

**DEDAN KIMATHI UNIVERSITY OF TECHOLOGY**

**SCHOOL OF COMPUTER SCIENCE AND IT**

**DEPARTMENT OF COMPUTER SCIENCE**

**BACHELOR OF SCIENCE IN COMPUTER SCIENCE**

**PROJECT FOR THE FINAL YEAR OF STUDY**

**BY**

**NDUNGU KELVIN KARIITHI**

**REG No. C026-01-1334/2019**

**WRONG SITTING POSTURE DETECTION USING MACHINE LEARNING.**

**A project submitted to the Department of Computer Science in the School of Computer Science and Information Technology in partial fulfillment of the requirements for the award of the degree in Bachelor of Science in Computer Science in Dedan Kimathi University of Technology.**

**January , 2023**

# DECLARATION

This project is my original work and has not been presented for a degree in any other University

Name: …………………………………………………………...

……………………….…. …………………

Signature Date

This project has been submitted for examination with my approval as university

**Supervisor**

Name: …………………………………………………………...

……………………… ……………….

Signature Date

# ACKNOWLEDGEMENT

I would like to express my sincere gratitude to all those who have helped me bring my final year project to fruition.

First and foremost, I would like to thank my supervisor, Professor Benson, for his unwavering guidance and support throughout my project. His guidance and expertise were invaluable in helping me stay focused, organized, and motivated to complete this project.

I am also grateful to my family, friends, and colleagues for their encouragement and support throughout this project. I am especially grateful to my parents for their unconditional love and support, and for believing in me.

I also wish to thank the staff at Dedan Kimathi University of Technology, for allowing me access to their library and other resources. I am especially grateful to the librarians for their assistance in locating relevant sources for my project.

Finally, I want to thank my fellow classmates and colleagues for their support and insights. Their comments and feedback were of great help in improving my project.

Once again, thank you all for your support and help in making this project a reality.

# ABSTRACT

Using an image classification algorithm, the system aims to identify the user's sitting posture and provide suggestions on how to better maintain it. The project will provide a new posture-detecting technology. The COVID epidemic has led to a significant rise in the number of people working from home, frequently without the requisite ergonomic gear and infrastructure. Incorrect desk heights, inadequate desktop chairs, and excessive laptop usage all offer serious health risks to users. To make matters worse, poor sitting posture may contribute to a variety of health problems, including but not limited to: back pain, headaches from pinched nerves, and spinal dysfunction. Lower back discomfort is on the rise, yet the number of sufferers has never been observed to decrease. In addition, around 20% of the population is affected by this kind of illness, with the majority of those affected being software developers. Responsible citizens may utilize the information gained here to develop their own plans for addressing the country's back pain crisis.

Solution: The suggested project uses technology based on webcams to identify and detect a person's sitting position in real time. Based on the results, the person is then given a result as to whether the sitting posture is correct or wrong. The application uses computer vision and convolutional neural networks to analyse human postures and immediately notifies the users to recheck their postures through the web app. The data provided as input to the network is divided into two sub-classes, and each posture is then recognized and analysed. For primary training and testing purposes, convolutional neural networks were used as they provide an intuitive way of looking at images, analysing adjacent pixels in small areas and inputting those readings into a pooling layer.

The performance of the application will be evaluated using accuracy, precision, recall, and F1 score metrics. The results of the evaluation will be used to improve the accuracy of the model.

In conclusion, this study will develop a web application that can detect wrong sitting postures using CNNs. The application will be able to detect wrong postures in real-time and provide feedback to help users correct their postures. The results of the evaluation will be used to improve the accuracy of the model.

# **TABLE OF CONTENTS**

[DECLARATION I](#_Toc127797203)

[ACKNOWLEDGEMENT II](#_Toc127797204)

[ABSTRACT III](#_Toc127797205)

[TABLE OF CONTENTS IV](#_Toc127797206)

[LIST OF TABLES VI](#_Toc127797207)

[LIST OF FIGURES VII](#_Toc127797208)

[DEFINITION OF TERMS VIII](#_Toc127797209)

[1. CHAPTER ONE: INTRODUCTION 9](#_Toc127797210)

[1.1. Background Information. 9](#_Toc127797211)

[1.2. Problem statement. 11](#_Toc127797212)

[1.3. Objectives. 12](#_Toc127797213)

[1.3.1. General objective 12](#_Toc127797214)

[1.3.2. Specific objectives. 12](#_Toc127797215)

[1.4. Research questions. 12](#_Toc127797216)

[Justification. 13](#_Toc127797217)

[1.5. Scope. 13](#_Toc127797218)

[1.6. Limitation 13](#_Toc127797219)

[2. CHAPTER TWO: LITERATURE REVIEW. 15](#_Toc127797220)

[2.1. Introduction 15](#_Toc127797221)

[2.2. Case studies. 15](#_Toc127797222)

[2.2.1. Case study 1: Fuzzy Neural network. 15](#_Toc127797223)

[2.2.2. Case study 2: Monocular depth images. 15](#_Toc127797224)

[2.2.3. Case study 3: Image classification using SVM 16](#_Toc127797225)

[2.2.4. Case study 4: Arduino-equipped method. 16](#_Toc127797226)

[2.3. Summary. 17](#_Toc127797227)

[2.4. Research Gap. 17](#_Toc127797228)

[2.5. Proposed methodology 18](#_Toc127797229)

[3. CHAPTER THREE: METHODOLOGY 19](#_Toc127797230)

[3.1. Introduction. 19](#_Toc127797231)

[3.2. Fact Finding Techniques. 19](#_Toc127797232)

[3.3. Datasets. 22](#_Toc127797233)

[3.4. Software development design. 22](#_Toc127797234)

[3.4.1. Characteristics of an Incremental module includes 23](#_Toc127797235)

[3.4.2. Phases of incremental model 23](#_Toc127797236)

[3.4.3. Justification for the method 24](#_Toc127797237)

[3.5. Preliminary Data processing and analysis. 24](#_Toc127797238)

[4. CHAPTER FOUR: ANALYSIS AND DESIGN 26](#_Toc127797239)

[4.1. Introduction 26](#_Toc127797240)

[4.2. Requirement analysis. 26](#_Toc127797241)

[4.2.1. Functional requirements. 26](#_Toc127797242)

[4.2.2. Non-functional requirements. 26](#_Toc127797243)

[4.2.3. Software and Hardware requirements. 26](#_Toc127797244)

[4.3. Data analysis. 27](#_Toc127797245)

[4.4. System Analysis. 30](#_Toc127797246)

[4.4.1. Use case. 30](#_Toc127797247)

[4.4.2. Context diagram 30](#_Toc127797248)

[4.4.3. Flowchart diagram. 31](#_Toc127797249)

[4.4.4. Model architecture 32](#_Toc127797250)

[4.5. System design. 32](#_Toc127797251)

[5. CHAPTER FIVE: TESTING AND IMPLEMENTATION. 36](#_Toc127797252)

[5.1. Introduction. 36](#_Toc127797253)

[5.2. Unit Testing. 36](#_Toc127797254)

[5.2.1. Test case 1: Log in and Sign-Up button. 36](#_Toc127797255)

[5.2.2. Test case 2: Logout button. 36](#_Toc127797256)

[5.2.3. Test case 3: About page. 37](#_Toc127797257)

[5.2.4. Test case 4: statistics page. 37](#_Toc127797258)

[5.2.5. Test case 5: Start camera for laptop. 37](#_Toc127797259)

[5.3. Integration Testing 37](#_Toc127797260)

[5.3.1. Test case 6: Database integration of log in and sign up. 37](#_Toc127797261)

[5.3.2. Test case 7: Database connection with the statistic page. 37](#_Toc127797262)

[5.3.3. Test case 8: Start running the model in the laptop from web application. 38](#_Toc127797263)

[5.4. System Testing. 38](#_Toc127797264)

[5.4.1. Test case 9: Whole system testing. 38](#_Toc127797265)

[5.5. System Deployment and Implementation Procedures. 38](#_Toc127797266)

[5.5.1. Process In the Implementation Plan 38](#_Toc127797267)

[5.5.2. System specification and module required. 39](#_Toc127797268)

[5.5.3. Installation process. 39](#_Toc127797269)

[6. CHAPTER SIX: CONCLUSION AND RECOMMENDATION 40](#_Toc127797270)

[6.1. Discussion 40](#_Toc127797271)

[6.2. Limitations. 40](#_Toc127797272)

[6.3. Recommendation 40](#_Toc127797273)

[6.4. Conclusion 41](#_Toc127797274)

[7. REFERENCE 42](#_Toc127797275)

[8. APPENDICES. 43](#_Toc127797276)

[APPENDICES A: GANNT CHART 43](#_Toc127797277)

[APPENDICES B: BUDGET 43](#_Toc127797278)

[APPENDICES C: SAMPLE CODE 44](#_Toc127797279)

# LIST OF TABLES

[Table 1: Incremental model 23](#_Toc121006574)

[Table 2: Accuracy table 28](#_Toc121006575)

[Table 3: Budget 42](#_Toc121006576)

# LIST OF FIGURES

[Figure 1: pie chart 1 19](#_Toc121006577)

[Figure 2: Pie chart 2 19](#_Toc121006578)

[Figure 3: Pie chart 3 20](#_Toc121006579)

[Figure 4: Pie chart 4 20](#_Toc121006580)

[Figure 5: Pie chart 5 21](#_Toc121006581)

[Figure 6: accuracy plot graph: 28](#_Toc121006582)

[Figure 7: Use case 29](#_Toc121006583)

[Figure 8: Context diagram 29](#_Toc121006584)

[Figure 9: DFD diagram 30](#_Toc121006585)

[Figure 10: Flow chart 1 30](#_Toc121006586)

[Figure 11: Flow chart 2 31](#_Toc121006587)

[Figure 12: CNN 31](#_Toc121006588)

[Figure 13: Dashboard 32](#_Toc121006589)

[Figure 14: Home page 32](#_Toc121006590)

[Figure 15: system running 33](#_Toc121006591)

[Figure 16: Statistics page 33](#_Toc121006592)

[Figure 17: register page 34](#_Toc121006593)

[Figure 18: Sample code model 43](#_Toc121006594)

[Figure 19: Sample code PHP 43](#_Toc121006595)

[Figure 20: Sample code PHP 44](#_Toc121006596)

# DEFINITION OF TERMS

HTTPS - Hypertext Transfer Protocol Secure

OpenCV - OpenCV is a library of programming functions mainly aimed at real-time computer vision.

Keras - is an open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library.

TensorFlow - is a free and open-source software library for machine learning and artificial intelligence

Kaggle - an online community of data scientists and machine learning practitioners. Kaggle allows users to find and publish data sets.

CNN - Convolutional Neural Networks

KNN - kernel network networks

HMM - Hidden Markov Models

ReLU - Rectified Linear Unit.

SVM - Support Vector Machine

Keypoint - image skeleton represents an individual’s orientation in a certain structure. It’s essentially a collection of data points that can be combined to characterize a person’s pose.

# CHAPTER ONE: INTRODUCTION

## Background Information.

According to World health organization, health care is defined as a state of total physical, mental and social well-being and medical care of diseases and illness. In the recent days the Covid pandemic has affect the living style of many people. Some people find themselves working at home and sitting in front of their laptops/computers for long hours throughout the day. Because our day-to-day lives are becoming more dependent on sedentary work, and the opportunities for physical exercise are decreasing (due to the COVID-19 pandemic and the associated restrictions and lockdowns), an increasing number of people are developing a variety of medical conditions that are directly associated with leading sedentary lifestyles. Back discomfort is one of the most common complaints, and poor sitting posture is one of the contributing elements to this issue. Long periods of sitting have been linked to early mortality, chronic disease, metabolic syndrome, and obesity. Moreover, sitting with improper posture may be detrimental to the health of the lumbar region. Poor sitting postures are characterized by an excessive flexion and extension of the lumbar, which causes a significant compression of the lumbar. In instance, back discomfort in infancy induced by incorrect sitting position might be a risk factor for adult lumbar illness. Also, sedentary behavior has a detrimental impact on people's well-being and health, therefore posture recognition is especially important for handicapped individuals (Ma et al., 2020) and the elderly (Chen et al., 2018b). Because both healthy and spine-pain-afflicted adults live in assisted-living facilities, methods to enhance their quality of life are needed.

When it comes to setting up their workspace, employees don't seem to be making advantage of the ergonomic tools and infrastructure that is available to them. Using laptops while slumped over a desk in a terrible posture, sitting at a desk that isn't properly aligned, or using a mismatched desk height are all common workplace problems. Back pain, neck pain, headaches, and even spinal dysfunction may be the outcome. The number of individuals who suffer from back pain is increasing at a steady pace, yet it never goes away. Workers in the software business should be on the lookout for diseases like these, since they impact 20% of the general population. Using an image classification algorithm, the system will identify the user's posture in real time and provide a result to use whether the posture is incorrect or right. The suggested project helps to tackle all of these issues. They may utilize this website to swiftly adjust their posture and lessen their risk of back discomfort while at work. Along with a poor diet, the sedentary and monotonous lifestyle of certain jobs may lead to a number of health issues, which can be complex and difficult to manage.

Research has been performed to create a posture monitoring system that aids individuals in maintaining a proper posture. Using pressure distribution data from sensors mounted to chair components like the seat and backrest is the most common way. Prior to the creation of the posture monitoring system, it is important to precisely categorize the sitting position in real time. The Hidden Markov Models (HMM), Nave Bayes (NB) classifier, and k-nearest neighbor (kNN) classifier have all been used in previous research to categorize sitting postures. A variety of traditional machine-learning techniques such as support vector machines (SVMs), and kNNs (kernel network networks) are still in use to achieve diverse research goals. Deep learning has recently been shown to be effective in a variety of fields of study, including image processing and voice recognition.

A popular type of deep learning method, the convolutional neural network (CNN) is able to take full use of two-dimensional input data. A variety of image recognition fields, such as motor defect detection, vision-based mobile robot navigation, physiological signal analysis in medical evaluation, and human emotion identification, have made extensive use of it. However, as far as we know, the CNN algorithm has never been applied to classify sitting positions. Because webcam data has a two-dimensional picture structure, it's plausible to believe that a CNN algorithm could classify the sitting position based on the image captured by the webcam and the fact that most laptop in the recent have a webcam mounted in them and also, they are so cheap to buy of the market.

The project also gives the user a friendly user interface to interact with. The interface does not only allow user to start the webcam but also provide the statics visualization about their sitting posture using graphs and pie charts. Developed in Python, Flask is a small, lightweight web application framework. It's a microframework since it doesn't need any specific tools or libraries. It does not have a database abstraction layer, form validation, or any other component that relies on third-party libraries for common functionality. As a result, Flask offers extensions that may be used as if they were built directly in the framework. The trained Convolutional neural network model will be hosted online and a user-friendly interface built in Flask server development Framework.

## Problem statement.

There has been a huge rise in the amount of time spent inactive rather than dynamic physical exercise since the emergence of computer technology and the economy. Sitting is 7 times more taxing on the lower back than laying down and 3 times more taxing than standing, and since the improper posture does not distribute weight uniformly, the body is put under additional stress. The more time you spend in a bad posture, the more likely it is that you may have back discomfort. It was found that 46.5% of all cases of domestic scoliosis in 2012 were caused by people sitting in an unsafe position, and in 2014, Steelcase Corporation studied 2,000 employees across 11 countries and developed nine problematic postures to emphasize the significance of sitting correctly.

Although the number of spines specialty hospitals has risen dramatically in recent years, and the percentage of spinal procedures has grown in tandem, therapy via these surgeries has demonstrated difficulties in treating the underlying causes of back pain. From January 2000 to December 2005, 894,594 elementary and secondary school kids in Seoul, Gyeonggi Province, were screened for spinal illness. 1.66 percent in 2001; 2.41 percent in 2002; 1.35 percent in 2003; 2.49 percent in 2004; and 3.53 percent in 2005. Currently, the issue is 3.08 percent. Correcting posture and adopting healthy habits, rather than direct therapy, is adequate to ease and improve until the curvature of the spine is severe. For this reason, there have been great need to develop system that aim at detecting a wrong posture and warning a person to try and correct the sitting posture.

Despite the fact that there may be established classical algorithms for predicting human posture, they often assume that the full human skeleton is visible in the frame. With everyone now working from home, it's impossible to meet together and discuss these scene compositional considerations. Body posture tracking with many cameras isn't feasible for everyone. Therefore, a solution that can warn the user (or their care provider) of improper posture using inexpensive consumer sensors such as laptop webcam, is needed to enhance the user's well-being without constant monitoring is needed. Using black box models, we can tackle many complicated tasks by transferring the bulk of computations away from the end-user device and into the training stage of machine learning, which has revolutionized computer vision tasks.

Sensors placed on a chair (pressure) or even on the user's body (gyroscope) are often used in most systems (more towards IOT). After that, the data is entered into a system and matched against a rule basis. However, this system is not scalable, difficult to set up, and prohibitively expensive to apply. Consequently, in the developing of the system, we need to solve the three shortcomings of prior systems.

## Objectives.

### General objective

1. To develop a web application that can detect wrong sitting postures using machine learning techniques.

### Specific objectives.

1. To collect and pre-process the dataset of sitting postures with labels indicating correct or wrong sitting postures.
2. To design and train a convolutional neural network model to classify sitting postures in real-time.
3. To develop a web application that can capture video input from a webcam and detect sitting postures in real-time.
4. To evaluate the performance of the developed model using accuracy, loss metrics and plotting the learning curves.
5. To record the number of correct or wrong sitting postures and visualize the data in the web application.

## Research questions.

1. What architecture and hyperparameters will be used to develop and train a convolutional neural network model for real-time classification of sitting postures?
2. How can a web application that can gather video input from a webcam and recognize seating postures in real time be created?
3. How can the accuracy, precision, recall, and F1 score metrics be used to evaluate the performance of the built application?
4. In the online application, how can the number of proper or incorrect sitting postures be recorded and visualized?

## Justification.

The following are the primary areas this project can be applied in: good posture at home helps avoids a variety of injuries and miscarriages, including those caused by watching TV, cooking, or cleaning; it also decreases the chance of miscarriage during pregnancy. It is also possible to avoid the strain on the shoulders and back that comes with playing indoor games like chess and carom by keeping proper body positions. Exercises that use heavy weights may help avoid long-term spine and cartilage issues. When doing a dead lift, for example, in an inappropriate position that causes spinal curvature injuries, for example, many outdoor activities require maintaining a strong back, including canoeing, kayaking, golf, and archery. In the office, this might be a Smart Campus Initiative that monitors IT staff sitting posture when utilizing systems. Also, individuals tend to adopt an improper stance while presenting presentations, team meetings, and formal management. Proper sitting posture of the body's parts means that the least amount of energy is needed to sustain a particular posture, which is what we mean by "ideal" posture. Improved circulation of physiological fluids may be achieved by proper posture. Simple posture monitoring systems, such as those discussed below, may be used to track posture and adjust it over time.

## Scope.

The system will contain a trained model deployed in a web application. The web application will include functionalities that will allow user to log in and out of the system, start the model to detect the sitting posture and also help user with a statistic of the sitting postures. The model will include algorithm to classify a wrong sitting posture, the ability to start the webcam and also a text indicating of the video a stream whether it’s a good posture or a wrong sitting posture. However, the project does not include the recommendation of a propriate sitting posture.

## Limitation

One of the most significant limitations I faced while developing my project was the need for a large, comprehensive dataset. The dataset I had available was limited in size, scope, and variety. This made it difficult to accurately and reliably detect bad sitting postures in images, as the machine learning model had limited exposure to images depicting bad postures.

Another limitation was the lack of resources available to develop the web application. I was limited in terms of the programming language and technologies I could use, as well as the time and budget available for development. This made it challenging to develop a user-friendly, intuitive web application that could effectively deploy my machine-learning model.

Finally, the lack of a robust computing infrastructure was a limitation. Machine learning models require significant computational resources for training and prediction, and my project was limited in terms of the hardware and cloud computing options available. This made it challenging to develop and deploy a machine-learning model with high accuracy.

# CHAPTER TWO: LITERATURE REVIEW.

## Introduction

Human posture has been the subject of several studies in recent years. There are many types of research done and techniques proposed in the field of detection using machine learning. Each technique has its pros, cons and variance in the expected result. Understanding the available options would be vital to achieving the targeted goal. A few essential components that are necessary for this project include methods of detecting posture, image pre-processing algorithms and commonly used training techniques for multi-label classification data. These are discussed below.

## Case studies.

### Case study 1: Fuzzy Neural network.

One methodology used for classification of human body postures was based on a neural fuzzy network which was proposed in [2]. Standing, sitting, bending, and reclining were the four most common body positions utilized to categorize human postures in this article. Another project [6] used the picture to identify the human body for the first time. Then, the body was segmented. Classification was then applied to the characteristics that had been removed from the silhouette against a light backdrop. In addition to the silhouette's length-to-width ratio, other traits included large magnitudes of notable Fourier transform coefficients. The use of different colored markers for posture detection has been described before, however as indicated in [7], several Marker-less techniques have also been found. The 3-D human poses were found from a large amount of data by identifying the body joints and distinct parameter sequences of the postures. It was possible to extract a wide range of information from 3-D representations of human bodies, such as the location and orientation of points, the height of a person, the variable lengths of different bones as well as numerous movement aspects. A Hidden Markov Simulate (HMM) was used to model movement characteristics and an exemplar-based HMM was used for configuration features.

### Case study 2: Monocular depth images.

A highly effective filtering system was also developed in [8] that employed depth photos, in this case monocular depth images, to follow human positions. A highly productive and discriminative model were integrated. A model was subsequently created, which helped to distinguish the many components of the human body. The datasets were used to get this evidence. The model-based local searches were improved with each iteration until the desired kinematic chain was achieved. On the MPEG-7 dataset, we've also been able to recognize human body position. For shape matching, they employed a variety of approaches and procedures, all of which were based on the specifications of MPEG-7. The project employed sophisticated projections [10]. First, the 3-D depth photos acquired by a depth Camera sensor or a Kinect sensor may be used to recognize the human body, and then the various human postures can be identified.[9].

### Case study 3: Image classification using SVM

Categorizing the data is accomplished via the usage of a Support Vector Machine. In order to classify the data, SVM employs supervised learning techniques, which are used for data analysis purposes. Images or videos acquired by cameras may be classified using SVM. Fuzzy logics have been shown to be an effective tool for analysing posture recognition, according to many research. Fuzzy logics may use a wide range of inputs, including horizontal and vertical projections, height and width ratios, and a wide variety of points to make their decisions. Among the topics covered were SVM optimization, theoretical confluence, classification into multiple classes, probabilistic estimation, and parameter selection [11]. Human posture identification is aided by a multi-scale morphological approach [18]. To increase the robustness and accuracy of the pose identification, it makes advantage of morphological scale space's different-areas-based form similarity. In [17], techniques for determining human body position using static photographs are discussed. Artificial Neural Networks are used to make this estimate. Functionalities and structural information collected from genuine neural networks are the focus of the Artificial Neural Network (ANN). An initial analysis is performed on the picture, employing a silhouette image to provide data as an input to a neural network, which then generates output based on symbolic representations of various elements of the human body. The body's two-dimensional coordinates may be found in all of these components. ' Next, the kalman filtering procedure is used to improve and optimize the acquired findings. Personal computers, laptops, and other portable devices may all be used to perform the aforementioned approach, which can also be used in real time for estimates. Experiment findings show that it is more practical and effective to estimate human body posture utilizing the above-described approaches than the previously discussed methods.

### Case study 4: Arduino-equipped method.

In the paper [3,] The author demonstrated a system that consists of an Arduino-equipped chair, an Android application that can measure the user's posture, and a server that can assume the position of the sitting user and enable them to monitor their own posture. As a starting point, two sensors were employed to monitor the pressure and ultrasonic waves in order to assess the user's posture by taking into consideration the entire body. Predicting user posture was made easier by using deep convolutional neural networks and a lower-balanced check network. Human sitting postures are given in a new and diverse dataset. It is also unique in that it assesses the whole body's balance rather than simply a specific area of the body. Last but not least, customers will get instructions on how to correct their sitting position through the system's comments, which apply the keyword matching algorithm. In addition, the software may send notifications to the user's phone and recommend films depending on the user's assessed posture. The method's goal is to increase workplace productivity while also benefiting the health of modern workers, many of whom spend the bulk of their waking hours sitting.

## Summary.

In all the above cases, it is clear that most of the method used employ the use of sensor to detect wrong posture and pressure method. Most of the method used applied the static data method to detect the sitting posture and did not employ the use of real time methods to detect the sitting postures. Additionally, from the related work one can argue that the algorithm used in the classification of the sitting posture are outdated and with the present computing power better methods and algorithms can be used to increase the accuracy of the detection.

Moreover, the method also assumes the use of full body detection posture and multiple frames for the detection of the sitting posture which can be a disadvantage as most be cannot afford the multi frame camera equipment in their work places and in their homes. The paper also proposed a system that identifies the sitting posture using 3-D frames which does will take a long time for the model designed to process input, others suggested the use of sitting pressure as the input data to the model to detect the sitting posture. This model would not seem to work out because the use of the pressure would differ with the content the user has in his/her pockets and the weight of the clothes.

## Research Gap.

From the literature review done, there is an area where its exploration is minimal and not much have been in the area of detecting sitting posture from a video stream using computer vision. Much of the project discussed above utilized the use of IoT device which can be costly to get from the market such as a chair with specialized sensor. In this proposed project the use of webcam from the those readily available in today’s most laptops, the view stream will be used to detect wrong sitting posture. Also, most of the paper just suggest the model to be used but did not provide an interface that can enable use interact with the model and the correct statics of their sitting postures.

## Proposed methodology

The overall work flow of the project is designed in such a manner that whenever a person is in state of wrong posture, the developed model will immediately inform the user that he/she is in the wrong postural state and he/she needs to change it, also it will inform what will be its adverse effect on that person. For carrying out the entire process we are taking the help of OpenCV and convolutional neural network to obtain the maximum classification accuracy using CNN algorithm. The proposed system has the following process on the dataset and the architecture, cut all images to a size of (60, 60) pixels, of which the edge parts are filled with black pixels. Label the images of correct sitting postures with “0” and those of incorrect sitting postures with “1”. Apply the data enhancement to expand the dataset. Due to the limited experimental conditions, the dataset constructed is not big enough. So, the data enhancement is necessary for expanding the dataset used for training. Normalize the image so that pixels are within the range of 0 to 1.

CNN model is composed of 19 layers, the input layer is for inputting the processed images with (64, 64) pixels. The output layer is the Classification layer, using the SoftMax Classifier. The Activation layer applies ReLU function, and the Pooling layer has a stride of (2, 2). The SGD + momentum Optimizer is used for training. The categorical\_crossentropy is adopted as the loss function, so, it is necessary to vectorize the classification labels with One-Hot Encoding based on the number of classifications. There are only two categories so the labels are turned to 2-dimensional. The range of random rotation of images during the data enhancement is set to 20. The range of horizontal shift of images is set to 0.2 (the ratio of width before and after the shift) and images can do horizontal reversal randomly. The dataset is categorized as training set, verification set and testing set, which takes up 80%, 10% and 10% respectively.

# CHAPTER THREE: METHODOLOGY

## Introduction.

Methodology is a set of systems, methods, procedures or rules that is used to conduct a structured research process for a thesis or dissertation. The main purpose of the methodology is to be able to come up with an original and significant outcome that will contribute towards the field of your research. In this chapter we will explain the method we use to gather information about the existing systems. This chapter will also mention every component involved in conducting this research from population, population frame, sampling techniques used for the interview and also the software development procedures used. Finally, this chapter provides a detailed explanation of the selected mode of analysis used and data collection method.

## Fact Finding Techniques.

Fact finding is the process of collection of data and information based on techniques which contain sampling existing documents, research, observation, questionnaires, interviews, prototyping and joint requirements planning. Generally, there are many ways of gathering facts: qualitative and quantitative methods in our research, qualitative (expressed in words), secondary and descriptive data were gathered through research and Site visits. During the fact finding for the project, we visited sites that had the research done on a sample of a population and the right description of the research and the population was provided.

In the year 2018 a company by the name, roost conducted a study of the current state of office ergonomic on a company. The final sample size of this survey was 412 (135 men and 275 women). Since two participants preferred not to identify their gender, the two cases were excluded from analyses related to gender. Poor office posture cost small businesses a hundred of a thousand of dollars each year from the data. The online questionnaire was composed of 22 items in 3 categories: 1) demographic information, 2) the discomfort in their sitting hadit, and 3) current neck and shoulder symptoms and symptoms during use which include duration, types, severity, and location of symptoms.

86% of the office workers reported some discomfort from their office furniture and equipment. Most symptoms discomfort was reported in the neck (84.6%), upper back/shoulder (65.4%), arms/hands (33.6%), and head (15.0%). The most common types of symptoms are stiffness (74.3%), soreness (48.1%) and aching or pain (42.5%). The most severe level of discomfort during tablet use was moderate (55.4%), but noticeably, 10.0% of participants reported severe symptoms (i.e. 7–10 on a 10-point Visual Analog Scale). Only 46.1% of the respondents reported that they would stop their current sitting habit when experiencing discomforts in the office. 15% of the users reported that the symptoms affect their sleep.

Figure 1: pie chart 1

From the data also it showed that 37% of women who sit more than 6 hours per day are more likely to die prematurely than those who sit 3 hours a day.

Figure 2: Pie chart 2

The research presented from the leave register shows that 33% of the days away from work are spent om musculoskeletal injuries. Which interpreted that many of the workers sitting posture at the office was bad.

Figure 3: Pie chart 3

From the research it shows that 98% of the workers showed signs of better posture when they had constant reminders about their posture while seated a desk, while 87% of the office workers felt more confident and energized while using the standing desk. Also 66% of the works believe that their posture would improve if they had application that detect their posture and remind them their good sitting posture.

The study on the office cost incurred showed that 33% of workers compensation cost were spent on ergonomic injuries which translate to $29,000 - $32,000 of money used. The cost can be cut down by employing one of the ergonomic solutions below.

Figure 4: Pie chart 4

Out of the total work injuries reported in their work place 33% of them were caused by posture related injuries as show in the pie chart below. From the data presented above this shows that bad sitting posture is a real and costly problem not only in the offices but also at home when watching.

33% of work injuries are posture-related

Figure 5: Pie chart 5

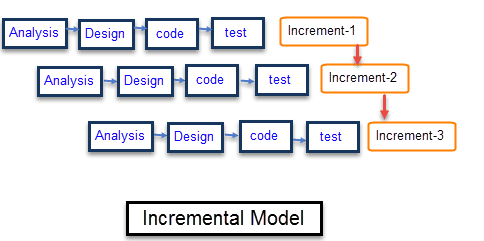
## Datasets.

Since the project employs the deep learning method by using convolutional neural network in OpenPose algorithm, we need good data to train the model in the detection of the body key points movement and angles. For the purpose of the training of the model we will use the data set called COCO, Common Objects in Context dataset. Existing human pose datasets contain limited body part types. The MPII dataset annotates ankles, knees, hips, shoulders, elbows, wrists, necks, torsos, and head tops, while COCO also includes some facial keypoints. For both of these datasets, foot annotations are limited to ankle position only. However, graphics applications such as avatar retargeting or 3D human shape reconstruction require foot keypoints such as big toe and heel. Without foot information, these approaches suffer from problems such as the candy wrapper effect, floor penetration, and foot skate. To address these issues, a small subset of foot instances out of the COCO dataset is labeled using the Click worker platform. It is split up with 14K annotations from the COCO training set and 545 from the validation set. A total of 6foot keypoints are labeled. We consider the 3D coordinate of the foot keypoints rather than the surface position. For instance, for the exact toe positions, we label the area between the connection of the nail and skin, and also take depth into consideration by labeling the center of the toe rather than the surface. The dataset was pretrained in Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields and its runtime from different algorithm was plotted as show in the figure below.

## Software development design.

Software design is a process to transform user requirements into some suitable form, which helps the programmer in software coding and implementation. It is the first step in SDLC (Software Design Life Cycle), which moves the concentration from problem domain to solution domain. In our system we decided to use the incremental model. Incremental Model is a process of software development where requirements are broken down into multiple standalone modules of software development cycle. Incremental development is done in steps from analysis design, implementation, testing/verification, maintenance. Each iteration passes through the requirements, design, coding and testing phases. And each subsequent release of the system adds function to the previous release until all designed functionality has been implemented.

Table 1: Incremental model



**Incremental Phases Activities performed in incremental phases**

Requirement Analysis • Requirement and specification of the software were collected.

Design • Some high-end function is designed during this stage for the application.

Code • Coding of software is done during this stage

Test • Once the system is deployed, it goes through the testing phase

### **Characteristics of an Incremental module includes**

* System development is broken down into many mini development projects
* Partial systems are successively built to produce a final total system
* Highest priority requirement is tackled first
* Once the requirement is developed, requirement for that increment is froze.

### Phases of incremental model

Requirement analysis- Requirement and specification of the software are collected

1. Design- Some high-end functions are designed during this stage.
2. Code- Coding of software is done during this stage
3. Test- Once the system is deployed, it goes through the testing phase

**Advantages of incremental model**

1. The software will be generated quickly during the software life cycles
2. It is flexible and less expensive to change requirements and scope
3. Thought the development stages changes can be done
4. This model is less costly compared to others
5. A customer can respond to each building
6. Errors are easy to be identified.

### Justification for the method

Incremental methodology will be the appropriate methodology because with it I will generate working software quickly and easily during the software life cycle as the modules will be taken to users and in case of any error it will be checked on immediately. This methodology allowed system test at various stages of increments to make sure that it meets the requirements up to that far it had been developed. The advantage of this method is that;

There is exploitation of knowledge gained in development of early increments as later increments are being developed.

1. Allows system test at various stages of increments to make sure that it meets the requirements up to that far it had been developed.
2. Testing is done after every stage hence identification of bugs at each stage is easy.
3. Ability to monitor the effect of incremental changes, isolate issues and adjust before users change over to the proposed system due to gradual implementation.

## Preliminary Data processing and analysis.

The study revealed that females and individuals with current musculoskeletal symptoms were more likely to be at risk for neck and shoulder symptoms. Certain postures during use were also identified as important risk factors, specifically sitting without back support and with the tablet and laptops in lap were significantly associated with symptoms during use. Results of the current study showed that females were significantly more likely to have symptoms during use, especially in the neck and upper shoulder regions. This finding was consistent with the current literature on the prevalence of neck and shoulder dysfunctions.

For individuals performing computer related activities, a previous study suggested that prolonged sitting and having the neck in forward flexion were risk predictors for neck pain29). Neck flexion postures can lead to an increase in gravitational load moment on the cervical spine, which increases cervical extensor muscle activity and may cause muscle strain of the neck extensors if such posture is prolonged. The study revealed that the odds of having symptoms during use for individuals who sit in a chair without back support, which may be related to a “slump” position, are 2.231 times greater than the odds of those sitting with back support.

The current study showed that there were apparent musculoskeletal health implications related to the bad sitting posture while using laptops and device in offices. Female gender, existing neck and shoulder symptoms, and sitting without back support during tablet use were shown as the most important risk factors.

# CHAPTER FOUR: ANALYSIS AND DESIGN

## Introduction

System analysis is the process of investigating an existing system finding its problem and recommending improvements. System Design is the process of defining the architecture, modules, interfaces and data for a system to satisfy specified requirements. System analysis and design involves identifying the system requirements which are either functional or non-functional requirements. Also, the architectural design is discussed here and its diagrammatic design is drowning. Also, the conceptual Data Modelling of the database is discussed.

## Requirement analysis.

System requirements are the required specifications a device must have in order to use certain hardware or software. There are two types of system requirements; functional and non- functional requirements.

### Functional requirements.

1. The system must allow user to log through their website.
2. The system must allow user to start the camera for the camera device attached.
3. The system should show the user history of the sitting posture, good or bad.
4. The system must indicate on the screen to the user whether the current sitting posture is good or bad.
5. The system must be able to visualize the sitting posture of the user in a pie chart or graph.
6. The system should allow new users to register their profile.

### Non-functional requirements.

1. The system Cross-platform Frontend Tool because it was developed using ionic.
2. The system should produce accurate prediction on the posture.
3. The system interface is easy to use.
4. Each page must load within 2 seconds
5. The system meets Web Content Accessibility Guidelines [WCAG 2.1](https://www.w3.org/TR/WCAG21/).

### Software and Hardware requirements.

**Hardware:**

Camera: Webcam camera

Processor:

Ram: 8GB ram or higher

Hard drive: 20GB or higher

CPU: core i5 or higher

**Software:**

Programming language: python3

Algorithm: OpenPose algorithm

Server: Xammp server

Frontend: JavaScripts and HTML

## Data analysis.

The model used a ratio of 60:40. 60% of the dataset was used for training, 40% for testing. Accuracy refers to the percentage of precisely ﬁtting across the entire case. The precision refers to the accuracy of the detection. That is, it indicates how many actual objects are included among the detection results. The recall means detection rate which refers to how well the target objects are detected without being missed. The F1 score harmonic mean of Precision and Recall. The confusion matrix and equations of each value were described in Table 1.

Table 1 Confusion matrix

|  |  |  |
| --- | --- | --- |
| Predicted Class | Actual Class | |
| Positive | Negative |
| Positive | True Positive (TP) | False Positive (FP) |
| Negative | False Negative (FN) | True Negative (TN) |

Precision = TP/(TP+FP)

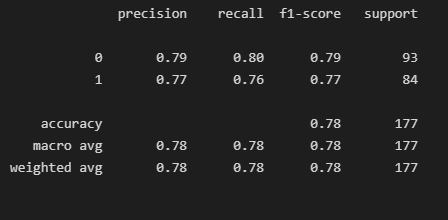
Recall = TP/(TP+FN)

Accuracy = (TP+TN)/(TP+TN+FP+FN)

F1 score = 2 × (precision ×recall)/ (precision + recall)

The accuracy represents the proportion of true predictions generated by the model, while the loss indicates the model's performance in reducing the gap between projected and actual outputs. For both the training and validation datasets, both measures are calculated.

Table 2: Accuracy table



From the above figure in the model, it shows the accuracy, precision, recall and f1-score which were out measurement metrics for the model. The model accuracy is 78% while the precision call is 79% and 77% respectively for the classes 0 and 1. The recall is at 80% and 76% respectively for the classes.

The training accuracy graph depicts the evolution of the model's accuracy during training. The graph depicts the model's ability to properly learn and identify patterns from training data. The accuracy of the model should rise as the number of training epochs increases, as shown by the training accuracy graph.

The validation accuracy graph depicts the evolution of the model's correctness during the validation procedure. The graph illustrates the model's capacity to generalize and categorize patterns in fresh data. The validation accuracy graph should demonstrate an improvement in model accuracy proportional to the number of training epochs.

The training loss graph depicts the evolution of the model's loss over time throughout the training procedure. The graph depicts how successfully the model minimizes the discrepancy between expected and actual training data outputs. The training loss graph should demonstrate a reduction in model loss as the number of training epochs grows.

The validation loss graph illustrates the evolution of the model's loss over time throughout the validation procedure. The graph depicts the degree to which the model minimizes the disparity between expected and actual outputs with fresh data. As the number of training epochs rises, the validation loss graph should demonstrate a corresponding reduction in model loss.

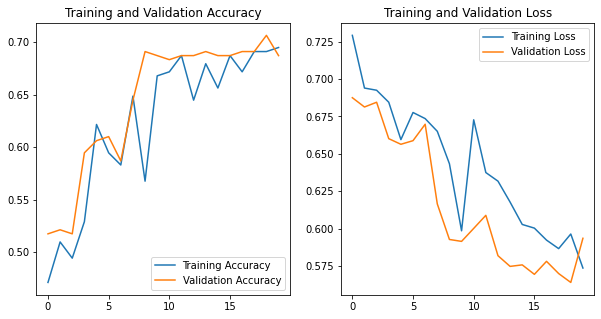


Figure 6: accuracy/loss plot graph:

## System Analysis.

### Use case.

The diagram describes how the system interacts with its users. They model the functionality of a system using actors and use cases.

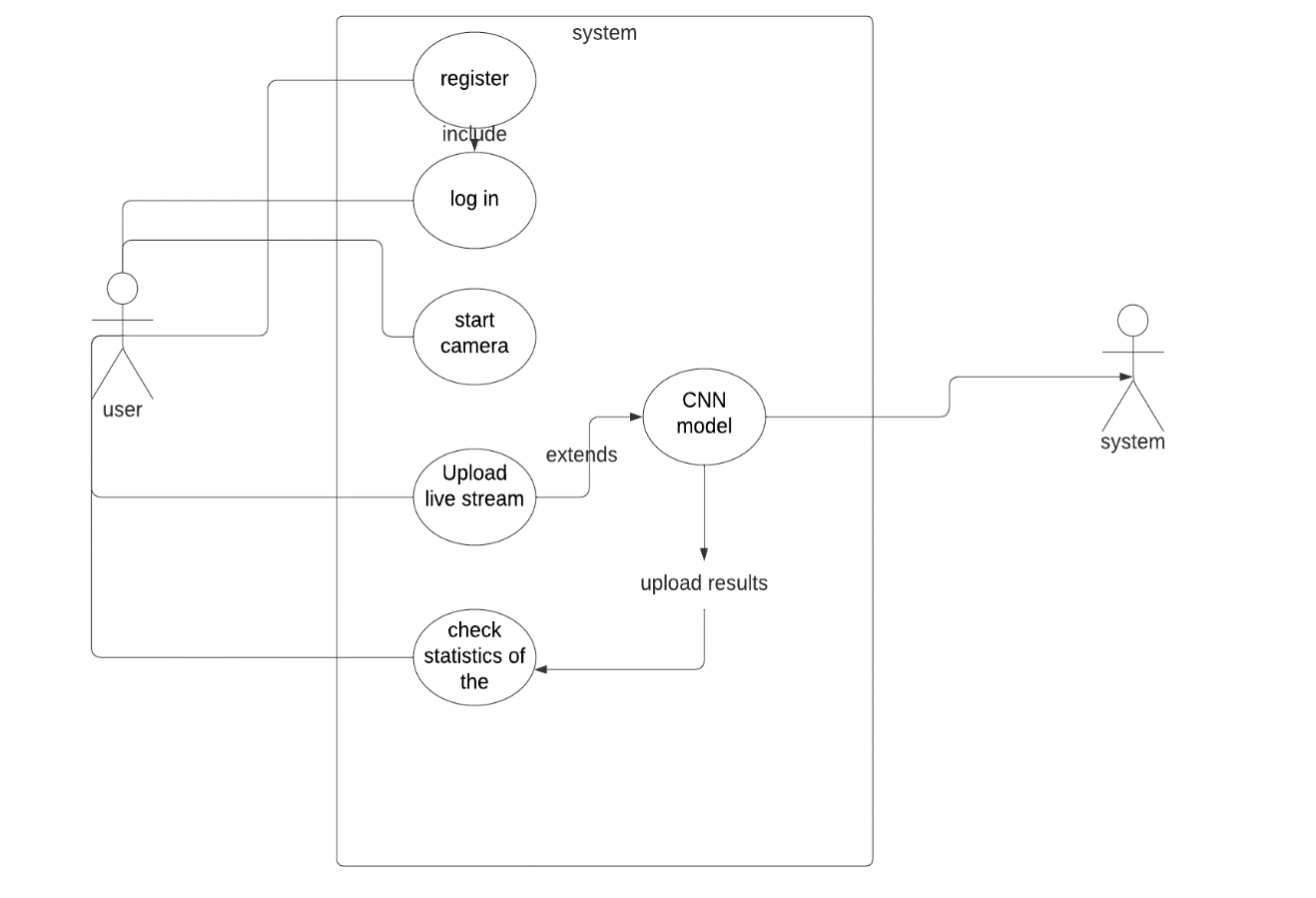


Figure 7: Use case

### Context diagram

The diagram describes all the components that may interact with the system, hence displays the entire software system as a unit.

The Level-0 DFD or context diagram of the system is show below.

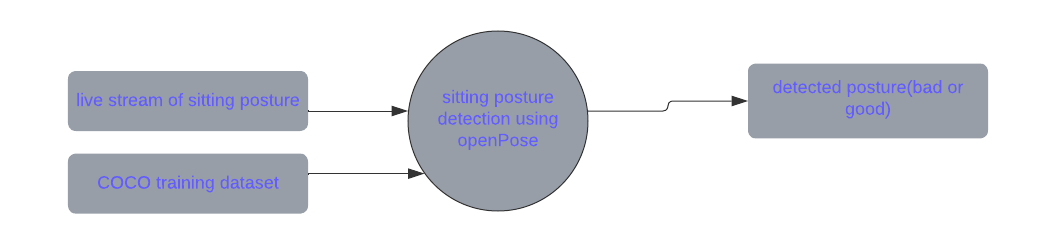


Figure 8: Context diagram

The Level-1 DFD of the system.

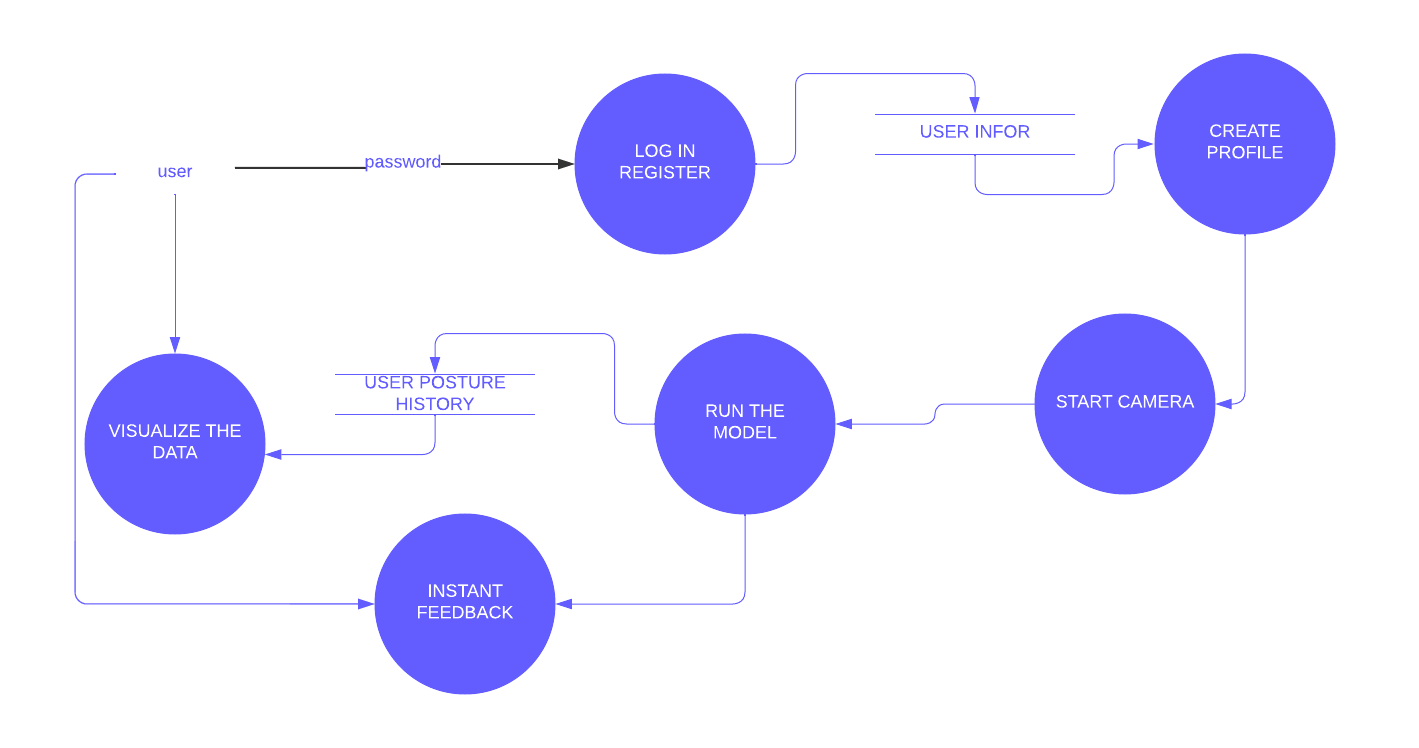


Figure 9: DFD diagram

### Flowchart diagram.

1. **Frontend flowchart of the system.**

With computer vision help, the camera will capture users’ body features, then passed to a deep learning AI model to locate your joints and determine your body posture. Our backend will analyze the output posture, and provide sitting instruction based on the healthy posture standard given by medical research papers

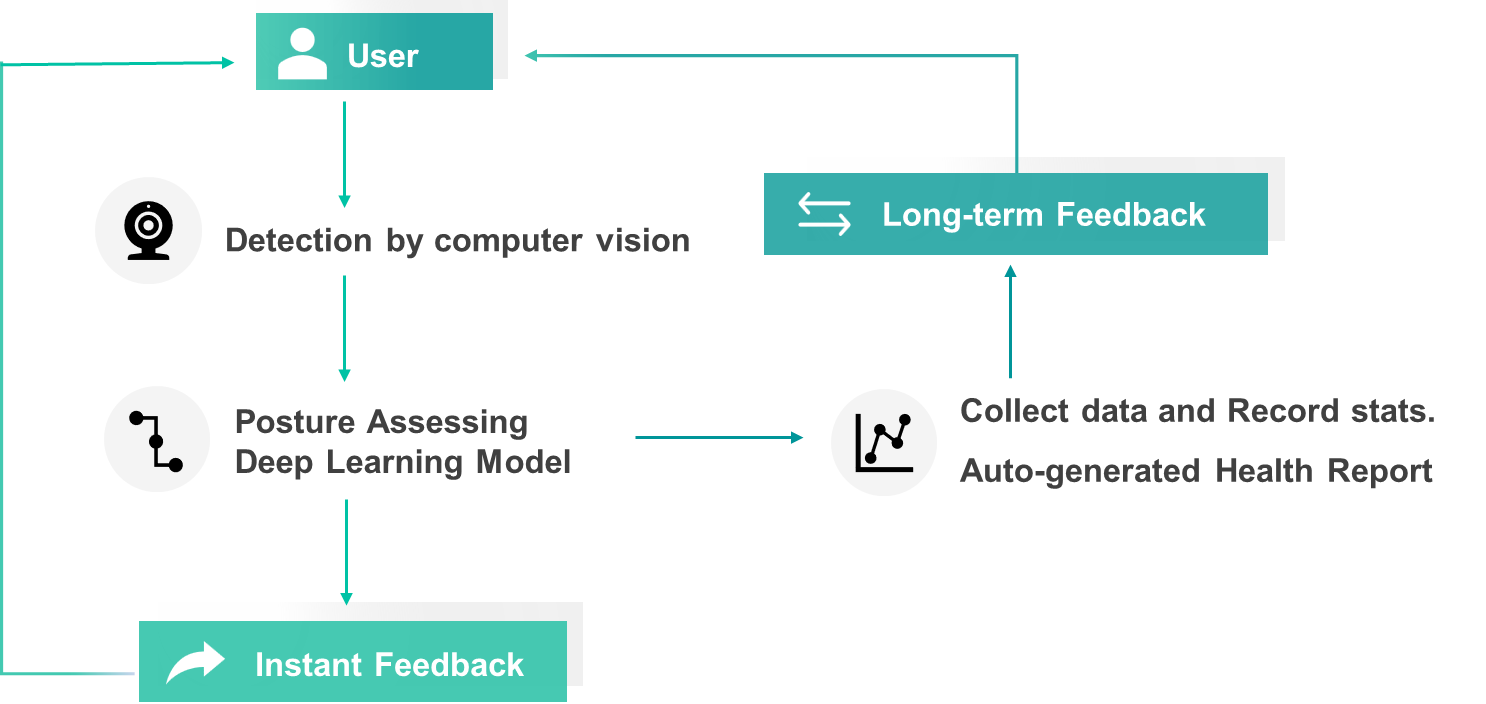


Figure 10: Flow chart 1

1. **Backend processing flowchart diagram**

The camera provides live stream of video to the model which produce the keypoint of the body and then a correct description of a good posture is matched with the output of the CNN algorithm and instant feedback is given to the user, the output is used to train the model further using federated learning.

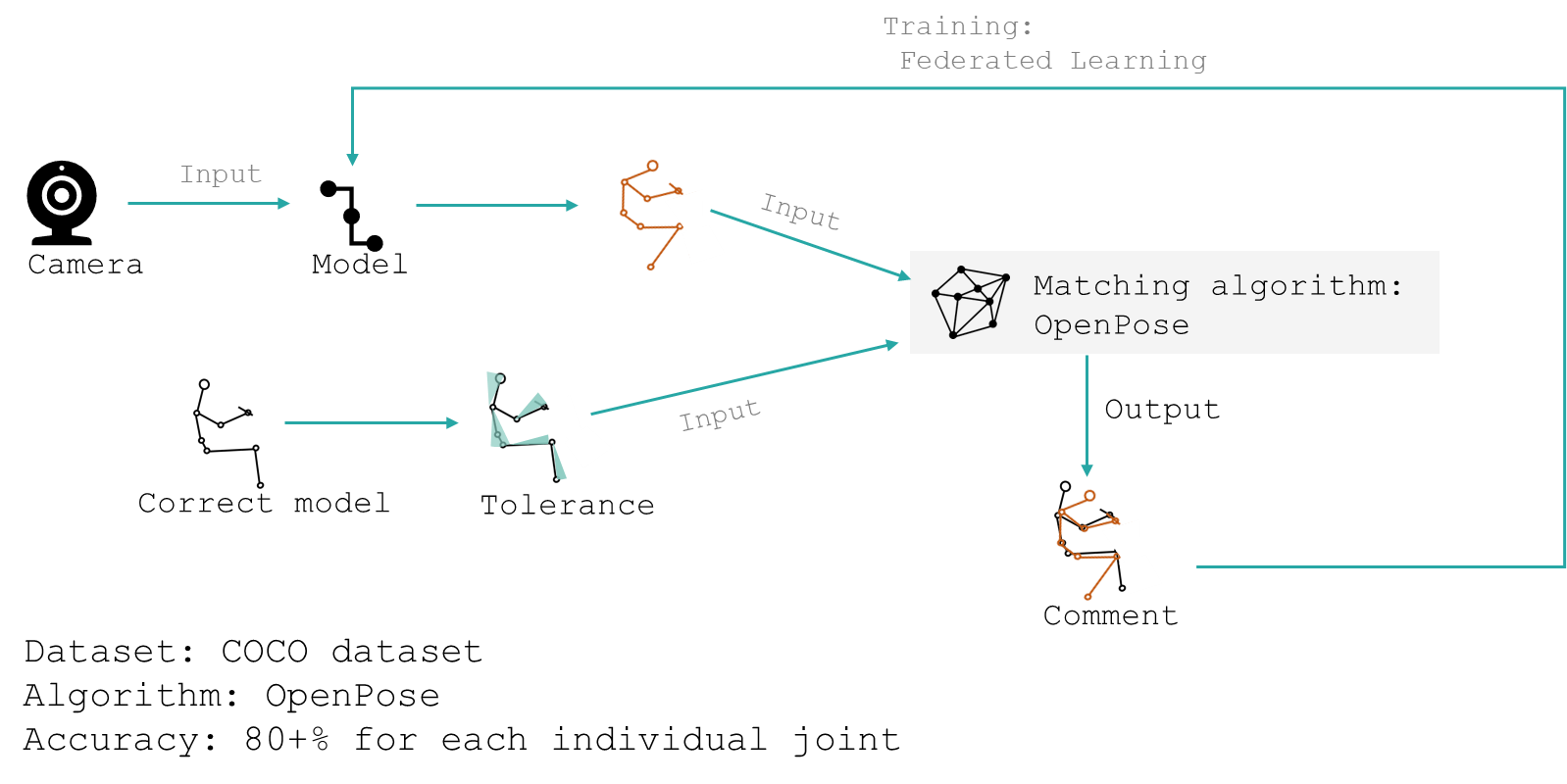


Figure 11: Flow chart 2

### Model architecture

First, a feed-forward network simultaneously predicts a set of 2D confidence maps (S) of body part locations (key points annotations from (dataset/COCO/annotations/) and a set of 2D vector fields of part affinities (L), which encode the degree of association between parts. After each stage, the two branches’ predictions, along with the image features, are concatenated for the next stage. Finally, the confidence maps and the affinity fields are parsed by greedy inference to output the 2D key points for all people in the image.

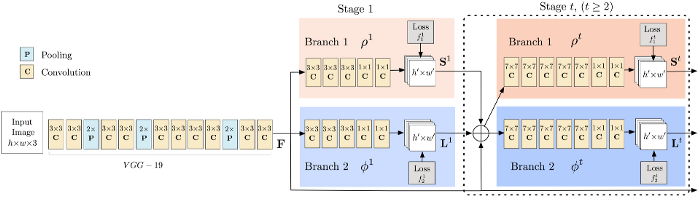


Figure 12: CNN

## System design.

The system frontend is developed using ionic framework that help run the application with a phone in the laptop and give you a good interface to interact with phone like interfaces.

**Dashboard.**

The image below is the model design of the system interface with the login and register buttons.

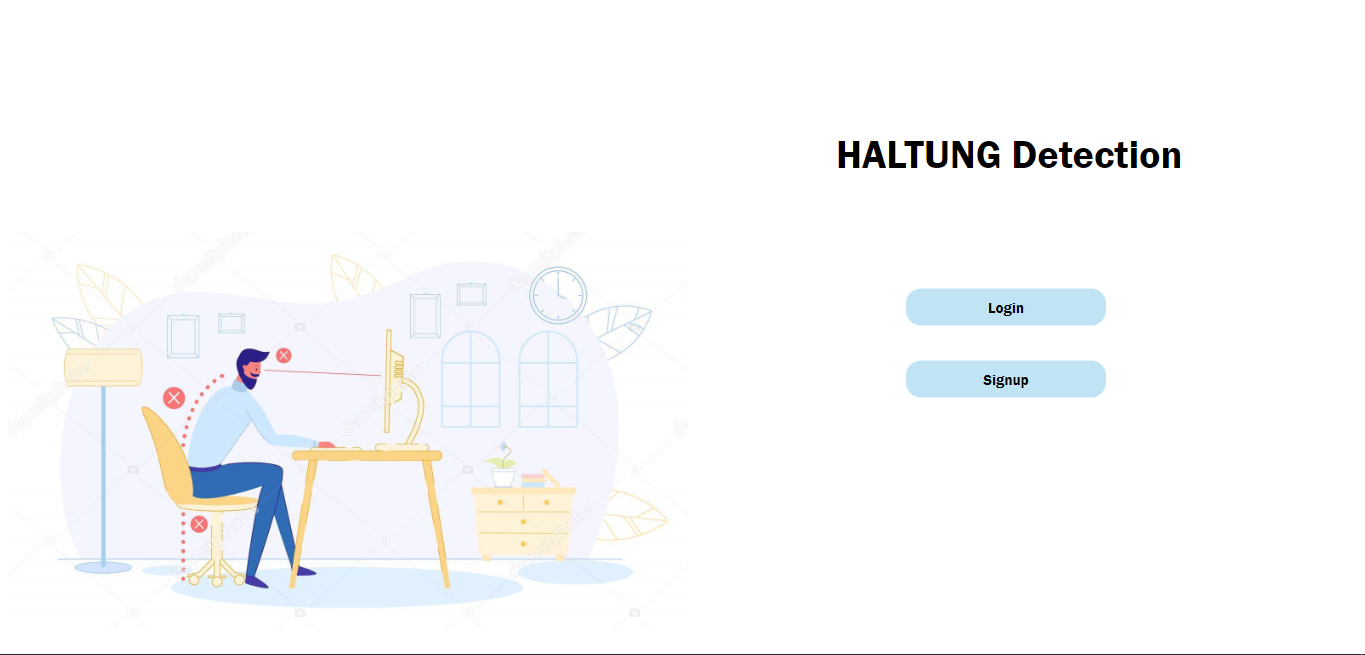


Figure 13: Dashboard

**Home page.**

The page below is the home page of the system. It has the start camera button that will invoke the model and the webcam camera.



Figure 14: Home page

**Running model live stream**

The image below shows the page that is displayed when the model is running in the system server.

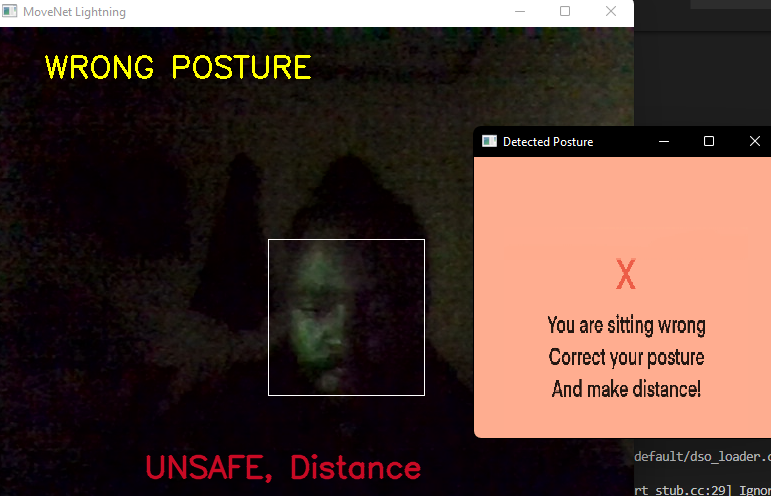


Figure 15: system running

**Statistics page.**

The page that shows the statistics displays weekly and daily data in pie chart and bar graph.

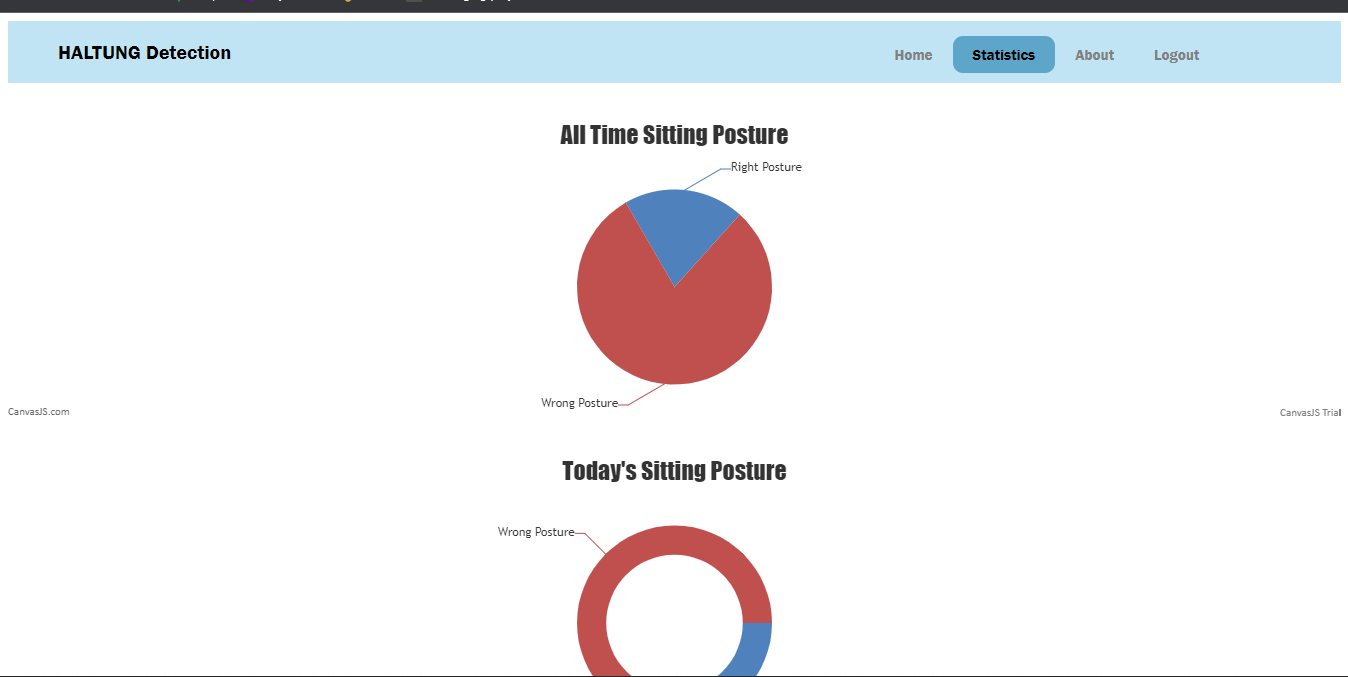


Figure 16: Statistics page

**Profile page**

The below page will display user profile entered during the registration in the system.

**Register page**

The register page asks for user input on email, username, a password and the user submit it where they are directed to the log in page.

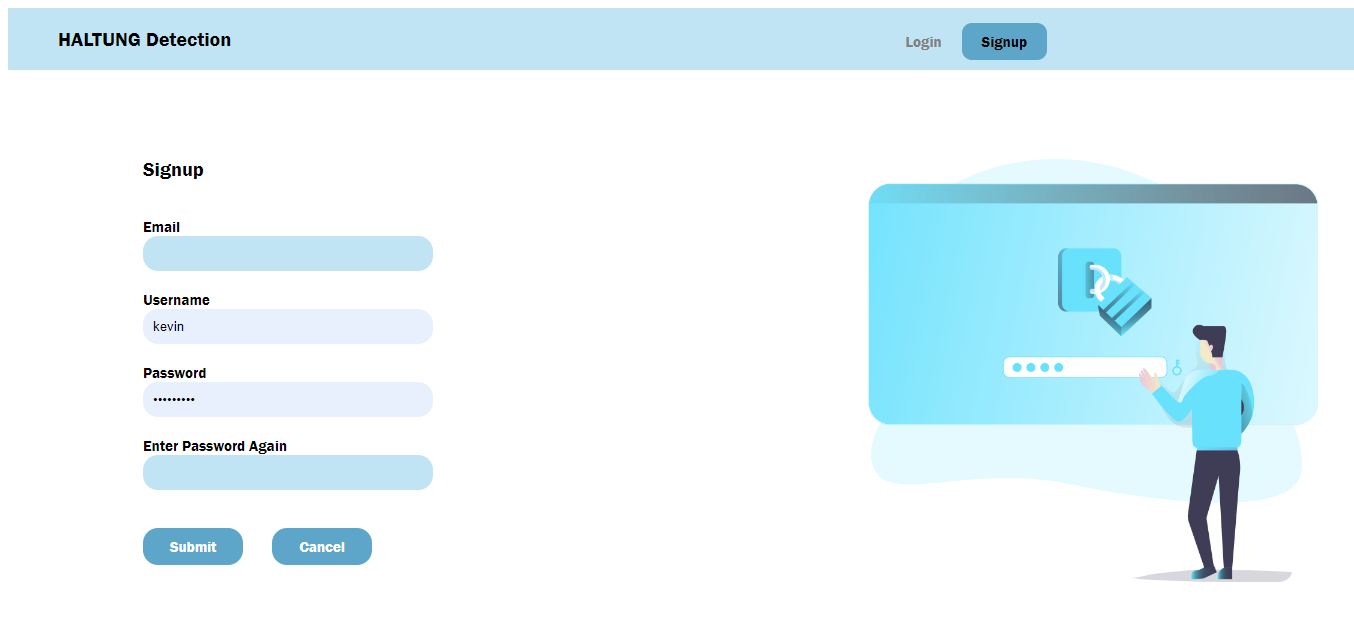


Figure 17: register page

# CHAPTER FIVE: TESTING AND IMPLEMENTATION.

## Introduction.

The process of ensuring that a system satisfies its functional, performance, and reliability criteria is known as system testing. It is usually done in a controlled environment where the system and its components are tested separately from the rest of the system. System testing is often conducted towards the conclusion of the development lifecycle, after the completion of integration and unit testing. System testing may include the use of specialized test equipment, software, and environments, and it often requires many test scenarios.

The steps of taking a tested system and delivering it into production are known as implementation and deployment. This procedure includes installing and configuring the system, as well as verifying that all essential resources are accessible and certifying that the system is ready for use. It also entails monitoring the system after it is operational, providing maintenance and updates, and diagnosing any problems that may emerge.

## Unit Testing.

Unit testing is a software development technique that tests individual units of source code, such as functions, classes, and modules, to see whether they are ready for usage. A software developer or tester performs unit tests to ensure that the code conforms to its design and works as intended. Each unit test is a procedure that takes an input and generates an output, which is then compared to an anticipated result.

### Test case 1: Log in and Sign-Up button.

**Input:** User sign up detail, Email, username and Password.

**Expected output:** Correct input details message or wrong details entered message.

**Test results**: When user entered correct results a successful message was displayed and otherwise for wrong details a message was displayed showing wrong details entered.

**Status:** Test passed.

### Test case 2: Logout button.

**Input:** Press the logout button.

**Expected output:** The user is log out of the system and directed to log in page.

**Test results:** Test passed.

### Test case 3: About page.

**Input:** Press the about page.

**Expected output:**  The about pages is displayed showing details about system.

**Test results:** Test passed and the about page was displayed in the web application.

### Test case 4: statistics page.

**Input:** Press the statistics page, user data and user id

**Expected Output:** Display the user statistic page.

**Test results:** Displayed user overall and today’s statistics.

**Test status:** Test passed.

### Test case 5: Start camera for laptop.

**Input:** Camera permission.

**Expected Output:** Camera starts and feed live stream video to model.

**Test results:** Camera started and video stream give as output.

**Test status:** Test passed.

## Integration Testing

Integration testing is a kind of software testing that examines the way in which many application components communicate with one another and the overall system. It is less concerned with the individual components themselves and more interested with the communication paths and interactions that exist between the various components.

### Test case 6: Database integration of log in and sign up.

**Input:** Database, user details, Email, username and password.

**Expected output:** The user successfully registers to database and log in using details in database.

**Test results:** Database successfully connected and details retrieved successfully.

**Test results:** Test passed.

### Test case 7: Database connection with the statistic page.

**Input:** User Id, User sitting posture datapoints.

**Expected output:** specific data from user retrieved from database and input data updated accordingly.

**Output:** User specific data was retrieved from the database and shown on the page.

**Test status:** Test passed.

### Test case 8: Start running the model in the laptop from web application.

**Input:** User press start button.

**Expected output:** A pop up window appears and a video stream is also displayed on the window.

**Output:** a window pops out with the detected posture shown on the screen.

**Test status:** test passed.

## System Testing.

System testing is a type of testing that verifies that the entire system, including hardware and software, works as expected. This type of testing is done after integration testing has been completed.

### Test case 9: Whole system testing.

**Input:** User sitting posture at current time.

**Expected output:** User posture is predicted as whether correct or wrong sitting posture.

**Test output:** User sitting posture was predicted as either wrong or correct and displayed statistics.

**Test status:** test passed.

## System Deployment and Implementation Procedures.

After the machine learning model was developed and evaluated its performance, the model was deployed in the web application developed using JavaScript, HTML, and CSS. The model was deployed by letting JavaScript start a new process that starts the camera and the model which will return the user data of current sitting posture and updated in the database consequently.

### Process In the Implementation Plan

* Identification of tools to implement the project.
* Installation of the system.
* User training.

### System specification and module required.

* Laptop or PC.
* Browser.
* Camera installed to PC
* XAMPP server.

### Installation process.

The system can be easily accessed through the browser is all the necessary data are hosted online otherwise here is the installation process.

* Install XAMPP server and start MYSQL and Apache server.
* Install PHP environment in the laptop.
* Update the database with the file halting.sql
* In the browser type <http://localhost/index.html>
* The log in or create an account and press start in home page.

# CHAPTER SIX: CONCLUSION AND RECOMMENDATION

## Discussion

My sitting posture detection application project focuses on detecting sitting posture using machine learning. The application can detect how a person is sitting and provide real-time feedback to the user by using a camera. The application employs convolutional neural networks to train machine learning models that can recognize various sitting postures. The application then gives the user suggestions on how to improve their posture, such as adjusting their chair or taking a break from sitting.

The application can also track the user's progress over time, so they can see how their sitting posture has improved. This can be an excellent way to encourage the user to maintain proper posture and make the most of their sitting experience. All of these features make it simple for the user to maintain proper posture and monitor their progress.

## Limitations.

There are some restrictions and difficulties encountered during the development of the project. First off, the data sets used to train the model are often not as accurate or diversified as required, which leads in erroneous predictions and outcomes. Second, the task's intricacy necessitates a lot of computational power, which may be challenging for many web application developers who lack the required tools. Additionally, because of the restricted access to the underlying hardware and the need for a dependable internet connection, using a web application for such a job might be challenging to accomplish. Finally, because of the nature of the work, the model must be continually updated and modified to reflect the evolving preferences in sitting positions. Since upgrading the model requires frequent changes in the code and deployment, this may be challenging to do with a web application.

Overall, there are a number of restrictions and difficulties that must be solved in order to assure the accuracy and dependability of the machine learning model developed for the detection of sitting position in a web application. Developers must be aware of the different challenges they may encounter in order to effectively finish the project, from gathering and preparing the data sets to integrating the model in a web application and updating it often.

## Recommendation

In order to ensure the accuracy of the model, it is recommended to use a large training dataset to train the model, as well as to use data augmentation techniques to increase the variety of data that is used for training. Additionally, it is important to use a cross-validation technique to assess the accuracy of the model, as well as to use hyperparameter optimization to fine-tune the model. By taking these steps, the model can be optimized to ensure the highest level of accuracy for posture detection.

Additionally, in the web application there can be included a functionality that reminds the user to change position after some few minutes. This can help user to be able manage the different sitting posture and improve their health.

## Conclusion

The goal of the project on sitting posture recognition in a web application using machine learning was to develop a web application that could recognize and categorize sitting postures in real-time. The major output of the research was the effective creation and use of a machine learning algorithm that could precisely identify and categorize sitting positions. This method was included in a web program that was intended to work on desktop computers. The user may change their posture in a more pleasant and healthful way thanks to the application's feedback feature, which was also able to be provided.

The project's findings indicate that sitting postures may be precisely detected and classified using machine learning algorithms built into web applications. The project's application gave users feedback and let them change their posture in a way that was good and more beneficial to their health. The study highlights the use of machine learning algorithms to recognize and categorize postures, enabling the creation of online apps that may help users' posture.

# REFERENCE

1. Babri, M. A., & Kaur, J. (2018). Real-Time Human Posture Detection Using Machine Learning Techniques. International Journal of Computer Applications, 173(7).

2. Chen, Y. H., & Liao, P. H. (2010). Human posture recognition from depth images with a deep belief network. Pattern Recognition, 43(9), 3040-3049.

3. Fu, Y., & Zhang, Y. (2017). Human posture recognition using a Kinect. Soft Computing, 21(5), 1389-1397.

4. Gan, Y., Chen, P., & Zhou, W. (2015). Human posture recognition using convolutional neural networks. Neural Computing & Applications, 25(4), 731-740.

5. Gao, S., Sun, J., & Zhu, Y. (2017). Human posture recognition based on multi-view RGB-D images. In 2017 IEEE International Conference on Multimedia and Expo (ICME) (pp. 612-617). IEEE.

6. Liu, Y., & Kanade, T. (2009). Human posture recognition using a single depth image. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on (pp. 1725-1732). IEEE.

7. Majumder, S. K., & Chaudhury, R. (2013). Human posture recognition using silhouette image. In 2013 International Conference on Electrical, Electronics, Computer Science and its Applications (EECS) (pp. 1-5). IEEE.

8. Mandal, S., & Mukhopadhyay, S. (2014). Human posture recognition using support vector machine. In 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 859-862). IEEE.

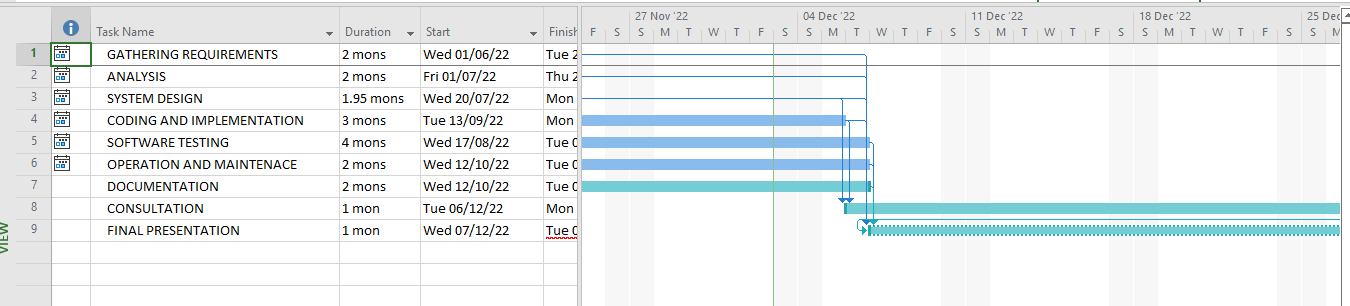
9. Ozbulak, U., & Topaloglu, A. (2008). Human posture recognition using neural network ensembles. In 2008 IEEE International Conference on Systems, Man and Cybernetics (pp. 3697-3702). IEEE.

10. Rao, S., & Reddy, S. (2010). Human posture recognition using SVM classifier. In 2010 Second International Conference on Computer Engineering and Technology (pp. V2–739-V2–742). IEEE.

11. Shen, Y., & Guo, S. (2014). Human posture recognition using support vector machine. In 2014 International Conference on Computer, Communication and Control (IC4) (pp. 642-645). IEEE.

# APPENDICES.

## APPENDICES A: GANNT CHART



## APPENDICES B: BUDGET

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **BUDGET** | | | | | |
|  |  |  |  |  |  |
| **ITEMS** | |  | **COST** | |  |
| **Laptop** | |  | **Ksh 50,000** | |  |
|  |  |  |  |  |  |
| **web camera** | |  | **Ksh 7,000** | |  |
|  |  |  |  |  |  |
| **Stationery** | |  | **Ksh 1,000** | |  |
|  |  |  |  |  |  |
| **Software licensing** | |  | **Ksh 2,500** | |  |
|  |  |  |  |  |  |
| **Software Figma** | |  | **Ksh 3,000** | |  |
|  |  |  |  |  |  |
| **Miscellaneous** | |  | **Ksh 10,000** | |  |
|  |  |  |  |  |  |
| **TOTAL** |  |  | **Ksh 73,500** | |  |

Table 3: Budget

## APPENDICES C: SAMPLE CODE

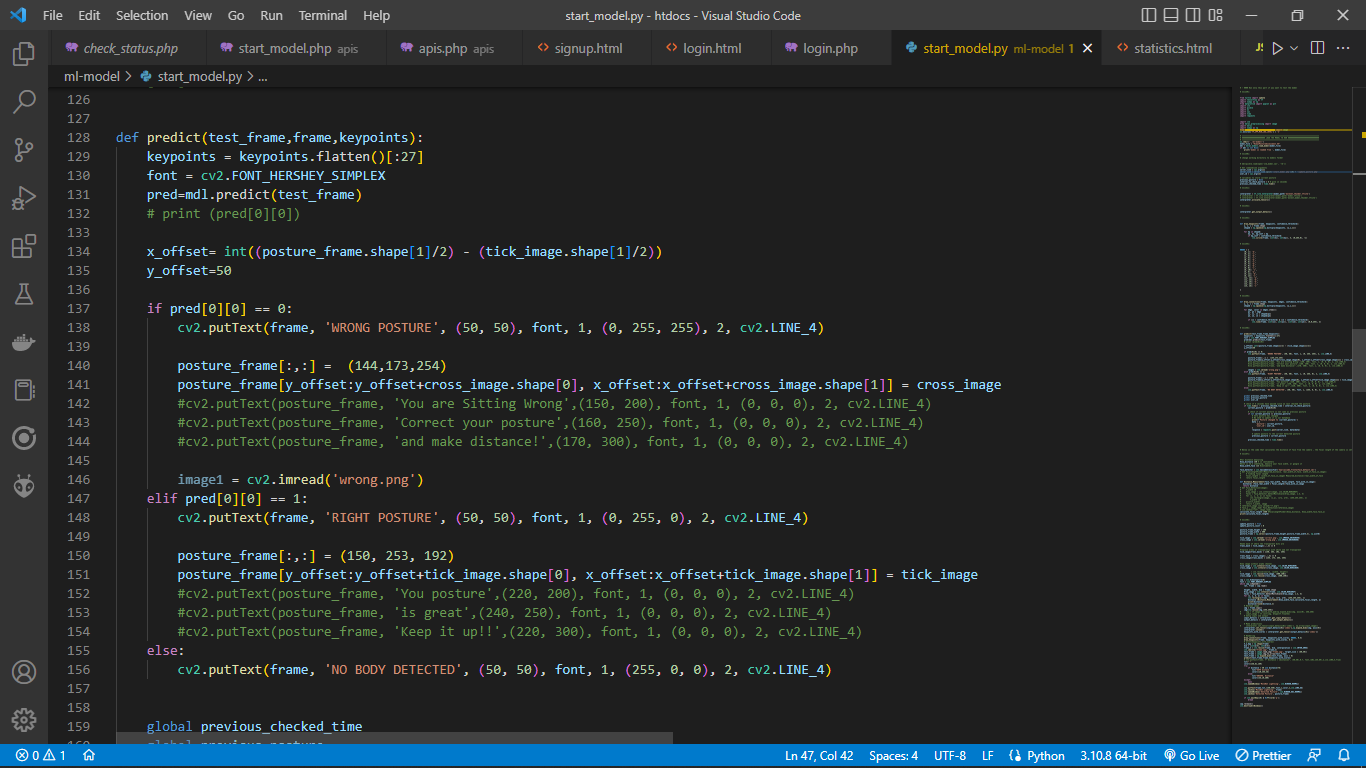


Figure 18: Sample predict model

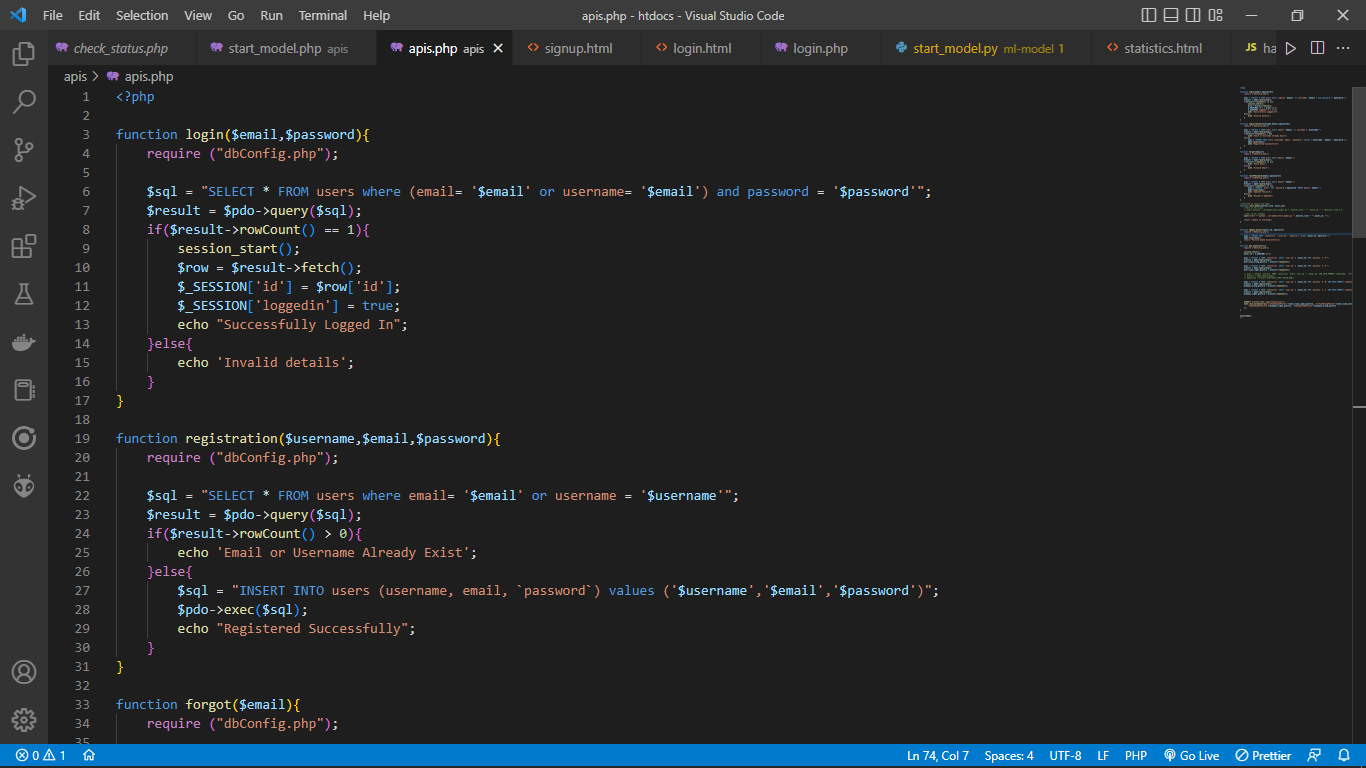
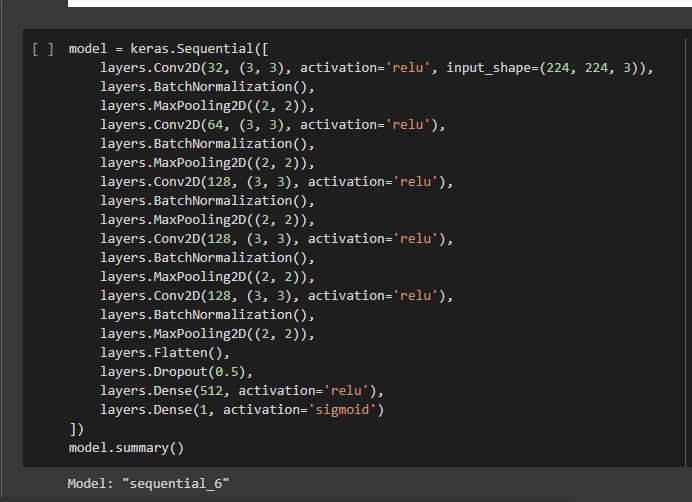
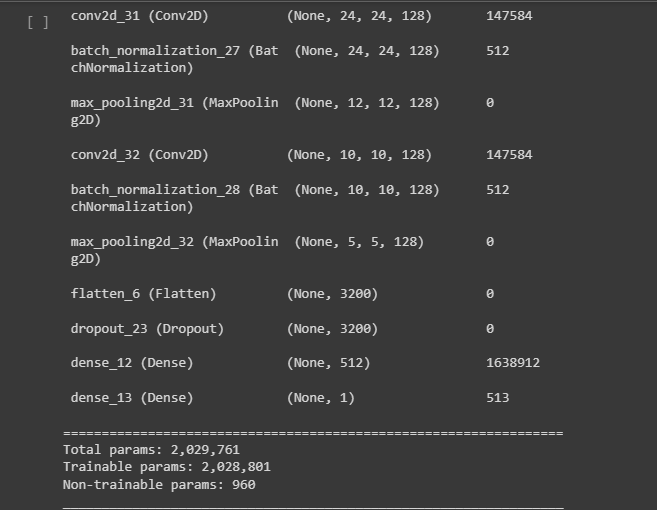


Figure 19: Sample code PHP

 Figure : CNN code