing feedback on exercises performed.

##### Implementation of Machine Learning Technique for Identification of Yoga Poses

Researches have been done on yoga pose detection and correction. Some researchers have used a Kinect device to form a human posture. This device used to capture the images but the important part is that this device contains an inbuilt infrared laser projector, a multiarray microphone and an RBG camera used to capture the color and depth images. This device also has a tool that makes a human body skeleton in 3D space which gives the information about the coordinates of the joint of the body. This method is good but the main disadvantage of this method is that the Kinect device is expensive and not user-friendly.

To avoid this problem tf-pose Algorithm has been used. This Algorithm creates a skeleton of a human body and gives the desired information about the joint in the human body. Using this one can find the coordinates of the joints and use that as a feature to detect the posture of a body. Paula Pullen, William Seffens used visual Gesture Builder feature of kinect sensor which used to capture yoga postures with high accuracy.

**Human Pose Estimation in the Field of Sport and Physical Exercise**

Based Human Pose Estimation (HPE) consists of estimating the position of different parts of the body, such as the joints in a 2D or 3D space depending on the estimation type, normally from visual information, such as images, and sometimes through other additional data obtained by different types of sensors, such as inertial sensors or depth sensors. This field of research can be considered a combination of Data Processing and Artificial Intelligence, more specifically, Computer Vision. Since 2014, and mainly the past 5 years, the use and interest in HPE has increased, mainly due to the introduction of Deep Learning to the field [1]. The methodology has evolved from the first simple neural networks to the complex Convolutional Neural Networks (CNN) of today. The use of filters to obtain lines, edges, silhouettes, and other remarkable characteristics of the elements contained in images, as well as the capability of providing information to a system that can learn some characteristics and then detect them when a similar situation is given, have supposed an inflection point.

##### Assessing Human Motion During Exercise Using Machine Learning

Rule-based approaches use predefined conditional statements to evaluate the properties of human motion (e.g., joint angles, range of motion, and relations between joints). Template-based approaches use algorithms such as Dynamic Time Warping (DTW) and Hidden Markov Models (HMM), which perform pattern matching using previously recorded motions. These methods are easy to implement and run in real-time; thus, they are ideal for assessing motion during exercise. However, the number of exercise executions that can be used to create rules/templates is limited. This reduces the scalability of these approaches [29]. Consequently, it is difficult to develop personalized solutions that can match the unique characteristics or impairments of each individual. To overcome the above limitations, there has recently been increased interest in using ML for HMQA. The process for developing such solutions includes several steps. A high-level schematic representation of the overall process is shown in Fig. 1. However, the exact steps that each solution adapts, can vary based on the input data modalities, type of assessment, and algorithm used.

##### Pose Trainer: Correcting Exercise Posture Using Pose Estimation

An approach which named Pose Trainer has been introduced by Chen et al. (2018) for detecting and correcting human posture by using pose estimation. This approach consists of three parts that constructed by six components (Figure 2.1). The first part has only one component, Video Recording and it can only be done by user. The next part is built from Pose Estimation , Keypoint Normalization and Exercise Detection. A state-of-the-art pose estimation deep neural network, OpenPose from OpenCV is used within Pose Trainer for inference. The third part involves detecting the quality of user’s predicted pose for specific exercise, which are Posture Evaluation and Feedback Correction.

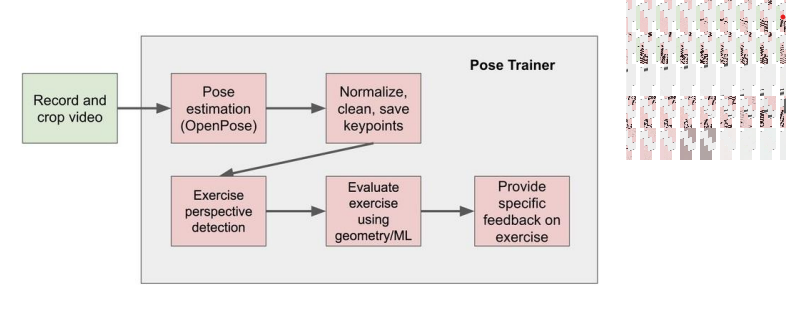


Figure 2.1: Flowchart of Pose Trainer

### Machine Learning Methods for the Automatic Evaluation of Exercises on Sensor-Equipped Weight Training Machines

On the other hand, Hristo and Arnold in 2012 introduced a pattern recognition techniques for the evaluation of exercises performed on weight training machines equipped with a load cell and rotary encoder. This approach is able to evaluate the exercise movement quality regarding predefind criteria such as duration, constancy, velocity and completeness. There are totally five steps which are Signal Input from sensor data, Filtering, Segmentation, Feature Extraction and Classification (Figure 2.2)

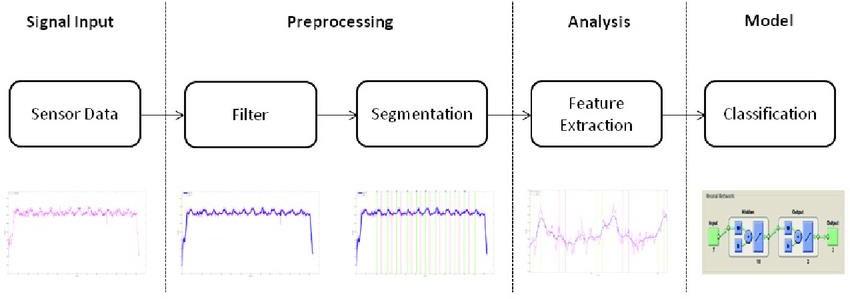


Figure 2.2: Proposed Method Flowchart

In the first step, the weight training machines are equipped with sensors such as load cell and rotary encoder for data collection purposes. There are two data that retrieved from sensor which are weight displacement value and force signal. Further determinants such as velocity, acceleration, power and repetition can be derived from sensor data.

For the data pre-processing step, it consists of cleaning, filtering and segmenting data. A strong low-pass filter is applied for smoothing the displacement value while an average low-pass filter is used for the force signal in order not to lose significant characteristics on occurred fluctuations.

The goal of feature extraction is to find and derive obvious characteristics that will identify the pre-processed data. First of all, an exhaustive set of attributes is extracted for each repetition in feature space. Then, the feature selection is based on time, velocity, constancy, and completeness properties in weight training.

In the last step, supervised learning method is chosen because the appraisements from professional coaches are available. These assessments can label the feature vector and then assigned an exhaustive set of features to specific classes such as well-performed posture class, improper posture class and inconsistent posture class. A traditional Artificial Neural Network (ANN) approach is applied on the labelled data and the modelling results shows 93% accuracy.

### Detecting Unseen Anomolies in Weight Training Exercises

An interesting approach had been presented by Kowsar et al. in 2016 to detect unseen anomalies in weight lifting training by using motion sensor and camera. This paper presented a workflow to detect performance anomalies from observations of correct performance of an exercise (Figure 2.3). The dataset used in this paper is unilateral bicep curl exercise.

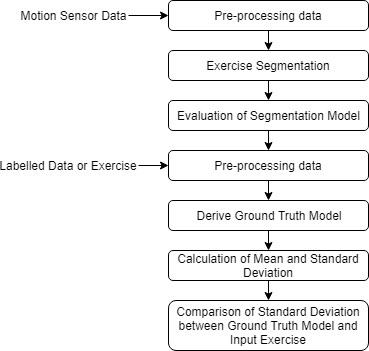


Figure 2.3: Proposed Method Flowchart

The first step of this approach is to pre-processing motion sensor data. It is common that motion sensor showed some degree of white noise that came from nature of the sensor and the white noise showed as small perturbations around the actual value. Any successful data analysis is start from removing noise from the data (Yun et al. 2006). Therefore, Kalman Filter had been chosen as a high-pass filter to eliminate noise with high accuracy.

Weight lifting exercise are repetitive tasks where same move were performed for a few repetitions. In this paper, the exercise is segmented to each every repetition by using motion sensor data. A function “*f*” is defined as change-of-position in space then chain rule is applied into this function. Minimum point indicating the start point of each move and it can be estimated from *f.* After finding out all minimum points on the acceleration-time graph, segmentation can be done based on those points. An evaluation is carried out to test the accuracy of *f* function and I has 96.5% of precision which is an acceptable result.

Standard deviation for ground truth is calculated for detecting anomalies in input weight lifting exercise. Each associated trajectory of segment of input exercise calculated. Then, mean and standard deviation for each point in time in the trajectory set are also calculated. Mean and standard deviation from input exercise are compared with mean and standard deviation from ground truth model. If any point is found outside the margin from the new trajectory, the segment is labelled as wrong posture . This approach gave a result with 99.4% True Positive

### Qualitative Activity Recognition of Weight Lifting Exercises

For the first approach, there are few steps taken to recognise and evaluate performed exercise (Figure 2.4). The main concept of this approach is to classify the exercise into different class, such as correct posture class or incorrect posture class. In this paper, 5 classes are defined while first class corresponds to correct performance of the workout while the remaining classes corresponds to other common mistakes made by trainee.



Figure 2.4: Flowchart of Machine Learning Approach

In the first step, video of executing the same workout correctly and with a set of common mistakes with some wearable sensors and machine learning is used to predict every mistake. In this way, training data is used as the activity specification and the classification algorithm as the means to compare the execution to the specification.

For feature extraction, an approach in sliding window with different lengths is used. Features are calculated on the Euler angles, as well as the gyroscope, accelerometer and readings of magnetometer for each step of sliding window. Eight features such as mean, variance, standard deviation and many more are calculated from Euler angles for each four sensors, generated in total 96 derived feature vectors. In order to identify the useful features, feature selection algorithm based on correlation was used. The algorithm was trained to use a “Best First” strategy based on backtracking.

For the final step, the exercise will be classified into different class based on extracted and selected features from previous step. In exercise recognition performance step, Random Forest approach is used due to the noise characteristics in the sensor data. Ten random forests were used and ten trees were implemented in each forest. The trained classifier is tested with 10 fold cross-validation and multiple windows sizes, with 0.5s overlapping. Lastly, the overall result in testing data was 78.2%

For second approach, there are also few steps taken to recognise and evaluate the performed exercise (Figure 2.11).

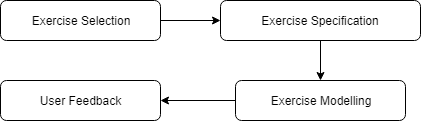


Figure 2.5: Flowchart of Exercise Specification Approach

In exercise selection, the repetitive exercise is segment into each repetition separately. This wat the analysis of each repetition can be done separately. For instance, a front raise exercise involves raising and lowering the dumbbell or barbell by using hands with multiple repetitions, therefore the beginning of the activity is defined as when the trainee started to lift it and the end as when it reached the initial position again.

##### Real-Time Detection of Exercise Form Using a Wearable Inertial Measurement Unit

Wearable inertial measurement units (IMUs) are small, lightweight devices that can be worn on the body to measure movement and orientation. They typically consist of a combination of sensors such as accelerometers, gyroscopes, and magnetometers, which can detect changes in acceleration, angular velocity, and magnetic field strength. Using a wearable IMU, it is possible to develop algorithms that can detect and classify different types of physical activity or exercise in real-time. This can be done by analyzing the data from the IMU sensors to identify patterns and characteristics associated with different types of exercise. For example, an IMU could be used to detect and classify exercises such as squats, lunges, or push-ups by analyzing the acceleration and angular velocity data from the sensors. There are several potential applications for real-time exercise form detection using a wearable IMU. For example, it could be used to provide feedback to athletes or fitness enthusiasts on their form during a workout, or to monitor the intensity and quality of an exercise program. It could also be used in rehabilitation settings to track progress and ensure that exercises are being performed correctly.

This paper describes a system that uses a wearable inertial measurement unit (IMU) to detect the type of exercise being performed in real-time. The system was trained on a dataset of 10 different exercises and was able to achieve an accuracy of 95% in detecting the correct exercise.

##### Exercise Form Detection Using a Wearable Sensor: A Feasibility Study

A feasibility study on exercise form detection using a wearable sensor would aim to determine whether it is possible to accurately detect and classify different types of exercise using a wearable sensor such as an inertial measurement unit (IMU). The study would likely involve collecting data from the wearable sensor while subjects perform a range of exercises, and using machine learning algorithms or other techniques to analyze the data and detect the exercise being performed.The feasibility study would likely consider a number of factors that could affect the accuracy of the exercise form detection system, such as the specific sensors used, the location of the wearable sensor on the body, the type and intensity of the exercises being performed, and any noise or variability in the sensor data. The study would then evaluate the performance of the system in detecting and classifying different exercises, and report on the feasibility and potential accuracy of the approach. This paper presents a feasibility study of using a wearable sensor to detect the form of exercise being performed. The system was able to accurately detect the type of exercise being performed with an accuracy of 87.5%.

##### Exercise Form Detection Using Machine Learning and Smartwatch Sensors

It is possible to use machine learning techniques and data from smartwatch sensors to detect and classify different types of exercise. Smartwatches typically include a variety of sensors such as accelerometers, gyroscopes, and heart rate monitors, which can provide data on a person's movement and physiological responses during physical activity.To detect and classify different types of exercise using machine learning and smartwatch sensors, researchers or developers can collect data from the sensors while subjects perform a range of exercises, and use this data to train a machine learning model. The model can then be used to classify new exercises based on the data from the smartwatch sensors.This paper presents a machine learning approach for detecting the form of exercise being performed using data from smartwatch sensors. The system was able to achieve an accuracy of 87% in detecting the correct exercise.

**Chapter 3**

# PROBLEM IDENTIFICATION

Fire detection is crucial task for the safety of people. To prevent damages caused by fire, several fire detection systems were developed. There are many technologies available for smoke and fire detection but still society is lacking reliable and accurate methods to predict smoke and fire at early stages, this deficiency may then lead to dangerous situations.

Fire outbreak is the common issue happening everywhere and the damage caused by this type of incidents is tremendous towards nature and human. Vision based fire detection system have recently gained popularity as compared to traditional sensor based fire detection system. However, the detection process by image processing technique is very tedious. We proposed a fire detection algorithm using Convolutional Neural Networks to achieve high-accuracy fire image detection, which is compatible in detection of fire by training with datasets.



### Fig. 3.1 Forest Fire and Smoke

Here in this project we are creating a system to monitor fire and smoke patterns and report it to the forest offices. It plays an important role in saving environment and wild life. By considering the various aspects of the problem, we consider images including fire and smoke patterns in various situations and backgrounds to train the system and achieve higher accuracy and efficiency in detection.

**Chapter 4**

# OBJECTIVES

The main objective of the proposed project is to find a solution for prevention of wild fire by detecting fire and/or smoke at the earliest stages and notify the concerned forest offices.

The features that we are implementing in this project are as follows:

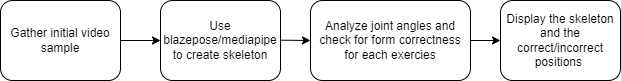
* To capture the image and Classifying Them Using Image Processing. Here we take images captured from the camera. These images are processed and are classified using Convolution Neural Network (CNN). The classification of these images is based on whether they detect fire and/or smoke patterns.
* To Detect Fire and Smoke in Forest and inform the concerned authorities. Here after the captured images are processed and classified, if those images are detected to have fire and/or smoke patterns, the concerned forest officers and registered farmers are notified about the same through various massages. To send notification and messages to farmers and forest officers we use Twillio Messenger.
* By sending the messages and notifying the farmers and forest officers, we can take immediate measures to stop or prevent the disaster of fire that might go beyond human control and lead to loss of huge amount of environment, wild life and even cause harm to humans.

**Chapter 5**

# METHODOLOGY

## SYSTEM IMPLEMENTATION

##### Structural Design

****

**Figure 5.1 Structural chart of the proposed system.**

The figure 5.1 shows that the proposed system involves the following steps. First step involves pre-processing of captured video. The pre-processed video undergoes feature extraction, where various features of the body parts are extracted and certain algorithms are applied. The data that is stored is compared with the pre-processed image and approximate result is generated**.**

##### Convolution Neural Network:

Convolutional neural network is the special type of feed forward artificial neural network in which the connectivity between the layers are inspired by the visual cortex. Convolutional Neural Network (CNN) is a class of deep neural networks which is applied for analyzing visual imagery. They have applications in image and video recognition, image classification, natural language processing etc. Convolution is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel. Each input image will be passed through a series of convolution layers with filters (kernels) to produce output feature maps. Here is how exactly the CNN works.

Basically, the convolutional neural networks have 4 layers that is the convolutional layers, ReLU layer, pooling layer and the fully connected layer.

##### Convolutional Layer

In convolution layer after the computer reads an image in the form of pixels, then with the help of convolution layers we take a small patch of the images. These images or patches are called the features or the filters. By sending these rough feature matches is roughly the same position in the two images, convolutional layer gets a lot better at seeing similarities than whole image matching scenes. These filters are compared to the new input images if it matches then the image is classified correctly. Here line up the features and the image and then multiply each image, pixel by the corresponding feature pixel, add the pixels up and divide the total number of pixels in the feature. We create a map and put the values of the filter at that corresponding place. Similarly, we will move the feature to every other position of the image and will see how the feature matches that area. Finally, we will get a matrix as an output.

##### ReLU Layer

ReLU layer is nothing but the rectified linear unit, in this layer we remove every negative value from the filtered images and replaces it with zero. This is done to avoid the values from summing up to zeroes. This is a transform function which activates a node only if the input value is above a certain number while the input is below zero the output will be zero then remove all the negative values from the matrix.

##### Pooling Layer

In this layer we reduce or shrink the size of the image. Here first we pick a window size, then mention the required stride, then walk your window across your filtered images. Then from each window take the maximum values. This will pool the layers and shrink the size of the image as well as the matrix. The reduced size matrix is given as the input to the fully connected layer.

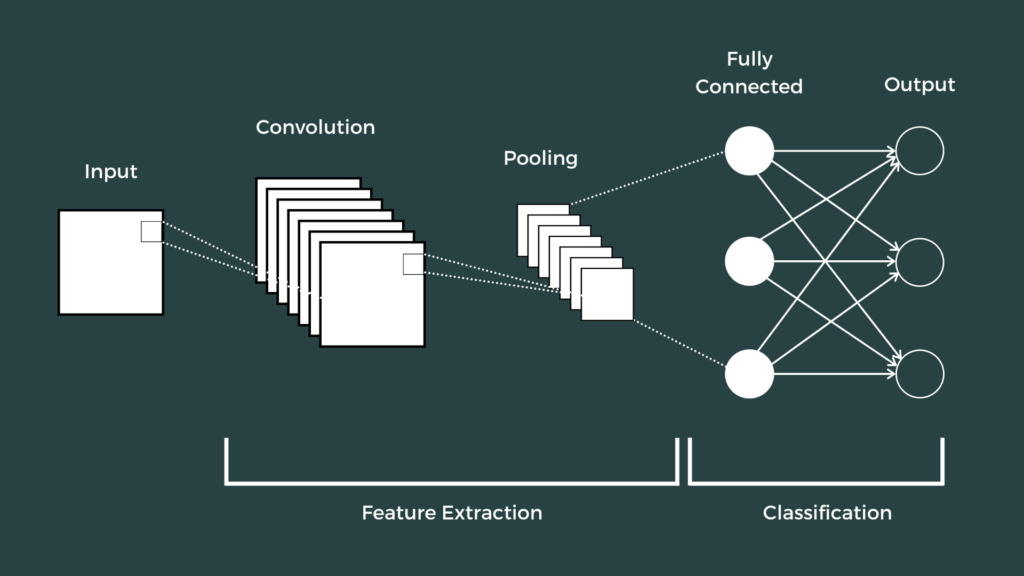
##### Fully Connected Layer

We need to stack up all the layers after passing it through the convolutional layer, ReLU layer and the pooling layer. The fully connected layer used for the classification of the input image. These layers need to be repeated if needed unless you get a 2x2 matrix. Then at the end the fully connected layer is used where the actual classification happens.

##### Typical CNN Architecture

CNN architecture is inspired by the organization and functionality of the visual cortex and designed to mimic the connectivity pattern of neurons within the human brain. The neurons within a CNN are split into a three-dimensional structure, with each set of neurons analyzing a small region or feature of the image.

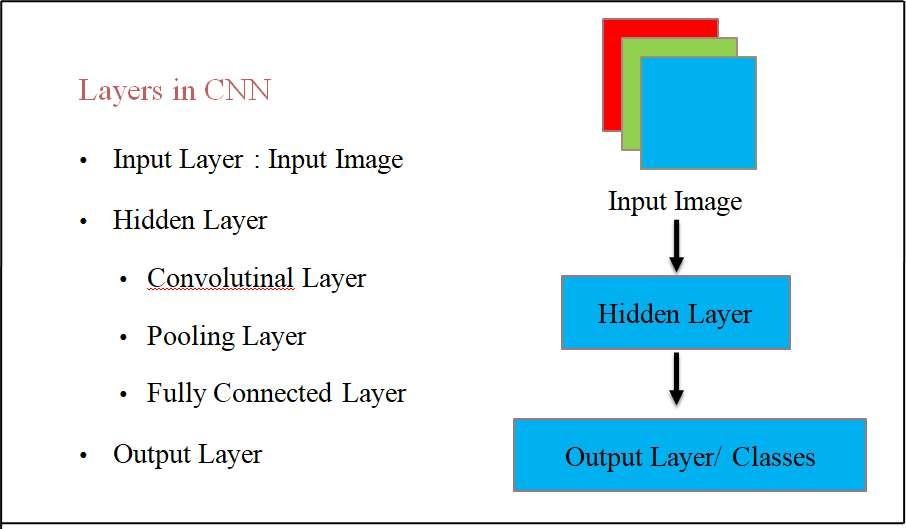
In other words, each group of neurons specializes in identifying one part of the image. CNNs use the predictions from the layers to produce a final output that presents a vector of probability scores to represent the likelihood that a specific feature belongs to a certain class. Figure 5.2 shows the Typical CNN Architecture.



##### Figure 5.2 Typical CNN Architecture

A CNN is composed of several kinds of layers:

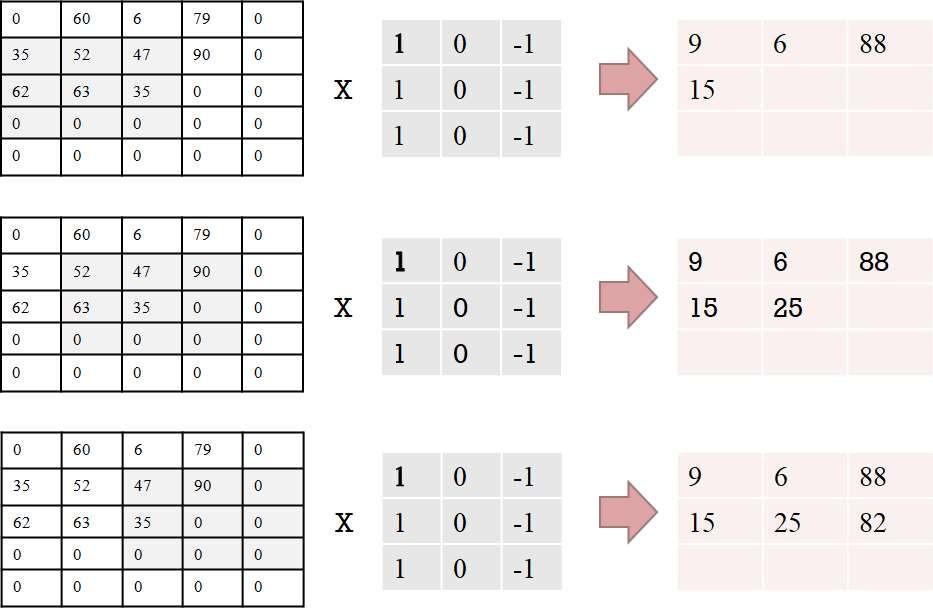
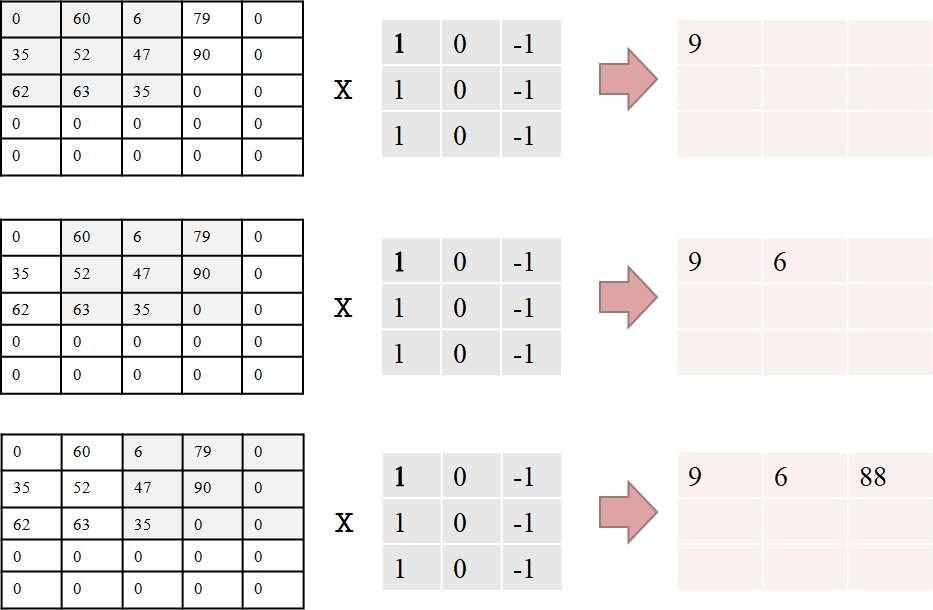
* **Convolutional layer**- In convolution layer after the computer reads an image in the form of pixels, then with the help of convolution layers we take a small patch of the images. These images or patches are called the features or the filters. By sending these rough feature matches is roughly the same position in the two images, convolutional layer gets a lot better at seeing similarities than whole image matching scenes. It creates a feature map to predict the class probabilities for each feature by applying a filter that scans the whole image, few pixels at a time.
* **Pooling layer (down sampling)**-scales down the amount of information the convolutional layer generated for each feature and maintains the most essential information (the process of the convolutional and pooling layers usually repeats several times).
* **Fully connected layer**- “Flattens” the outputs generated by previous layers to turn them into a single vector that can be used as an input for the next layer. Applies weights over the input generated by the feature analysis to predict an accurate label.
* **Output layer**-generates the final probabilities to determine a class for the image.

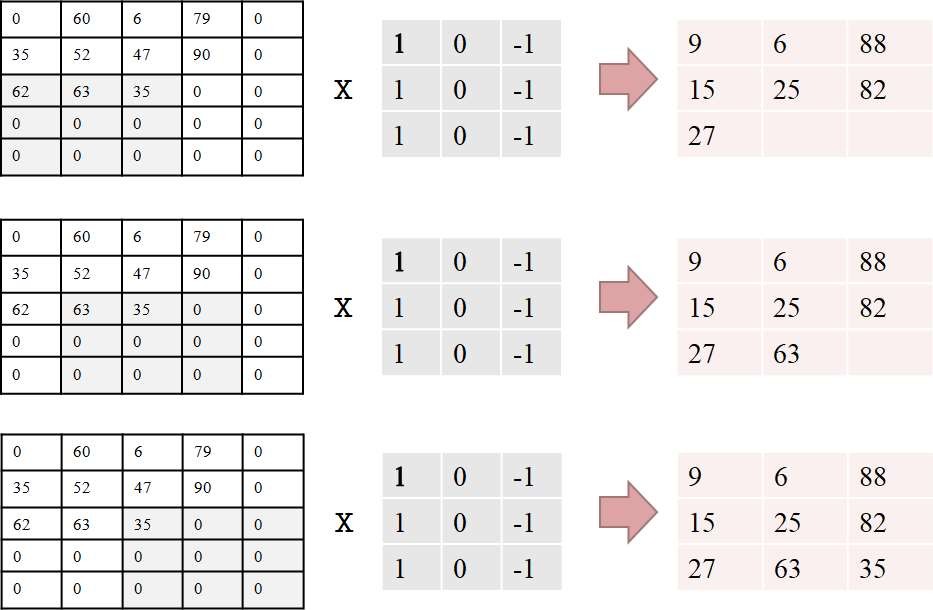
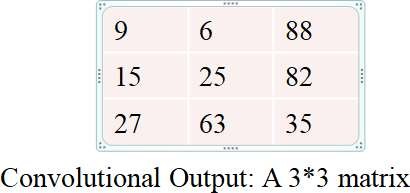


**Fig. 5.3 Layers in CNN**

##### Convolutional Layer

Convolutional Layer is the first step in CNN, here 3\*3 part of the given matrix which was obtained from High-pass filter is given as input. That 3\*3 matrix is multiplied with the filter matrix for the corresponding position and their sum is written in the particular position. This is shown in the below figure. This output is given to pooling layer where the matrix is further reduced. Figure 5.4 shows the Convolutional Layer.

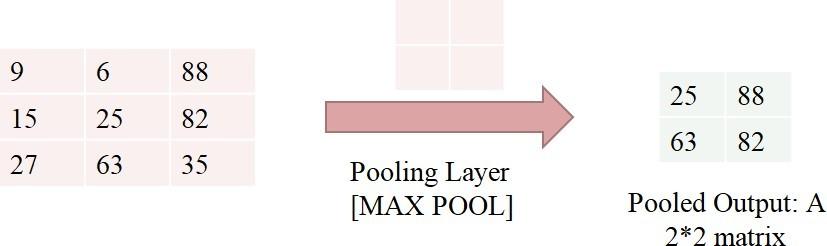




**Figure 5.4 Convolutional Layer**

Convolution is followed by the rectification of negative values to 0s, before pooling. Here, it is not demonstrable, as all values are positive. In fact, multiple iterations of both are needed before pooling.

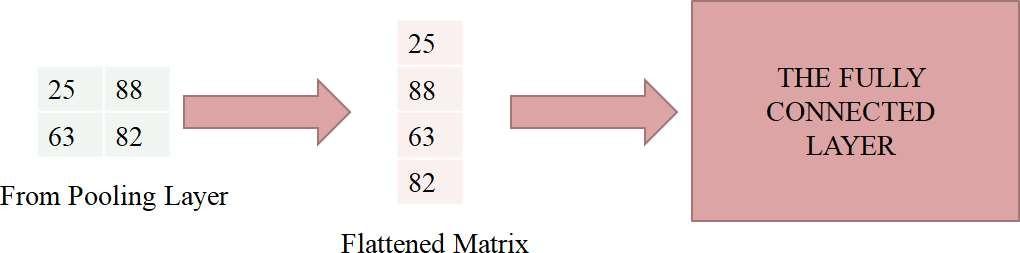
* + - **Pooling Layer**

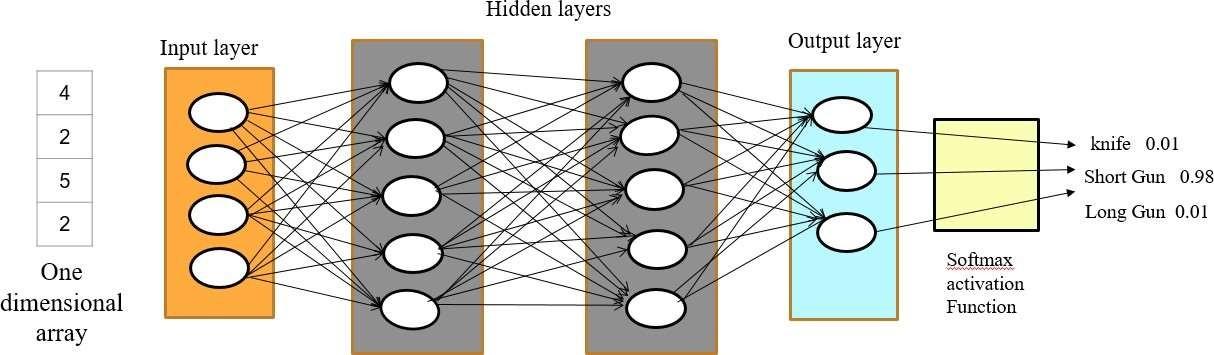


**Figure 5.5 Pooling Layer**

In Pooling layer 3\*3 matrix is reduced to 2\*2 matrix, this is done by selecting the maximum of the particular 2\*2 matrix for the particular position. Figure 4.16 shows the Pooling Layer.

* + - * **Fully connected layer and Output Layer**





**Figure 5.6 fully connected layer and Output Layer**

The output of the pooling layer is flattened and this flattened matrix is fed into the Fully Connected Layer. In the fully connected layer there are many layers, Input layer, Hidden layer and Output layers are parts of it.

**Chapter 6**

* 1. ***System requirements***

# SYSTEM DESIGN

System requirement specifications gathered by extracting the appropriate information to implement the system. It is the elaborative conditions which the system need to attain. Moreover, the SRS delivers a complete knowledge of the system to understand what this project is going to achieve without any constraints on how to achieve this goal. This SRS not providing the information to outside characters but it hides the plan and gives little implementation details.

###### Specific Requirement

* + - Require access to a client session of Python and Keras toolbox for job submission.
    - A shared file system between user desktops and cluster.
    - Maximum of Python worker per physical CPU core.

###### Hardware Requirement

* + - Processor: Intel core
    - Processor Speed: 1.86 GHz.
    - RAM: 4GB+
    - Hard Disk Space: 500 GB+
    - Monitor: 15 VGA Color

###### Software Requirement

* + - Operating system: Windows 10
    - Coding Language: Python
    - Software Tool: Keras
    - Toolbox: Video processing toolbox
  1. **Functional and Non Functional requirements:**

###### FUNCTIONAL REQUIREMENTS:

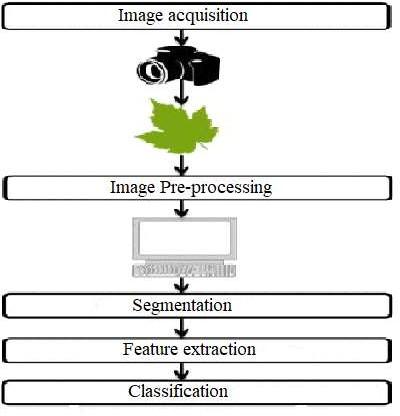
* + - * System should do minimal computations on its own.
      * System should capture image.
      * System should automatically detect body-points on its own.
      * System should automatically intimate.

###### NON-FUNCTIONAL REQUIREMENTS:

* + - * The Camera is used take Video
      * Requirement data will be stored in the python database
      * System should be reliable
      * System should be flexible for future enhancements
      * System should be Easily Implementable
      * System should be Easy to Implement.
      * Cost of Implementation should be low.

# System Design

* + - The proposed system includes five modules. The initial stage is the image/video acquisition stage through which the real world sample is recorded in its digital form using a digital camera or webcam.
    - In the next stage of the research image was subjected to a pre-processing stage. Making use of it the size and complexity of the image is reduced.
    - The precise digital information is subjected to segmentation process which separates the physical body of the user from the background.
    - The feature extraction aspect of an image analysis focuses on identifying inherent features of the objects present within an image,
    - Classification maps the data into specific groups or classes.



##### Fig 6.1 Project Modules

High-level design (HLD) explains the architecture that would be used for developing a software product. The architecture diagram provides an overview of an entire system, identifying the main components that would be developed for the product and their interfaces. The HLD uses possibly nontechnical to mildly technical terms that should be understandable to the administrators of the system. In contrast low level design further exposes the logical detailed design of each of these elements for programmers.

High level design is the design which is used to design the software related requirements. In this chapter complete system design is generated and shows how the modules, sub modules and the flow of the data between them are done and are integrated. It consists of very simple phases and shows the implementation process.

***Design Consideration:***

The design consideration briefs about how the system behaves for the boundary environments and what action should be taken if the abnormal case happens. Some of the design considerations are data collection, pre-processing methods and Classification and prediction.

The design considerations are formulated to bring to the attention of the designers in applying the universal accessibility design principles and requirements to buildings and facilities. They can also be used to identify barriers in existing systems.

The proposed system has the following steps-

1. Video Pre-Processing.
2. Identification
3. Feature Extraction
4. Body-point Detection
5. Form correction

###### Video Pre processing

The video processing is a mechanism that focuses on the manipulation of images which are derived from the frames of the video in different ways in order to enhance the image quality. Images are taken as the input and output for image processing techniques. It is the analysis of image-to-image transformation which is used for the enhancement of image. Firstly, we convert RGB image to gray scale image. It helps to reduce the complexity in the image and also make the work easy. Then by using min-max scalar method converts the gray scale values into binary values. The obtained binary values are taken as the input for the further process. In the obtained binary matrix consider one value regions as white and zero value region black. By using these values, the region of interest can be identified. So that the values are useful for feature extraction and identification of region of interest.

###### Identification:

In this stage identify the region which needs to proceed for further process, it is involved in the identification of the particular region of the image that is used for the further process like feature extraction and classification of the images. The output of the pre-processing step is given as the input for the identification process. This process is based on the binary values obtained in the pre- processing step. The region with black are consider as region of interest. The region of interest obtained by the pre-processing of the images. That region is considered as proceeding part of the image from which body points such as joints and skeletal structure will be identified. The identified images are given to the feature extraction process.

###### Feature Extraction

In this stage extract the required feature from the identified region which are obtained from the previous step. That region is compressed by converting reduced size matrix to control over fitting. The reduction of the matrix size helps in reduce the memory size of the images. Then the flattening process is applied to the reduced matrix, in which the reduced matrix is converted to one-dimension array, which is used for final detection.

###### Body-points Detection:

The methodology is proposed using CNN (Convolutional Neural Networks) model. The image datasets are created from body images captured by videos and converting it into frames. Some images in the dataset are added to dataset taken from internet. This is given as a input to the convolutional layer. In these operations, several kernels of different sizes are applied on the input data to generate feature maps. The model consists of 64 convolution filters of size 3x3 each. The feature maps go through a ReLU activation function. This function updates positive portion of the feature map rapidly. These features maps are input to the next operation known as max pooling.

These feature maps are subjected again to convolution layer and pooling layer which has kernel size of 3x3. Then a flatten layer which converts 2D feature maps into a vector that can be fed to fully connected layer. Among these three main operations, the convolution and fully connected layers contain neurons whose weights are learnt and adjusted for better representation of the input data during training process. A dense layer represents a matrix vector multiplication. The values in the matrix are the trainable parameters, which are updated during back propagation. Therefore, you get an m dimensional vector as output. System Architecture:

A system architecture is the conceptual model that defines the structure, behavior, and more views of a system. A system architecture can consist of system components and the sub-systems developed, that will work together to implement the overall system.

The video sample is first gathered by MediaPipe’s calculator graphs which are used to process streams of video and audio frames by receiving successive packets with increasing timestamps. The solution utilizes a two-step detector-tracker ML pipeline.

Using a detector, the pipeline first locates the person/pose region-of-interest (ROI) within the frame. The tracker subsequently predicts the pose landmarks and segmentation mask within the ROI using the ROI-cropped frame as input.

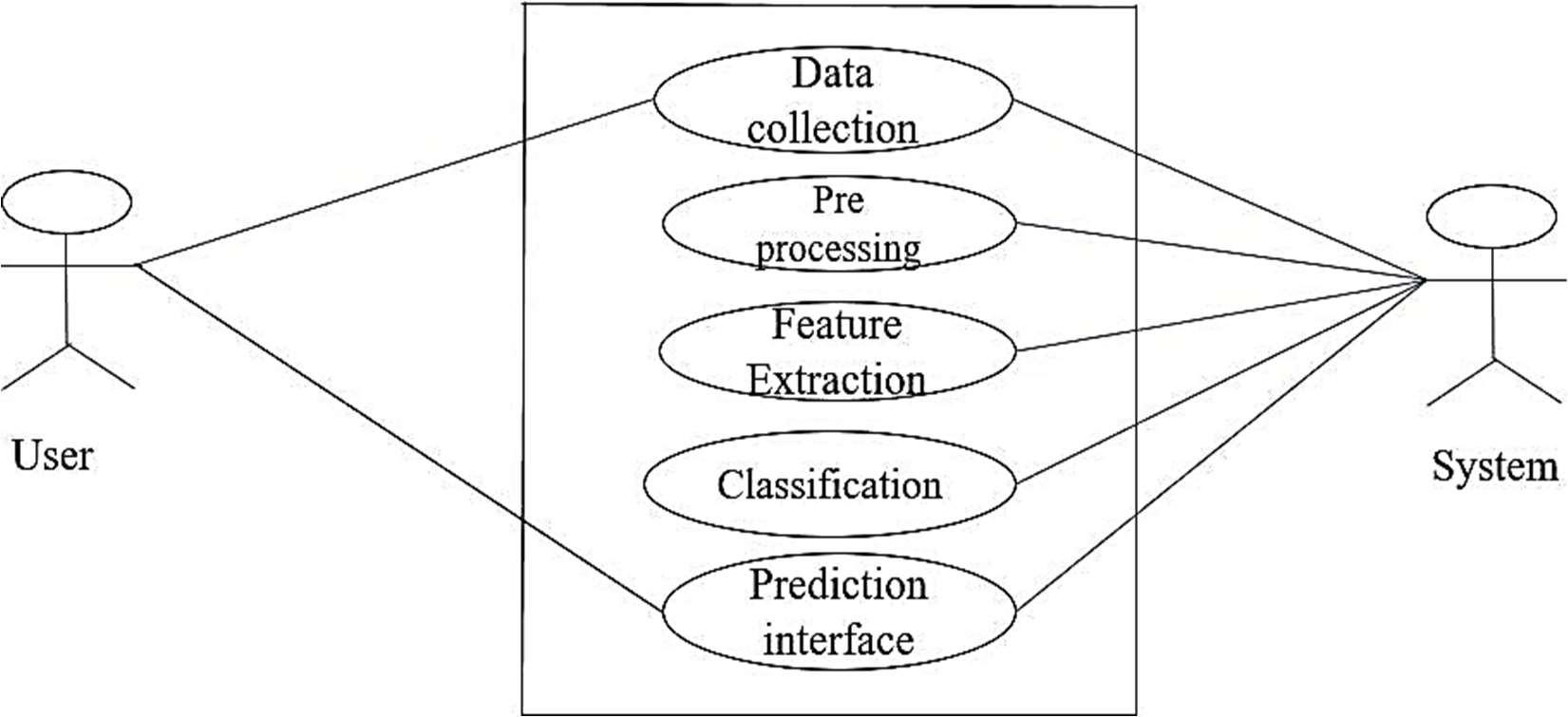
A skeleton of the region of interest is created by MediaPipe which is trained on the Objectron dataset. Finally, the created skeleton is checked manually for each exercise by defining the joint angles required for it to be executed correctly.



**Figure 6.2 System Architecture of the body-points identification**

The figure 6.2 shows the system architecture for the proposed system. The input image is pre- processed and converted to gray scale image to get the clear vision of the image. Then it will be converted into binary values. In the next step identifies the part which needs to proceed further. Then required feature are extracted by In the CNN convolution layer.

#### Specifications using use case diagrams:

A use case is a set of scenarios that describing an interaction between a source and a destination. A use case diagram displays the relationship among actors and use cases. The two main components of a use case diagram are use cases and actors.

##### Figure 6.3 Use case diagram of the proposed system

The figure 6.3 shows that the use case diagram in the Unified Modelling Language (UML) is a type of behavioral diagram defined by and crated from a use-case analysis. Here the user can collect the data and load the data to the system. The system can store the data for training and testing

the model, here system is taken as actor. The training and testing data are given to the CNN for further classification.

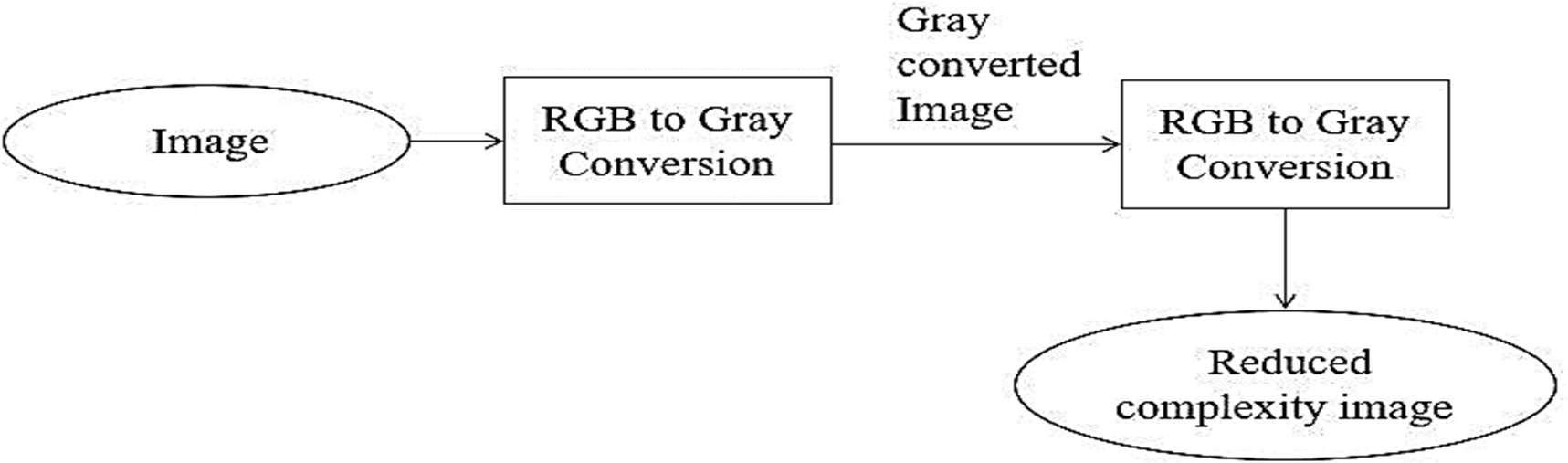
* 1. ***Data flow diagram***

A data flow diagram (DFD) is graphic representation of the "flow" of data through an information system. A data flow diagram can also be used for the visualization of data processing (structured design). It is common practice for a designer to draw a context level DFD first which shows the interaction between the system and outside entities.

Data flow diagrams show the flow of data from external entities into the system, how the data moves from one process to another, as well as its logical storage. There are only four symbols:

1. Squares representing external entities, which are sources and destinations of information entering and leaving the system.
2. Rounded rectangles representing processes, in other methodologies, may be called 'Activities', 'Actions', 'Procedures', 'Subsystems' etc. which take data as input, do processing to it, and output it.
3. Arrows representing the data flows, which can either, be electronic data or physical items. It is impossible for data to flow from data store to data store except via a process, and external entities are not allowed to access data stores directly.
4. The flat three-sided rectangle is representing data stores should both receive information for storing and provide it for further processing.
5. It is also used to analyses a particular problem and the solution for it in steps.
6. A user loads the data and the system reads the data provided by the user.
7. Based on feature extraction and classifier the model will be trained and tested.

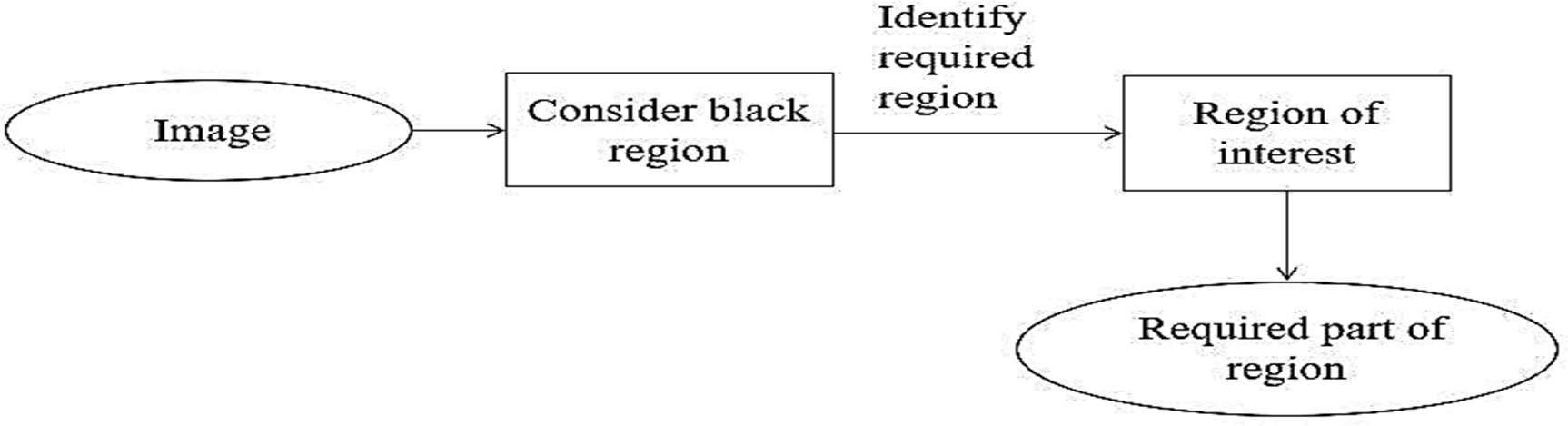
###### Data Flow Diagram for Pre-processing



***Figure 6.4 Data flow diagram for pre processing***

The figure 6.4 shows that the image is given as input. As we giving the color image so that RGB image is converted into gray scale values to reduce complexity in the image. For efficient feature extraction gray scale values are converted into binary values. Then the image with reduced complexity is send to the next process.

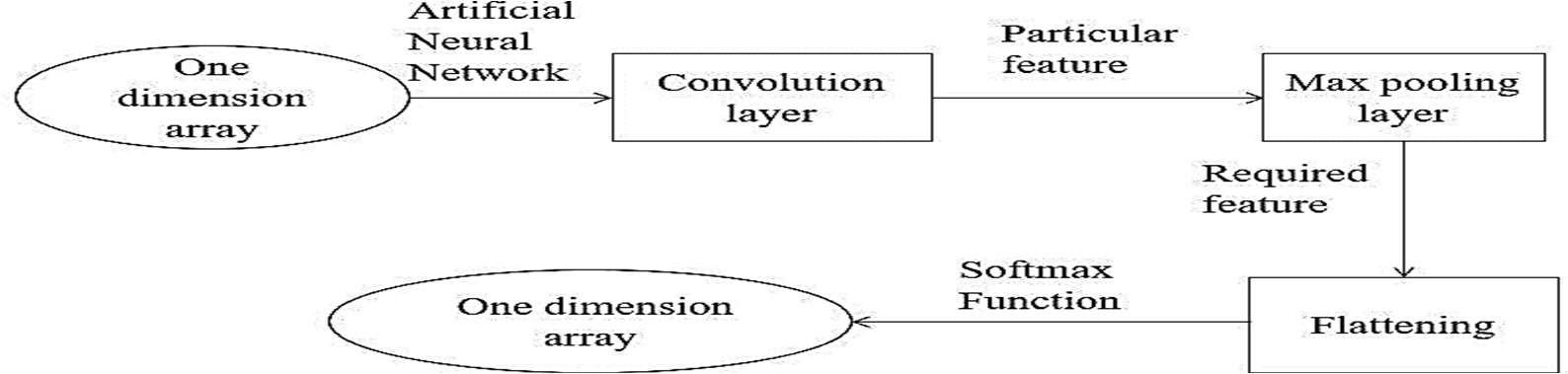
* ***Data Flow Diagram for Identification***



**Figure 6.5 Data flow diagram for Identification**

The figure 6.5 shows that the image with reduced complexity is considered as input. Here the region with the value of one is considered as black that region is considered for next process.

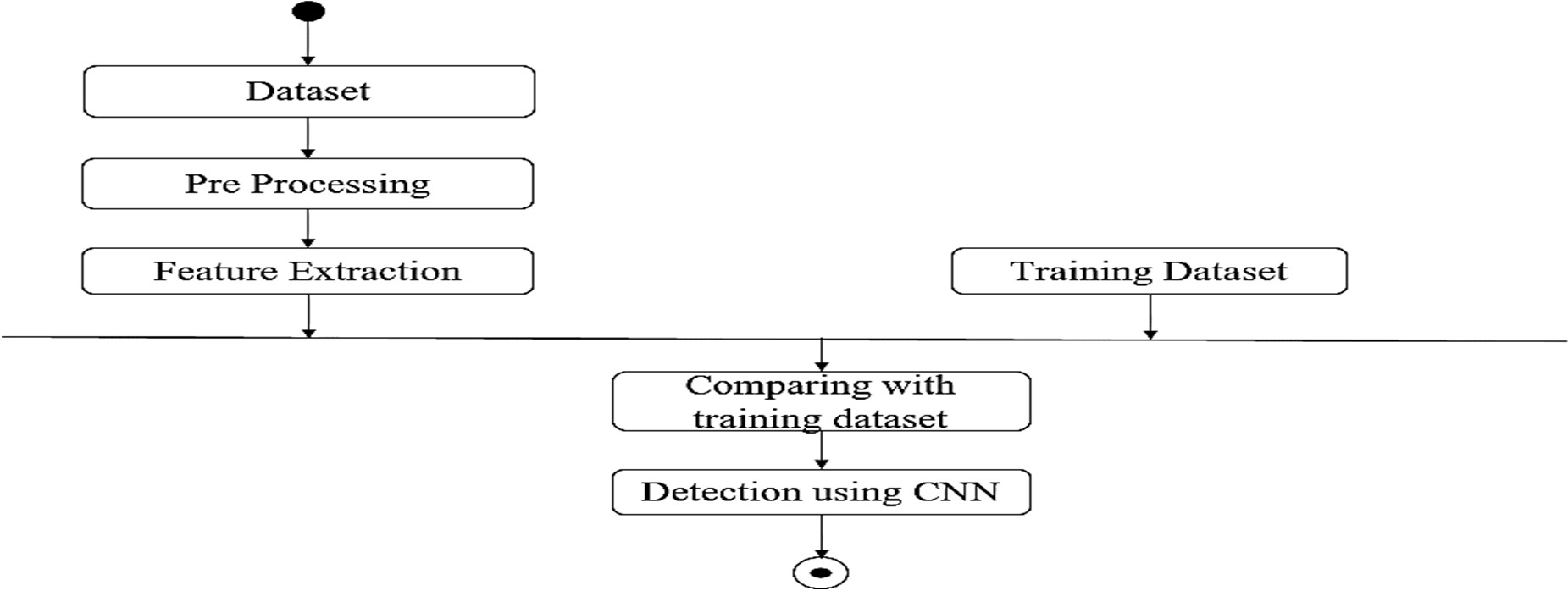
###### Data Flow Diagram for Classification and Detection



**Figure 6.7 Data flow diagram for Classification and Detection**

The figure 6.7 shows that the one-dimension array is send to fully connected layer of CNN. Artificial neural network method is applied to this layer. Firstly, one-dimension array is sent to input layer. Some particular feature which is required for the detection is identified by the hidden layer of ANN. The continue connection from hidden layer to output layer will help to identify accurate result. By considering all the features output layer gives the result with some predictive value.

##### Activity Diagram:



**Figure 6.8 Activity diagram for body-points detection**

The figure 6.8 shows the activity diagram of body-point detection. Here single circle indicates the start of the process and double circle indicates the end of the process. Here the pre- processing of the image by converting the RGB to grayscale image and feature extraction is done by first layer that is convolution layer of the neural network and detection done by using fully connected layers of convolutional neural network.

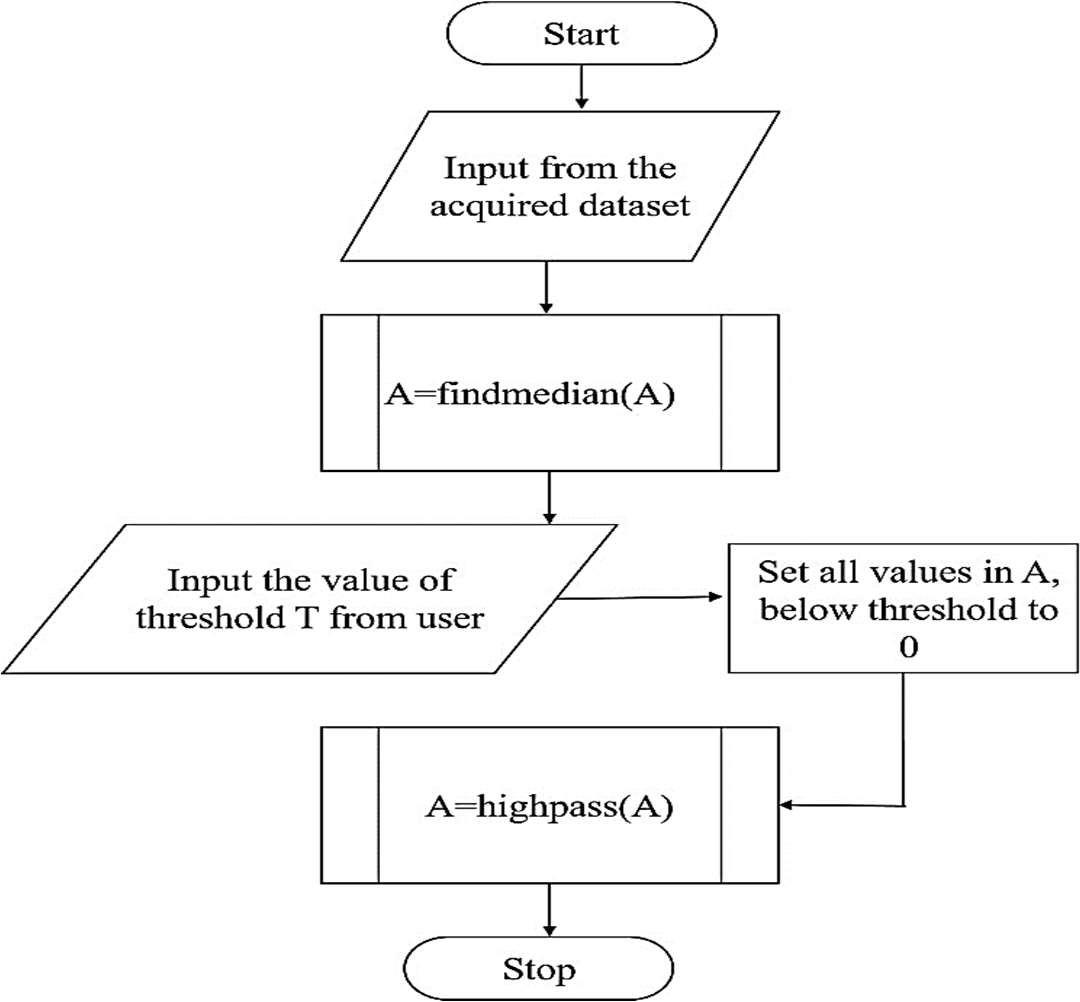
* 1. **Detailed Design**

##### The flowchart for Data Acquisition

**Figure 6.11 Flowchart for data acquisition**

The flowchart for collecting data is as depicted in the figure 6.11. The data set is collected from a source and a complete analysis is carried out. The image is selected to be used for training/testing purposes only if it matches our requirements and is not repeated.

##### Flowchart for Pre-Processing the Data Set

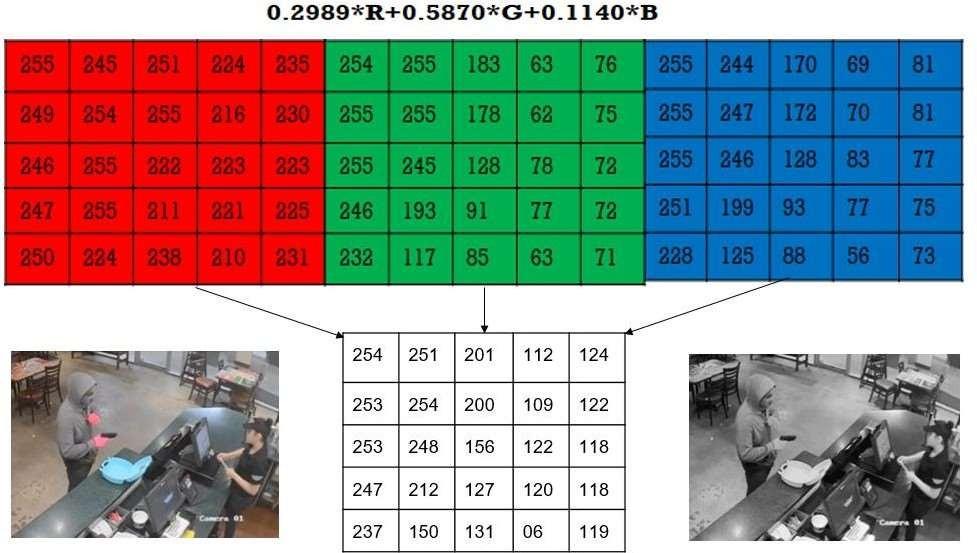


**Figure 6.12 Flowchart for the preprocessing module**

The figure 6.12 shows the flowchart for the pre-processing of the images received from the output of the previous step. This involves converting the image from the RGB format to grayscale to ease processing, the use of an averaging filter to filter out the noise, global basic thresholding to remove the background and consider only the image and a high- pass filter to sharpen the image by amplifying the finer details.

##### Conversion from RGB to Greyscale

The first step in pre-processing is converting the image from RGB to Greyscale. It can be obtained by applying the below formula to the RGB image. The figure 6.13 depicts the Conversion from RGB to grayscale.



##### Figure 6.13 Conversion from RGB to grayscale

**Advantages of converting RGB color space to gray**

* + To store a single-color pixel of an RGB color image we will need 8\*3 = 24 bits (8 bit for each color component).
  + Only 8 bit is required to store a single pixel of the image. So we will need 33 % less memory to store grayscale image than to store an RGB image.
  + Grayscale images are much easier to work within a variety of task like In many morphological operation and image segmentation problem, it is easier to work with single layered image (Grayscale image) than a three-layered image (RGB color image).
  + It is also easier to distinguish features of an image when we deal with a single layered image

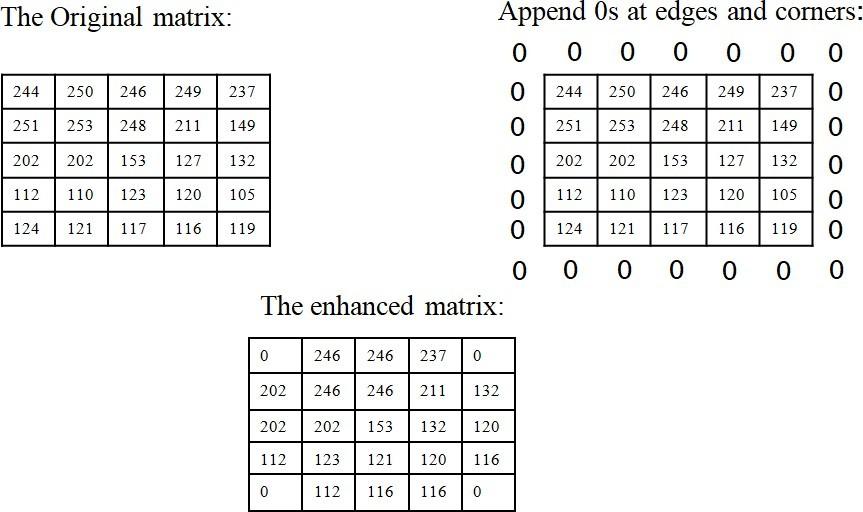
##### Noise removal

Noise removal algorithm is the process of removing or reducing the noise from the image. The noise removal algorithms reduce or remove the visibility of noise by smoothing the entire image leaving areas near contrast boundaries. Noise removal is the second step in image pre- processing. Here the grayscale image which was obtained in the previous step is given as input. Here we are making use of Median Filter which is a Noise Removal Technique.

##### Median Filtering

The median filter is a non-linear digital filtering technique, often used to remove noise from an image or signal.

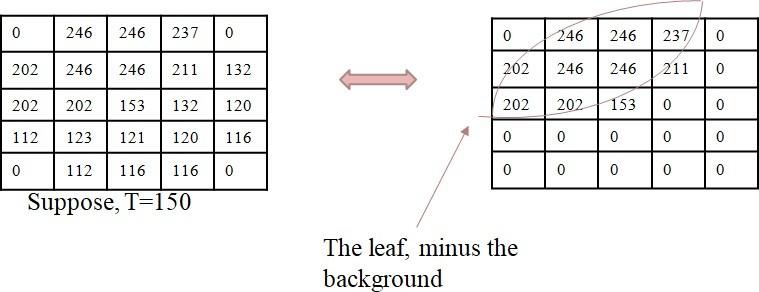
Here 0’s are appended at the edges and corners to the matrix which is the representation of the grey scale image. Then for every3\*3 matrix, arrange elements in ascending order, then find median/middle element of those 9 elements, and write that median value to that particular pixel position. The figure 6.14 depicts Noise filtering using Median Filter.



##### Figure 6.14 Noise filtering using Median Filter

* **Basic Global Thresholding**

Thresholding is a type of image segmentation, where we change the pixels of an image to make the image easier to analyze. A (i, j) is greater than or equal to the threshold T, retain it. Else, replace the value by 0.

Here, the value of T can be manipulated in the frontend, to suit the varying needs of different images. We use trial and error method here to obtain threshold value which may be best suited for us. Thresholding using basic global thresholding is shown in figure 6.15

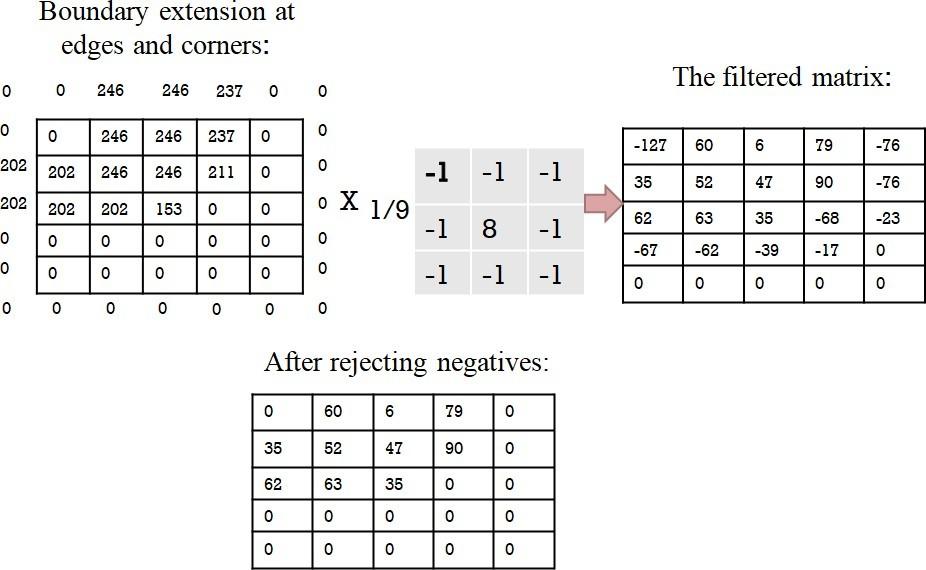
##### Figure 6.15 Thresholding using Basic global Thresholding

* **Image Sharpening**

Image sharpening refers to any enhancement technique that highlights edges and fine details in an image, Increasing yields a more sharpened image.

##### High-Pass Filtering

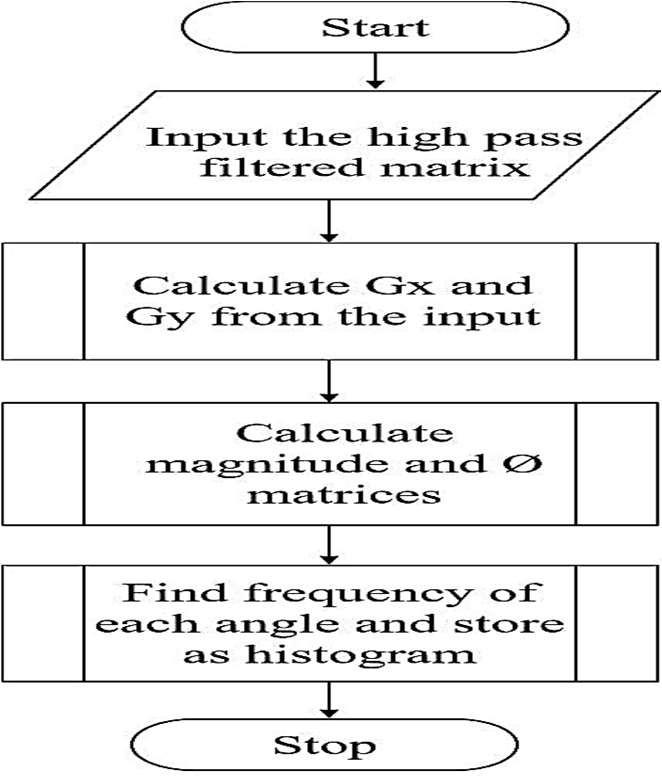
A high-pass filter can be used to make an image appear sharper. These filters emphasize fine details in the image. Here the output from the thresholding is given as input. Here, we are making use of a filter, first we append the nearest values to pixels at the boundary pixels. The figure 6.16 depicts Image Sharpening using High-Pass Filter



##### Figure 6.16 Image Sharpening using High-Pass Filter

We multiply the elements of the 3\*3 input matrix with the filter matrix, this can be represents as A(1,1)\*B(1,1), in this way all the elements in the 3\*3 are multiplied and their sum id divided by 9, which gives the value for the particular pixel position. In the same way the values of all the pixel positions are calculated. The negative values are considered as zero, as there can be no such thing as negative illumination.

##### Flowchart for Feature Extraction



**Figure 6.17 Flowchart for Feature Extraction**

Here, we use a method called Histogram Orientation Gradient (HOG) to extract the features from the preprocessed image received as input. It involves multiple steps like finding Gx and Gy, which are gradients about each pixel in the x and y axes. Then, these gradients are substituted in relevant formulae to get the magnitude and gradient of the pixel’s orientation. Then, the angles and their respective frequencies are plotted to form a histogram, which is the output of this module. The flowchart for feature extraction model is shown in figure 6.17

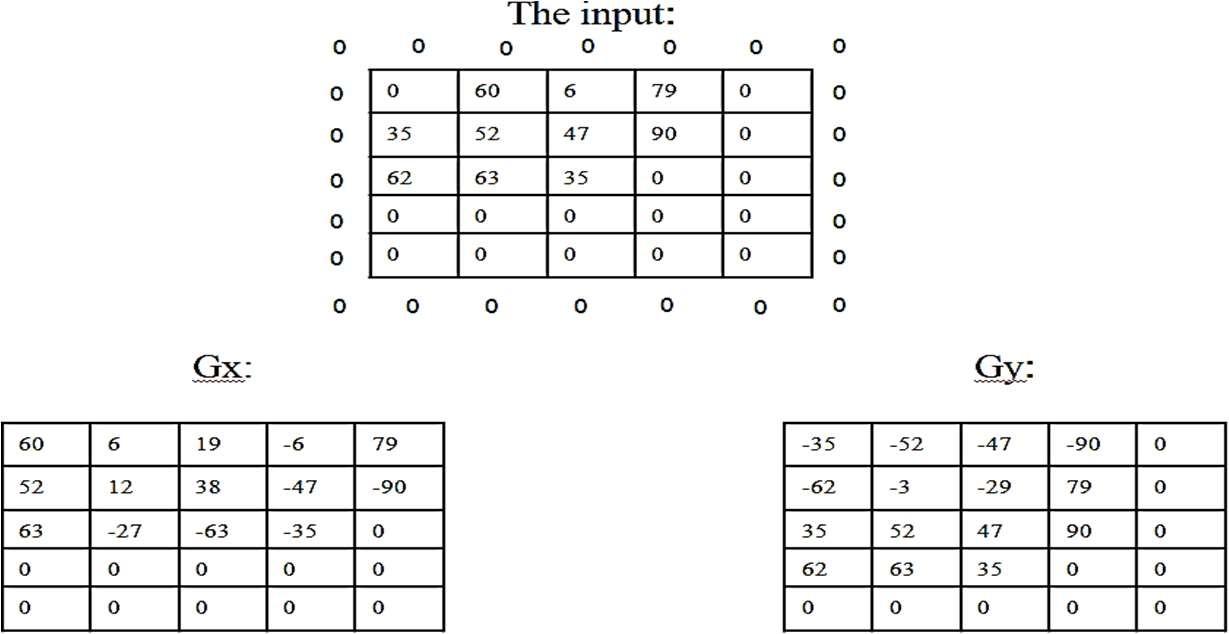
##### Feature Extraction

Feature extraction is a process of dimensionality reduction by which an initial set of raw data is reduced to more manageable groups for processing.

##### Histogram Orientation Gradient

The Histogram of Oriented Gradients (HOG) is a feature descriptor used in computer vision and image processing for the purpose of object detection. The technique counts occurrences of gradient orientation in in localized portions of an image.

Here 0’s are appended at the edges and corners to the matrix. Then Gx and Gy are calculated. Gx is calculates as Gx = value on right –value on left and Gy is calculated as Gy=value on top-value on left. Figure 6.18 shows Gx and Gy in HOG.

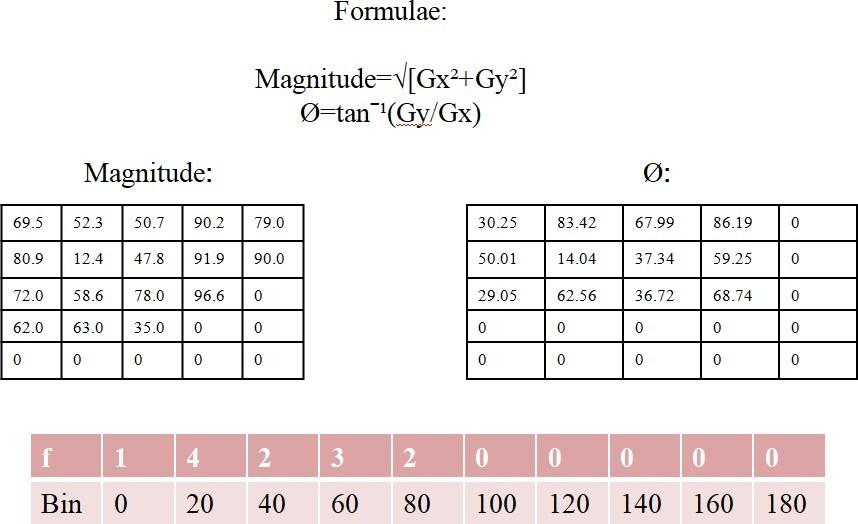


##### Figure 6.18 Gx and Gy in HOG

Then, using the formula given in figure 6.19 Magnitude and the orientation are calculated. Feature Extraction using HOG is shown in figure 6.19 Magnitude is the illumination and degree the orientation is the angle of orientation

After the angle of orientation is calculated, the frequency the angles for the particular intervals area are noted and they are given as input for the classifier. Here we zeroes are not

considered for finding the frequency. For example for the interval from 40 to 59 there are 2 occurrences, so we are writing the frequency as 2.



##### Figure 6.19 Feature Extraction using HOG

* **Flowchart for classification and detection**

In CNN, we take the output from the high-pass filter as input, leaving out feature extraction, as CNN is a classifier which simply has a feature extracting process of its own, using convolution, rectification and pooling as the 3 sub-modules, which work in iterations to give out a final comparison matrix.

##### Form correction-

##### This is the final stage in the process where after identification of the body-points in the images using CNNs, we detect the required angles for each individual exercise manually and indicate to the user whether the angles between their body-points are accurate for the exercise by displaying a red-line for the wrong angle when the user’s points exceed or fall below the required angle for that particular exercise.

# CONCLUSION

This project provides motivation for the user to exercise by displaying timely achievements and milestones. By maintaining a record of all previous exercises, it can create and present the user’s routines and activities in the form of reports.

The project provides users with Graphical Statstics of their activities in the form of reports which are easy to analyse and work upon . These reports could be used by users to develop better Routines or to make it a regular daily habit and thus providing motivation to better health .

This project presents a labeled dataset for training Deep Learning models to help correct posture in Physical exercises . It proposes a solution based on Convolutional Neural Networks and evaluates the performance under different environmental conditions (background, camera angle, distance , movements , lighting etc.).

The project focuses to display a notification when the user reaches a certain milestone, such as completing a certain number of workouts or reaching a certain number of steps. This makes the user to feel a sense of accomplishment and encourages them to keep up their exercise routine. Additionally, the model offers rewards for reaching certain milestones, such as discounts on gym memberships or access to special content. This can further motivate the user to stay active and engaged with the project .

This project employs a Framework called Mediapipe which is open sourced by Google and is a cross platform pipeline framework to build custom Machine learning solutions for live and streaming media . Mediapipe is also used in this project to make a completely new Dataset from scratch which involves all the movement and exercise videos .

# REFERENCES

**[1]**  <https://mediapipe.dev/>

**[2]** A. Rahmadani, B. S. Bayu Dewantara and D. M. Sari, "Human Pose Estimation for Fitness Exercise Movement Correction," 2022 International Electronics Symposium (IES), 2022, pp. 484-490, doi: 10.1109/IES55876.2022.9888451.

**[3]** Militaru, C., Militaru, M.D. and Benta, K.I., 2020, October. Physical Exercise Form Correction Using Neural Networks. In *Companion Publication of the 2020 International Conference on Multimodal Interaction* (pp. 240-244).

**[4]**  Biebl, J.T., Rykala, M., Strobel, M., Bollinger, P.K., Ulm, B., Kraft, E., Huber, S. and Lorenz, A., 2021. App-Based Feedback for Rehabilitation Exercise Correction in Patients With Knee or Hip Osteoarthritis: Prospective Cohort Study. *Journal of Medical Internet Research*, *23*(7), p.e26658.

**[5]**  Królak, A., Wiktorski, T., Bjørkavoll-Bergseth, M.F. and Ørn, S., 2020. Artifact correction in short-term hrv during strenuous physical exercise. *Sensors*, *20*(21), p.6372.

**[6]** Velloso, E., Bulling, A., Gellersen, H., Ugulino, W. and Fuks, H., 2013, March. Qualitative activity recognition of weight lifting exercises. In *Proceedings of the 4th Augmented Human International Conference* (pp. 116-123).

**[7]** Chen, S. and Yang, R.R., 2020. Pose Trainer: correcting exercise posture using pose estimation. *arXiv preprint arXiv:2006.11718*.

**[8]**  Novatchkov, H. and Baca, A., 2012. Machine learning methods for the automatic evaluation of exercises on sensor-equipped weight training machines. *Procedia Engineering*, *34*, pp.562-567.

**[9]** Agrawal, Y., Shah, Y. and Sharma, A., 2020, April. Implementation of machine learning technique for identification of yoga poses. In *2020 IEEE 9th international conference on communication systems and network technologies (CSNT)* (pp. 40-43). IEEE.

**[10]** Frangoudes, F., Matsangidou, M., Schiza, E.C., Neokleous, K. and Pattichis, C.S., 2022. Assessing Human Motion During Exercise Using Machine Learning: A Literature Review. *IEEE Access*, *10*, pp.86874-86903.

Dept. of ISE,RNSIT 2021-2022 37