**Workspace Ergonomics and Posture Effectiveness Detection using Computer Vision with Machine Learning Approach**

By

Muhammad Haris Fahmi Bin Faizal Hisham
17007780

Dissertation submitted in partial fulfilment of
the requirements for
Bachelor of Information Technology (Hons)
(BIT)
January 2022

Universiti Teknologi PETRONAS
32610 Seri Iskandar
Perak Darul Ridzuan

**CERTIFICATION OF APPROVAL**

**Workspace Ergonomics and Posture Effectiveness Detection using Computer Vision with Machine Learning Approach**

by

Muhammad Haris Fahmi Bin Faizal Hisham

17007780

A project dissertation submitted to the
Information Technology Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
Bachelor of Information Technology (Hons)
(BIT)

Approved by,

________________________

(Ts Ahmad Izuddin B Zainal Abidin)

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
January 2022

**CERTIFICATION OF ORIGINALITY**

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

MUHAMMAD HARIS FAHMI BIN FAIZAL HISHAM

# ABSTRACT

The continuous emergence of people using computers in the recent decade together with the adoption of online learning and teleworking during the Covid-19 pandemic has greatly resulted people especially employees and students to use computers for longer periods of time on a daily basis. Consequently, individuals may experience health side-effects associated with extended use of computers caused from maintaining poor postures and having poor workspace ergonomics. Hence, this final year project aims to addresses the health complications that stems from prolonged poor posture and having workspace with poor ergonomics, to develop a software capable of detecting individual's posture and the effectiveness of the workspace ergonomics through the implementation of machine learning and computer vision technologies and utilize the developed software and collecting relevant data from surveys in order to evaluate the effectiveness or rather the impact of implementing this software to individual's health.

Findings indicated that unhealthy workspace ergonomics can lead to musculoskeletal disorders, carpal tunnel syndrome, computer vision syndrome and even contributing to development of poor postures which can further instigate back pain, neck pain, headaches, poor circulation, impaired lung function, incontinence, disrupted digestion, constipation, heartburn, and kyphosis.

After conducting posture and face distance experiments with the assistance of the developed prototype software and collecting symptoms data from test participants through surveys, the results retrieved further solidifies the hypothesis and proves as supporting evidence that having poor posture and poor workspace ergonomics can lead to the negative health effects as previously stated.

Additionally, this study also includes the role of computer vision technology as well as machine learning which together can be integrated into a desktop application to assist in detecting poor workspace ergonomics and poor posture to further prevent or reduce the magnitude of such effects. It is proven that the developed prototype featuring posture and workspace ergonomics detection capabilities had showcased that computer vision can effectively detect human posture as well as improving one's workspace.

# ACKNOWLEDGEMENTS

I would like to begin this dissertation by acknowledging all the important people that had supported me throughout the entirety of my Final Year Project. First and foremost, I would like to firmly express my utmost gratitude to my supervisor, Ts Ahmad Izuddin B Zainal Abidin for showing his genuine commitment and support. This project would not be realized if it wasn't for the joint efforts of my supervisor. I was able to receive thorough feedback regarding my project and direct me to do the right things and for giving me the approval to proceed with my Final Year Project on the selected topic, 'Workspace Ergonomics and Posture Effectiveness Detection using Computer Vision with Machine Learning Approach'. I would like to personally thank him for believing in my talent and putting the effort to support me during my FYP1 semester dated in September 2021 and my FYP2 semester in January 2022. Secondly, I would also like to express my utmost gratitude and my appreciation to my former FYP1 examiner, Assoc. Prof. Ts. Dr. Wan Fatimah Wan Ahmad for providing her critical feedback and suggestions to further improve my project. Thirdly, I would like to thank my FYP coordinators Dr. Manzoor Ahmed Hashmani and Assoc. Prof. Ts Dr Said Jadid Abdulkadir for providing clarifications for any course related enquiries. Lastly, I would like to thank my family and close friends that had supported me throughout my life. They had shown commitment and assisted me during difficult times. I am truly grateful with everything that I have, and I truly appreciate and thank everyone.

# TABLE OF CONTENTS

## LIST OF FIGURES



## LIST OF TABLES

# ABBREVIATIONS AND NOMENCLATURES

| Abbreviation | Explanation |
| --- | --- |
| CNN | Convolutional Neural Network |
| MSD | Musculoskeletal Disorder |
| CV | Computer Vision |
| ML | Machine Learning |
| OSHA | Occupational Safety and Health Administration |
| CVS | Computer Vision Syndrome |
| WHO | World Health Organization |

# CHAPTER 1

# INTRODUCTION

This Final Year Project will embark in the field of data sciences which utilizes computer vision technologies for the development of an artificial intelligence-based application that aims to detect individual's posture as well as workspace ergonomics using a camera sensor which allow the computer to detect and recognize human postures together with identifying the angle rotation of associated workspace objects and perform monitoring and evaluation to promote a healthy workspace ergonomics together with maintaining an effective posture.

## 1.1 Background

After the official declaration of the COVID-19 as a pandemic by the World Health Organization (WHO) on 30$^{th}$ January 2020 and following the ever-widespread of the Covid-19 pandemic, many countries had imposed nationwide lockdown to restrict movement of citizens to prevent potential transmission of the coronavirus. As a direct consequence, companies and institutions had declared working from home and online learning regime where employees and students are required to stay at home and conduct their day-to-day activities online. Typically, when working from home, the surroundings does not possess a similar environment as working from office. Some households lack a properly dedicated and healthy workspace and furthermore, some households do not have proper equipment, tools and even the appropriate furniture for their own personal workspace. Some will use a coffee table instead of a computer desk. This raises the concern for proper workspace ergonomics. Workspace ergonomics involves with the science of designing a workplace and its environment that focuses on the capabilities and limitations of the person involved in the work activity. Without implementing an effective computer workspace ergonomics, it can gradually result in health issues like musculoskeletal disorders and even contributing to poor posture development in the long term.

Computer vision refers to the application of machine learning and neural networks to provide supervised learning to machines and enable systems to derive important information based on the provision of digital images, videos, and other visual inputs. It is an interdisciplinary field under the subfields of artificial intelligence. Computer vision first introduced in 1960s where researchers had attempted to mimic a human visual capability by integrating camera sensors to computers. In the recent decade, there has been continuous improvements in deep learning techniques and technologies that brings revolutionary advancements in computer vision which arise tremendous applications of complex computer vision technologies in our daily lives such as self-driving cars. Thus, computer vision can also be applied into detecting individual's posture and associated workspace items to evaluate the effectiveness of workspace ergonomics and posture.

## 1.2 Problem Statement

Due to working from home and online learning, office workers and students spends hours sitting in front of a screen which tend to result in the development of bad posture habit. Additionally, poor workspace ergonomics can also be a health concern which typically exist at homes due to lack of knowledge or consideration. Khan et al. (2012) had conducted a study on people's awareness on workspace ergonomics and revealed that 47.67% of the test subjects had never heard of the term ergonomics. This shows that not all individuals who regularly use computers in their daily lives are aware of ergonomics and how poor ergonomics can bring health concerns. In the long term, poor workspace ergonomics and poor posture can bring negative effects which gradually affect the health and the quality of life of individuals.

## 1.3 Objectives

- To investigate how computer vision can assist in detecting workspace ergonomics & poor posture and the effects.
- To develop a software capable of detecting individual's posture and the effectiveness of the workspace ergonomics through the implementation of machine learning and computer vision technologies.
- To evaluate the effectiveness of implementing this software to individual's health.

## 1.4 Scope of Study

The scope of study refers to the extent to which the research area will be explored throughout the project. The scope of this project will involve with studying the negative health effects of workspace ergonomics and poor posture by conducting evaluation and survey on university students ages from 20 to 23 years old. This segmentation on user demographics primarily focuses on university students as they undergo online learning for their current semester at the time of writing this dissertation. The software that will integrate the computer vision and machine learning will be limited to detecting the individual's posture from the side profile and the workspace ergonomics which only features to evaluate based on estimating the distance between the person to the monitor and measuring the angle orientation of workspace objects specifically a chair.

# CHAPTER 2

# LITERATURE REVIEW AND/OR THEORY

The primary goal of this literature review is to develop understanding and discover the level of widespread on computer use and the various negative health effects associated with prolonged poor posture along with having an unhealthy workspace. Moreover, it also explores the present situation of computer vision domain and covering the applications of computer vision in detecting human postures.

## 2.1 Emergence of Computer Use

The emergence of computers throughout the decade has shown a rising trend as computers becomes more advanced and affordable. Computers are considered as a necessity to conduct day-to-day operations. Most work and leisure activities nowadays are being digitalized which gradually results in the need of computers. Especially now during the pandemic where having a computer is essential to carry out daily tasks for students and employees which resulted in an even more drastic increase of computer users.

Similarly, Ryan and Lewis (2017) mentioned that over the past few decades, the utilization of computers has risen tremendously. According to their research article which only focuses on a single country that is United States of America revealed that 89% of households have access to a computer and further disclose that there is a rising trend of computer use in the past decade and is expected to rise exponentially in the coming years. This indicates that computer is a common necessity amongst families in their everyday life. At the present time, people commonly use computers for a broad range of applications such as online banking, entertainment, communications, accessing health care and more.

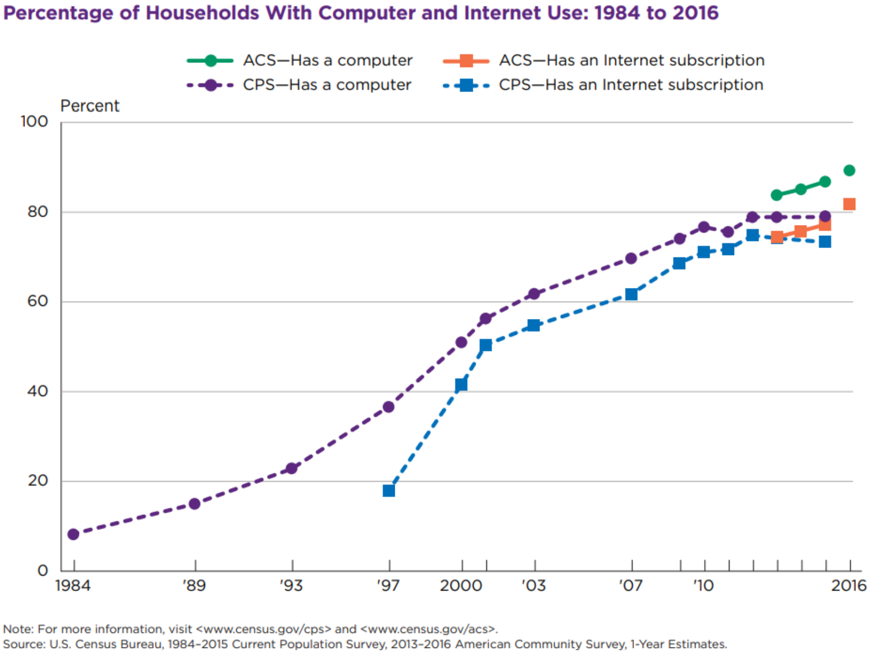



**Figure 2.1.1 Percentage of Households with Computer and Internet Use in America: 1984 to 2016**

There are several limitations to this study, although most of their sources are retrieved from a reliable source namely the American Community Survey (ACS), the data provided are roughly estimates with existence of marginal errors which results in deviations from the total population that uses computers from the different states of America which implies that the results provided from the article are not entirely accurate. Additionally, this study only limits to one country specifically America, therefore, it does not reveal the precise number of populations that uses computers for the whole wide world. Furthermore, even though this article was published in 2016, it does not reflect today's current population of computer users. Regardless, this article is proven useful as it showcases the rising trend of computer use as the years increases.

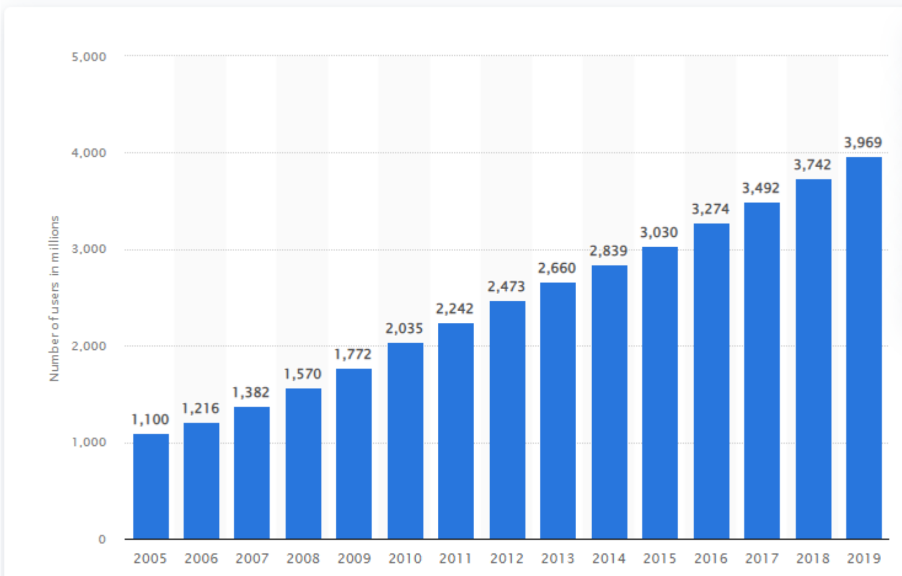



**Figure 2.1.2 Number of Internet Users Worldwide From 2005 to 2019 (in millions)**

Although the graph provided in Figure 2.1.2 showcases the number of internet users worldwide, it can be noted that it also includes users who uses the internet from their computers and even smartphones. However, it can be assumed that an increase in internet users can also indicate that computer users are also increasing as well. Today as of 2021, there are 4.66 billion internet users which makes up about 59.5% of the global population. Additionally, stated that according to market tracker International Data Corp, it is noted that there are 302 million units of laptops and desktop computers were being sold back in 2020 which is a subtle increase of 13% from the previous year sales and the highest number of units sold since 2014 (Pressman, 2021). Based on this observation, it is evident that the number of people using computers will continuously keep on increasing as the years go by.

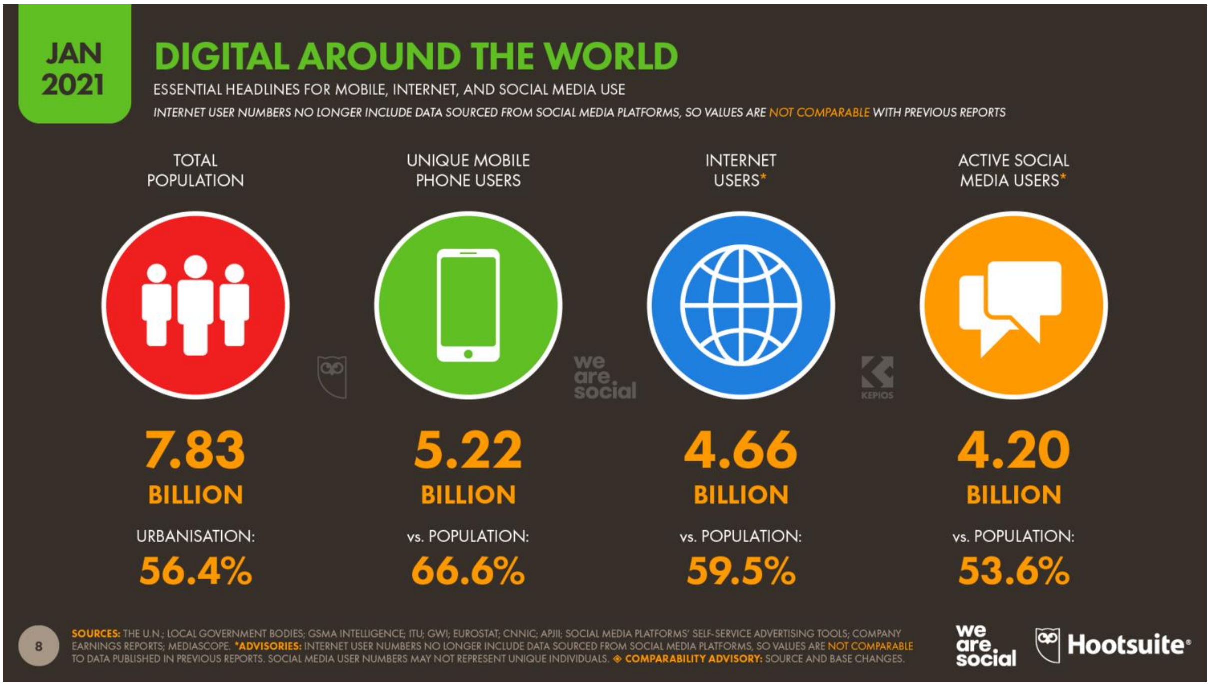



**Figure 2.1.2 Total Number of Internet Users in 2021**

There is a downside to this as more and more people use computers, it is more frequent for people to experience the underlying health complications of poor workspace ergonomics and poor posture and since people had to undergo online regimes, it further enhances the likelihood and severity for people to get diagnosed with the negative health effects. This phenomenon is formally regarded as “pandemic posture”.

## 2.2 Negative Health Effects Associated with Poor Workspace Ergonomics

According to the International Ergonomics Association, ergonomics is defined as the scientific discipline which applies physiological, anatomical, psychological, and engineering theories to support interactions among individuals and other elements of a system. Designing a workplace that follows this concept in terms of worker’s tasks, equipment uses, and the surrounding environment is referred to as ergonomic design and with proper implementation can enhance worker’s potential by improving productivity and job satisfaction, while also benefit the employer with reduction in health costs and absenteeism (Khan et al., 2012).

Khan et al. (2012) stated that the lack of good ergonomic design for extended periods of time can adversely affect computer users with visual and muscular fatigue and discomfort especially in various anatomical parts particularly the eyes, the muscles within the neck, upper back, shoulders, and arms. Long hours of sitting can contribute to fatigue as it reduces the blood flow circulation to the muscles, bones, tendons, and ligaments, with possibilities of leading to stiffness and even pain. Maintaining such steady positions for greater hours without taking a break and stretching can place even greater stress on the muscles and joints.

Musculoskeletal disorders are defined by the Bureau of Labour Statistics of the Department of Labour as a musculoskeletal system and connective tissue diseases and disorders when the event or exposure leading to the case is bodily reaction, overexertion, or repetitive motion. The body utilizes the muscles, tendons, and ligaments to perform any physical tasks, often in awkward positions or in frequent activities which can gradually cause disturbing pain and injury. Repetitive movements, sudden motions, and prolonged muscle effort are the primary causes of musculoskeletal disorders. This generally involves with injuries or disorders located at the muscles, nerves, tendons, joints, cartilage, and spinal discs. Examples of musculoskeletal disorders includes sprains, strains, and tears, back pain, carpal tunnel syndrome and hernia.

Borhany et al. (2018) concluded that musculoskeletal symptoms are related with extended use of computers. They had conducted a survey of patients spending more than 3 hours of computer use and share their experience of any discomfort or pain. Result shows that out of a total of 150 participants aged from 17 to 50 years old, about roughly 40% experience pain, discomfort, and musculoskeletal problems. This evidently stipulates that it is common for workers and even students to experience musculoskeletal disorders and for those who spends more hours using computers are more probable to experience those symptoms and diagnosed with musculoskeletal disorders. Hence, it can be concluded that musculoskeletal symptoms are associated with prolonged use of computers.

Another notable finding derived from this study reveals that 16% of the participants experience wrist pain. This can be associated with the ergonomic aspect during their computer activity. This could potentially be related to carpal tunnel syndrome as addressed later in this section.

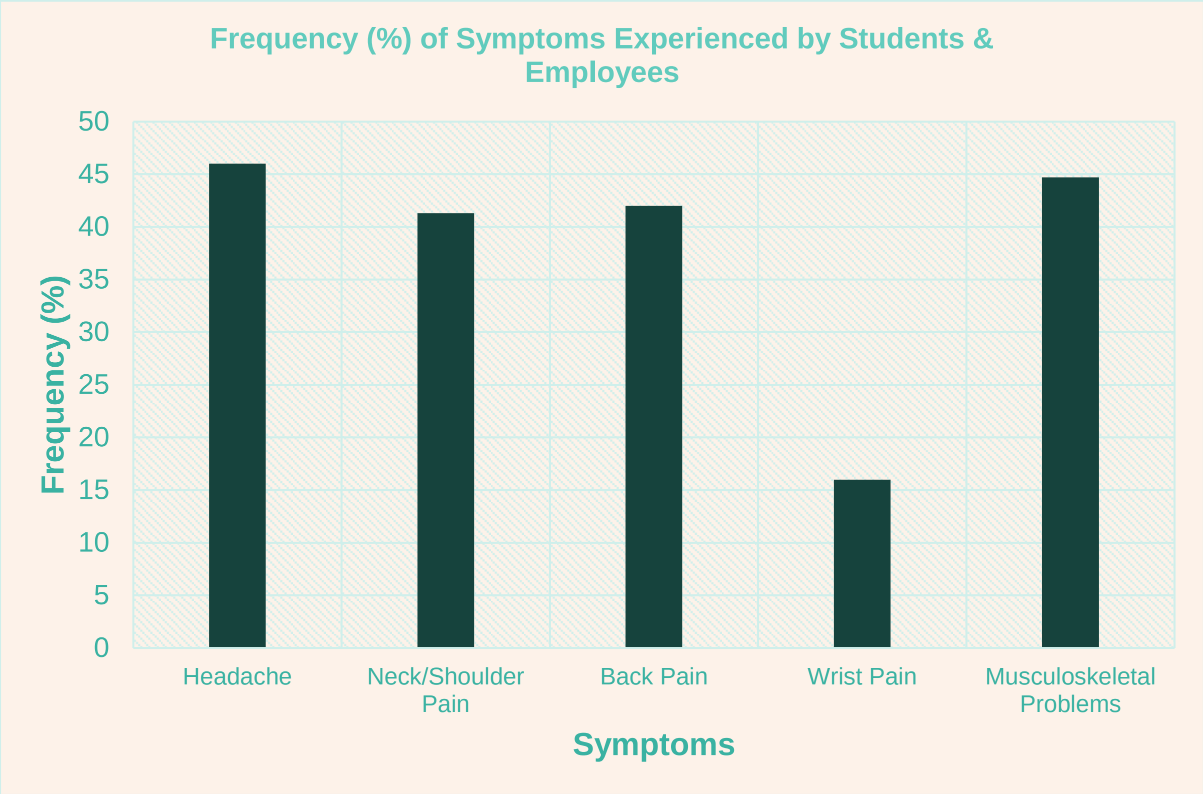



**Figure 2.2.1 Survey Results Showing the Frequency of Symptoms Experienced by Students & Employees**

To provide criticism to this study, this survey only has 150 participants which is sufficient to form a strong finding, but with more participants involved can provide much reliable results. Additionally, the participants are made up of 80% being male and 20% being female. Therefore, it cannot be concluded whether gender could potentially be a factor when it comes to developing musculoskeletal disorders. This study is considered quite good as it surveys participants who are students or employees and further categorized them based on their level of education and even their lifestyle such as involving with smoking or exercising. However, it does not reveal from the results whether the lifestyles or level of education did have an impact to the results.

Another known illness related to poor implementation of workspace ergonomics is carpal tunnel syndrome which is a compression neuropathy of the median nerve as it passes through the carpal tunnel (Thomsen et al. 2008). The carpal tunnel refers to as the narrow passageway surrounded by bones and ligaments located at the palm side of the hand. It is considered as the most frequent compression neuropathy that is commonly found in computer professionals. Mohamed and Sathiyasekaran (2006) argued that the trauma resulted from repetitive hand movement has been identified as an irritating factor for carpal tunnel syndrome affecting particularly to individuals whose tasks required repeated forceful finger and wrist flexion and extension as part of their daily activities. Obviously, students and employees who uses computers are also susceptible to carpal tunnel syndrome because they usually involve hand movement through typing with a keyboard and interacting and navigating with the mouse.

Another notable effect is computer vision syndrome which refers to as the eye and vision problems associated with near work experienced during computer use (Khan et al., 2012). It is very common for individuals to experience eye strains as the symptoms includes blurry or even double vision, irritation, headaches, eye fatigue, colour perception change, decreased and visual efficiency. With proper implementation of workspace ergonomics, this syndrome can be minimized.

### 2.3 Causes of Poor Posture

The prevalence of poor posture within individuals occurs when the spine is positioned in unnatural positions, in which the curves of our spine are being emphasized. This results in the joints, muscles and vertebrae being in stressful positions. Prolonged poor posture leads to a buildup of pressure on these tissues. Ultimately, this can result in bodily discomfort and even chronic back pain.

The ideal posture happens when the skeletal alignment is in a state of balance that protects the supporting structures of the body against injury or progressive deformity. The spine is composed of three different segments. Given from a side view profile, the spine assembles three natural curves where it is situated at the neck, mid back, and low back which refers to as the cervical spine, thoracic spine, and lumbar spine respectively. The proper posture where individuals are recommended to follow is where the spines are aligned at its anatomically neutral position where the three main curves are being properly maintained and not emphasized. Hecht (2020) stated that by positioning the spine at its proper spinal alignment indicates balance between the muscle and bones which further protects the body against potential injury or stress.

Many individuals may instinctively lean forward to have a closer look at a laptop’s screen or monitor when doing computer tasks. This can lead to the development of poor posture if being held for long periods of time as the head is being protruded forward and the curve within the neck is being emphasized. When the head protrudes forward at a 45-degree angle, the neck will act like a fulcrum where pressure is applied to maintain the balance. As the neck is being pressured, it also results in adjustments of the position of the upper back and lower back. This position essentially compresses the neck which can cause fatigue, headaches, weak concentration, muscle tension and back pain.

Furthermore, Drockrell et al. (2012) argued that with poor workspace environment can also contribute to the development of poor posture. According to their research article, they had tested to see how the test subject were to behave as they arrange the computer and the computer accessories at a given angle and positioned the keyboard beyond the edge of the desk together with the lack of sufficient leg room. The result showed that the test subject had bad posture when adapting to such poor environment.

Overall, this reveals that long hours of computer use, and adopting unhealthy workspace ergonomics can gradually lead to the development of poor posture. As students and even employees who are routinely using computers as part of their daily activities, the possibility of bad posture development is very likely.

### 2.4 Health Effects of Prolonged Poor Postures

Although maintaining poor posture for a short amount of time does not bring detrimental health effects, extended period of unhealthy postures can derive serious complications to individuals and with it usually being associated with a variety of health-related effects typically affecting one's general well-being. As addressed by the Spine Institute of Southeast Texas, these health consequences include back pain, neck pain, headaches, poor circulation, impaired lung function, incontinence, disrupted digestion, constipation, heartburn, and kyphosis. Hecht (2020) had also concluded that bad posture can even result in degeneration of the muscles and the joints.

### 2.5 Solution to Poor Workspace Ergonomics

Workspace ergonomics includes numerous different factors that needs to be considered. Ergonomics is essentially the requirement to have the body parts to be maintained at a comfortable position. According to the Occupational Safety and Health Administration, there are various guidelines and procedures provided to help design a healthy workspace.

First and foremost, the chair the person sits are the most crucial furniture when it comes to ergonomics as it primarily contributes to the individual's spinal alignment. It is important to have a chair configured with armrest and backrest support which allows the shoulders to be relaxed and the arms comfortably placed on the sides with armrests slightly below the elbows.

The chair should be placed within the desk giving close access workspace items. The chair must be set at an appropriate height and position the wrist in line with elbow. The backrest of the chair should accommodate and support the person's entire back. It should be adjustable and contour to the curve of the person's lower back. The seat pan's height must be adjusted so that the feet of the person sitting lays flat on the surface of the ground.

Additionally, for visual concerns, the display of the monitor must be positioned at or slightly below the eye level with the area of the screen in which the person normally focuses on should be 15 degrees below the eye level. Secondly, the optimal distance between the screen and the eyes should be about an arm's length roughly 40cm to 75cm. Thirdly, it is suggested to obstruct the screen from any external sources of light as it could lead to glares. This can safely prevent or reduce computer vision syndromes.

Computer accessories also plays a pivotal role in workspace ergonomics. The keyboard should be placed entirely flat or slightly tilted and the wrists should be positioned straight when typing. Furthermore, it is recommended to use wrist rests as it can support the wrist and the palms during resting period. On the other hand, the mouse should be ergonomically design where it is comfortable for the hand. The wrists should be kept straight during mouse work activities, and it is substantially advisable to move the whole arm instead of bending the wrists sideways.

### 2.6 Optimal Viewing Distance

The optimal viewing distance refers to as the ideal distance between the individual and the screen of the computer. Finding the optimal distance can greatly reduce the chances of inducing any negative health effects such as eye strains, dry eyes and more while also allowing the individual to operate with the computer effectively in which it does not greatly influence their productivity.

The ciliary muscles enable the lens in the eyes to adjust its shape in order to give focus to objects at a close distance or at a far distance. Eye strain is a consequence that occurs when the ciliary muscle in the eyes is being overworked from performing activities which requires the eyes to focus on nearby objects or far objects for long periods of time (Bedinghaus, 2021).

Generally, it will need a greater muscular effort to focus on nearby objects. Hence, as a consequence, it can result in eye discomfort. On the other hand, sitting from a further distance does not strain the muscles to focus on nearby objects which can prevent any risk for eye discomfort but nonetheless, it can also potentially affect the person's visual ability to see the characters and the images displayed and their ability to interact effectively. Additionally, if the viewing distance is far, it can eventually cause the person to unmindfully lean forward to the screen in order to clearly see the characters and images displayed which can affect the individual's posture.

Suggestions for the optimal distance varies. In fact, there are various sources which recommended different distances for people to comply with. Chawla et al. (2019) suggested that it is recommended to keep the distance of roughly 76 to 100 centimetres. On the other hand, Goplan and Haral (2018) claimed that it is advisable to remain between 50 to 100 centimetres as the ideal viewing distance. Similarly, according to the Occupational Safety and Health Administration (OSHA), the preferred viewing distance that should be followed is between 50 to 100 centimetres. Regardless of varying inputs from different sources, people should adjust their viewing distance within those specified ranges as long as it does not reach below 50 centimetres and settle where the person deem comfortable. This allows the person to effectively interact with their setup while preventing any risks for eye discomfort. Nonetheless, to take further precautions, people are advised to sit at a further distance within the given range as sitting at a closer distance is not ideal.

While viewing distance is an important factor in preventing risks of computer vision syndrome, it is also advisable for people to take frequent breaks in between. People are suggested to act in accordance with the '20-20-20 Rule'. This rule indicates that for every 20 minutes, the person should take a rest from their screen and observe objects that are 20 feet away for a minimum of 20 seconds (Goplan & Haral, 2018). Taking frequent breaks and looking at distant objects allows the eyes to rests from after having it to focus on nearby objects for extended periods of time.

## 2.7 Computer Vision and Convolutional Neural Network Algorithm

Computer vision refers to as an interdisciplinary field that is categorized under artificial intelligence which involves giving the ability for machines to observe objects using a camera which further enables computer to recognize, track and measure associated objects for processing images (Tian et al., 2020). It was first introduced in 1960, where the concept of extracting 3D geometrical information from a 2D perspective was initially proposed by a person named ‘Larry Roberts’.


**Figure 2.7.1 Example of Computer Vision with Object Detection**

Advancements in computer vision technologies combined with deep learning has been achieved and perfected over the years using one specific algorithm namely ‘convolutional neural network’. Convolutional Neural Network (CNN) is an algorithm exclusively used in deep learning which essentially consume a set of input data in the form of images, assign importance to various aspects or objects in the image and then distinguish one another (Saha, 2018). Saha further explained that CNN is able to retrieve spatial and temporal dependencies from an image by applying relevant filters and is able to compress images into a form that can be processed effortlessly without sacrificing any quality or features which is important for attaining accurate predictions. Fundamentally, CNN utilize principles from linear algebra, notably matrix multiplication in order to analyze images and recognize their patterns.

CNN is comprised of three main layers namely the convolutional layer, the pooling layer and the fully connected layer which is used to identify the intended object from a given image.

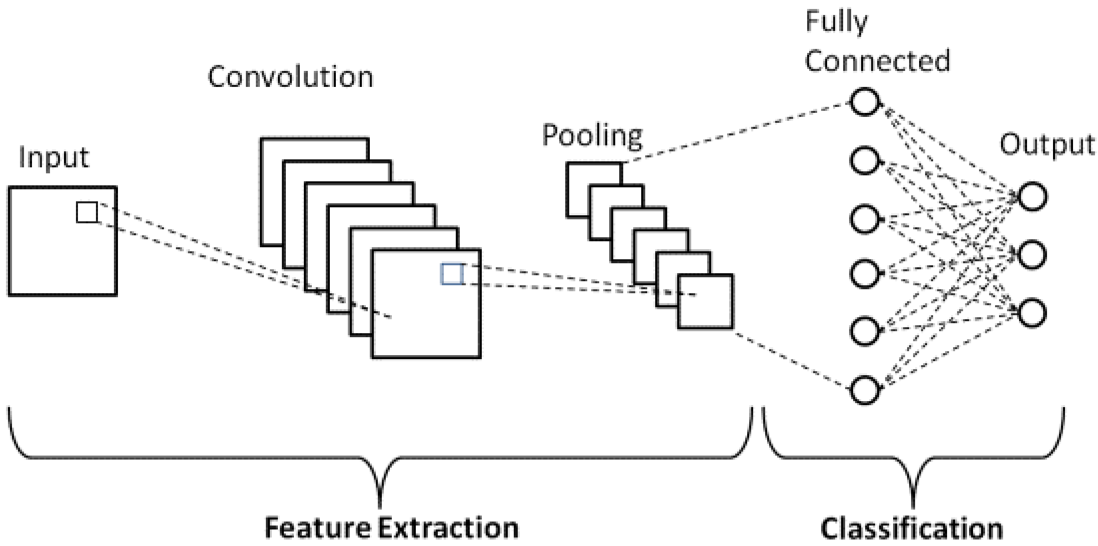



**Figure 2.7.2 CNN Processes Visualized**

First and foremost, the convolutional layer and its main purpose is for feature extraction. It requires different components such as the input data, a filter, and a feature map. The filter is a feature detector which performs a process called as convolution which basically assess the receptive fields of the image and check the presence of any high-level features like edges. Due to the unstructured nature of image data, it is impossible to directly use images for processing. The core principles when dealing with images is that images can be represented in the form a matrix of numeric pixel values. Hence, the convolutional layers convert images into numerical values which then further enables it to be interpreted by the neural network and extract the important patterns. Secondly, the pooling layers performs dimensionality reduction which reduces the parameters in the input. Similar to the convolution layer, this layer also uses filter to assess the across the entire input and applying aggregation function to the values within the receptive field and populate those values into the output array. Some of the advantages of the pooling layer is that it improves efficiency, reduces complexity, and lowers the risk of overfitting. Finally, the fully connected layer performs classification based on the features that was previously extracted prior to this layer to deliver the output. The final classification process distinguishes between dominating and certain low-level features within the images and utilizes SoftMax activation function to classify the inputs accordingly.

### 2.8 Applying Machine Learning together with Computer Vision

Machine learning refers to as the field of study of computer algorithms that can improve automatically through experience with the use of data. In basic terms, computers are able to learn based on the data that has been gathered. Thus, with computer vision, it is able to capture the raw data in the form of visual representations and is then being consumed into machine learning models to arrive predictions. It is evident that applying machine learning together with computer vision helps provide effective solutions to a broad spectrum of problems from numerous industries. Some good applications of computer vision with machine learning that is being used to solve problems include detecting plant diseases in agriculture, performing fault diagnosis in industrial engineering, recognizing brain tumour and segmentizing endoscopic images for cancer in healthcare, and object detection in autonomous vehicles (Kulikajevas et al., 2021).

### 2.9 History of Monitoring Postures

In the past, there has been different approaches to monitoring human postures using camera sensors such as using depth cameras, 3D-motion analysis, video surveillance, and Kinect sensors or through another viable solution which is to install wiring sensors directly to the human body to acquire data such as using wearable textile sensors, inertial and magnetic sensors and radio-frequency identification (Kulikajevas et al., 2021). Both methods have its drawbacks as using camera-based systems requires establishing an appropriate distance, good lighting, proper calibration, and no obstructions whereas placing sensors to the human body can hinder the person’s movement and interactions.

### 2.10 Application of Computer Vision for Posture Detection & Related Work

As previously mentioned, computer vision imitates human’s visual capabilities by giving computers access to use camera sensors to detect objects from its surroundings. To achieving such results, a python library known as ‘OpenCV’ can be used. OpenCV is an open-source library used for image and video analysis (Culjak et al., 2012). Additionally, other libraries can also be used for detecting human postures namely OpenPose and Mediapipe which refers to as a library that is capable of jointly detecting human body, face, and hands by manipulating specific joints within the human body. These joints are referred to as ‘key points.’ Both libraries can be integrated together with OpenCV for posture detection. Kulikajevas et al. (2021) stated that posture detection enables the capture of kinematic parameters of the human body, which is essential for various applications, namely assisted living, healthcare, physical exercising and rehabilitation.

An article published by Chen (2019) aims to achieve similar idea of developing a system on sitting posture detection. It further shows that the OpenPose library was being integrated and could detect poor sitting posture from frontal view. The study disclosed that it requires dataset on poor posture and training the machine learning model and apply an appropriate algorithm to help the computer to recognize the person’s posture. Results shows that the system can detect poor sitting posture. However, there’s a limitation to this system because frontal view can only provide whether the body posture is good or not based on whether the body is tilted from side to side. It is one of the ways of indicating poor posture however, it is not an effective way to evaluate postures and determine whether the spinal posture is in an ideal or unpleasant position. In comparison, my project focus on the side view angle where the curvature of the spinal posture can be easily observed which can effectively assess individual’s posture. Additionally, the research produced by the author did not cover on the study of health concerns associated with poor posture and workspace ergonomics but rather on technical implementation of computer vision on posture detection.

### 2.11 OpenPose and MediaPipe Library Comparison

OpenPose is widely regarded as one of the most popular open-source libraries which features pose estimation (Cao et al., 2018). This library can be used for detecting individual’s posture when it is being integrated together with OpenCV. Once a person’s posture is detected, it will automatically lay out the person’s body key points. However, there are certain limitations when implementing this library. First and foremost, Sharma (2020) stated that one of the disadvantages of OpenPose is that there is a trade-off between speed and accuracy. Having high accuracy is slow which requires a longer time to execute the pose estimation process. Secondly, due to its delayed processing, it can also affect its tracking performance in real-time whereas it does not have any issue if the source of input is a still image. It can be noted that attempting to track an individual's posture using OpenPose in real-time can result in slow performance and rapid movement of the body key points. Since my project focuses on real-time posture detection using a camera, it can suffer from these limitations as delays and sudden movements of retrieving the precise coordinates of the individual's body key points can potentially affect the final outcome of my project in terms of the system's user experience from the lack of fluidity and the overall accuracy of posture data retrieval.

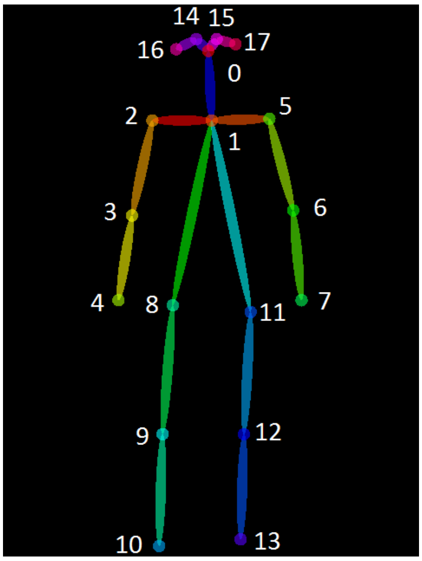



**Figure 2.11.1 OpenPose Key Points**

Another common library which offers the same set of features is known as 'MediaPipe. MediaPipe is a library that provides open-source cross-platform and assorted machine learning solutions for live and streaming media. Such available solutions include hand tracking, face mesh, selfie segmentation, object detection, face detection and of course human pose detection and tracking capabilities just to name a few. Alavi (2020) mentioned that the advantage of implementing MediaPipe over OpenPose is its noticeably faster performance which was achieved by utilizing graphics processing unit acceleration and multi-threading. Furthermore, MediaPipe enables a wider range of detecting different body parts which consist of 32 different key points available whereas OpenPose is only limited to 17 key points.

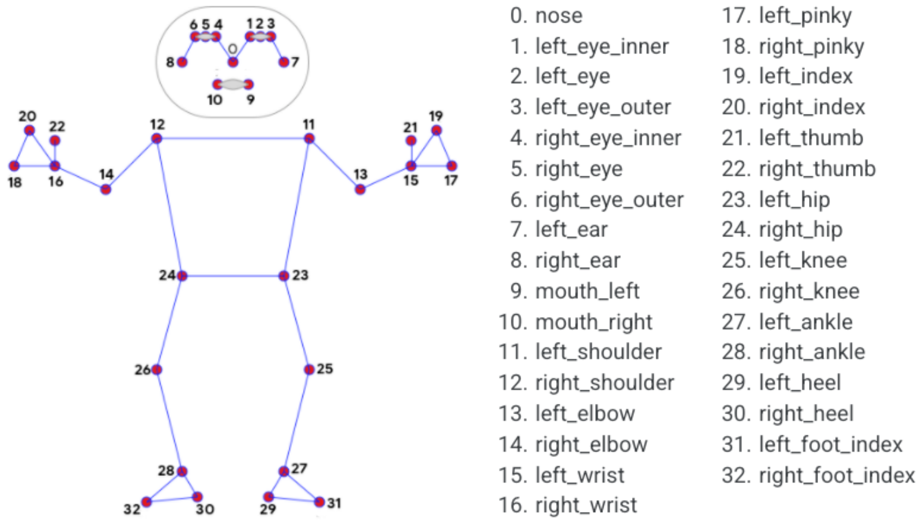



**Figure 2.11.2 MediaPipe Key Points**

Based on this information, it can be concluded that MediaPipe is the better option in comparison to OpenPose. Hence, the system to be developed should integrate MediaPipe to derive more accurate results without sacrificing system performance.

To further test the performance of each library, a 328 second video featuring a person performing a deadlift exercise is being used together with each library and the performance is derived based on the frames per second that is shown during the demonstration. Higher frames per second indicates that the video can run smoothly whilst having each respective libraries to execute its processes for detection and assigning the key points to its designated locations. The frames per second can be seen in the white text on the top left corner of the diagrams provided.

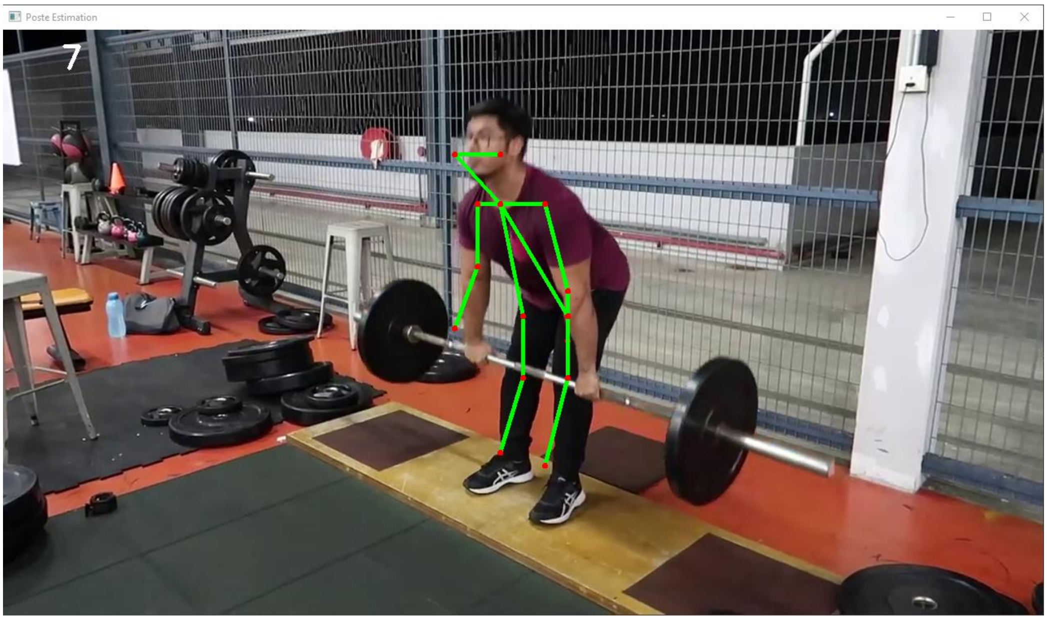



**Figure 2.11.3 Posture Estimation using OpenPose Library**

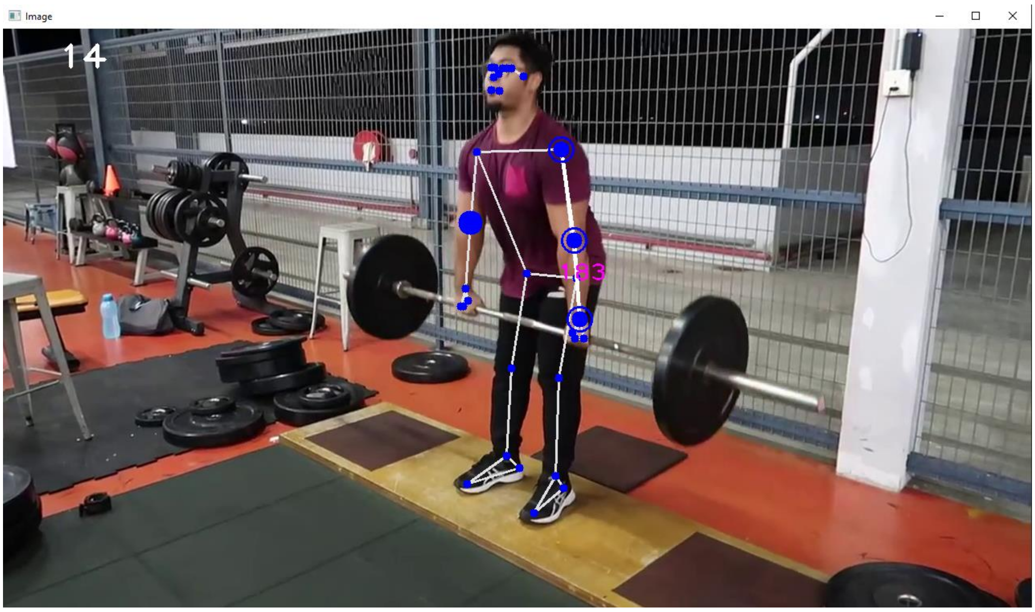



**Figure 2.11.4 Posture Estimation using MediaPipe Library**

Following the demonstration, OpenPose library had achieved an average of 9.23 frames per second whereas MediaPipe has an average of 19.99 frames per second. This observation denote that MediaPipe performed 116.576% better than OpenPose and it can be concluded that MediaPipe is significantly more efficient pertaining to handling posture estimation.

# CHAPTER 3

# METHODOLOGY/PROJECT WORK

## 3.1 Introduction

In this chapter, it scrutinises the methodologies and the necessary processes applied for the development of the system and address how the experiment was conducted to derive information from students to assess and evaluate posture as well as the ergonomic aspects of workspaces. The methodology lists out specific procedures or techniques used to achieve the project's deliverables and satisfy the stated objectives. In this case, the methodology refers to the series of processes used to deliver the software and how the research experiment was carried out.

## 3.2 Agile Methodology

The agile methodology is the methodology that was be applied for this project development. The agile methodology is defined as a software development methodology that emphasize in the notion of iterative development. It is a project management process that promotes continuous inspection in every aspect which ultimately delivers progress and improvements in an iterative and incremental manner. As the name implies, the methodology offers flexibility compared to other known software development methodologies like the waterfall method. In each scenario where clients or other stakeholders requires sudden amendments to the software during the development stage, the developers can comply and adjust efficiently to the new requirements. The agile methodology includes 5 different phases namely requirements, design, development, testing and deployment as arranged in Figure 3.2.1.

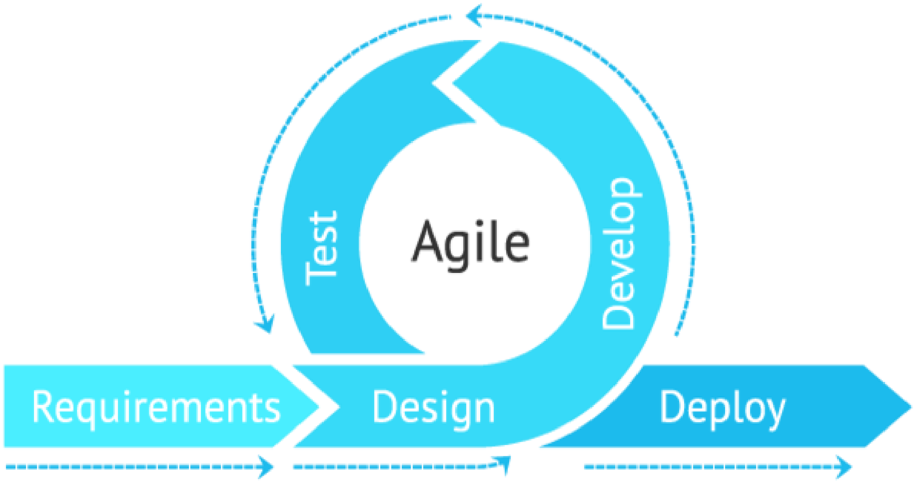



**Figure 3.2.1 Agile Methodology Process**

### 3.2.1 Requirements Phase

The requirements phase involves gathering all the necessary requirements prior to the development of the software itself. It addresses the scope, the features and how the software will function. This phase comprises of 4 processes namely performing feasibility study, requirement gathering, software requirement specification and software requirement validation. The documents related to the requirements and planning are presented in Figure 5.1, Figure 5.2, Figure 5.3, Figure 5.4 in the appendix.

### 3.2.2 Design Phase

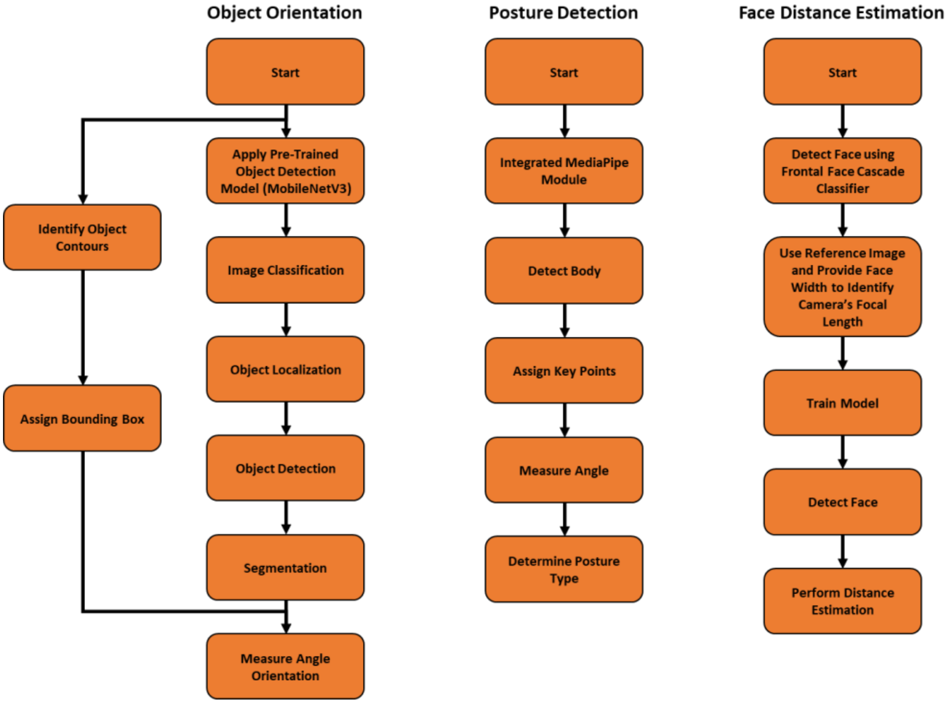
The design phases involve with producing the designs of the interface for the software and integrate all the identified requirements into the designs. This phase identifies the essential tools such as the programming language, syntax libraries, and basic frameworks. This outcome of this phase is the software's graphical user interface (GUI).

In this case, the software that is planned to develop involves using Python programming language as it's the primary language involved in machine learning projects. The software that will be used for the software development is PyCharm for the main software functionalities that includes computer vision and machine learning and PyQt5 designer for the front-end design. Additionally, the python libraries involved to detect human postures is MediaPipe and OpenCV. Lastly, the model used for object detection is MobileNETV3.

### 3.2.3 Development Phase

This phase is the part where the actual development begins. This involves applying the designs as stated in the design phase and perform the coding to develop the software itself. This project was developed mainly in PyCharm Integrated Development Environment and utilized the OpenCV and MediaPipe libraries together. It further implements PyQt5 framework to develop the application that can detect human postures as well as identifying workspace objects for evaluating the effectiveness of the workspace ergonomics.

#### 3.2.3.1 Computer Vision Development Processes




**3.2.3.1.1 Computer Vision Development Process**

Since the development of the software involves with integrating computer vision, there are specific procedures required to be followed to effectively construct the deliverables in the proper way to give the desired outcomes. The system comprises of three different functionalities, each will be explained independently.

For the object orientation, it integrates a pre-trained object detection model called as “MobileNetV3” which consist of everyday objects with certain workspace objects namely chair and table included. Secondly, it requires image classification where the images identified are being classified into their respective classes such as “monitor”, “chair”, “person”. Thirdly, after classifying the objects, it is required to perform object localization by which it locates the presence of object in an image and mark its location within a bounding box. Furthermore, the process will undergo object detection where after locating the presence of the object with a bounding box, it predicts the classes of the objects located. Moreover, the project will undergo segmentation which involves identifying image elements and determining which object they belong to. In the meantime, the process also involves with identify object contours from the real-time image in order to recognize as one object and proceed with assigning the bounding box. After the processes are done, only then the system will be able to measure the angle orientation based on the orientation of the bounding box.

The posture detection utilizes MediaPipe module which recognizes human body. Once a body is detected, it will assign the body key points to its respective body parts. With the given key points, it allows the system to recognize the angle and further proceed to determine the posture type.

The face distance estimation, it uses a frontal face cascade classifier to detect faces. In order to train the model to perform the distance estimation, it first requires training the model by referring to a reference image with the width of the person’s face provided. This allows the system to identify the camera’s focal length. Once the model has been trained, only then the system is able to detect the face and then proceed to perform the distance estimation.

Following that, is the actual development itself where all the core functions are integrated into PyQt5 software. Before the software can be deployed, it is crucial to evaluate the software to ensure that software has all the necessary features needed and functions the way it is intended. Lastly is the deployment of the finalized deliverable where the final product can be used.

### 3.2.4 Test Phase

The testing phase requires conducting tests on the software developed to ensure that the software is functioning as intended. This phase is compulsory prior to the software release as it will identify any potential issues or errors made that may had been overlooked during the design phase. Unit testing, system testing, integration testing, and acceptance testing are the various tests that can be made to determine any issues persist within the software.

## 3.3 Testing Methodology

The testing methodology addresses the methodologies used for the experiment to achieve information regarding person's posture and understand the workspace ergonomics. It is essential for the experiment to be conducted effectively and accurately. This means that certain variables involved during the experiment must be controlled whereas the independent variables must change accordingly.

There are few types of experiments to be performed namely posture test and eye distance test. These tests will be performed on university students aging from 20-23 years old. Each of these tests will require a duration of 3 hours to complete. Additionally, the data collection will involve with gathering primary data by using the software developed to collect the posture and the face distance data and through surveys to collect the symptoms that test subjects had experienced during the experiment.

### 3.3.1 Posture Test

The aim of this experiment is to prove the hypothesis to find out whether the person using the computer for long hours while maintain bad posture can result in health-related issues from maintain poor postures whereas maintain good posture can prevent such symptoms. Participants will be given 3 hours to use the computer and they are allowed to do any computer related activities. There will be 2 separate tests involved. The first test will require the participant to sit at a good posture for 3 hours continuously whereas the second test will require to maintain bad posture within 3 hours.

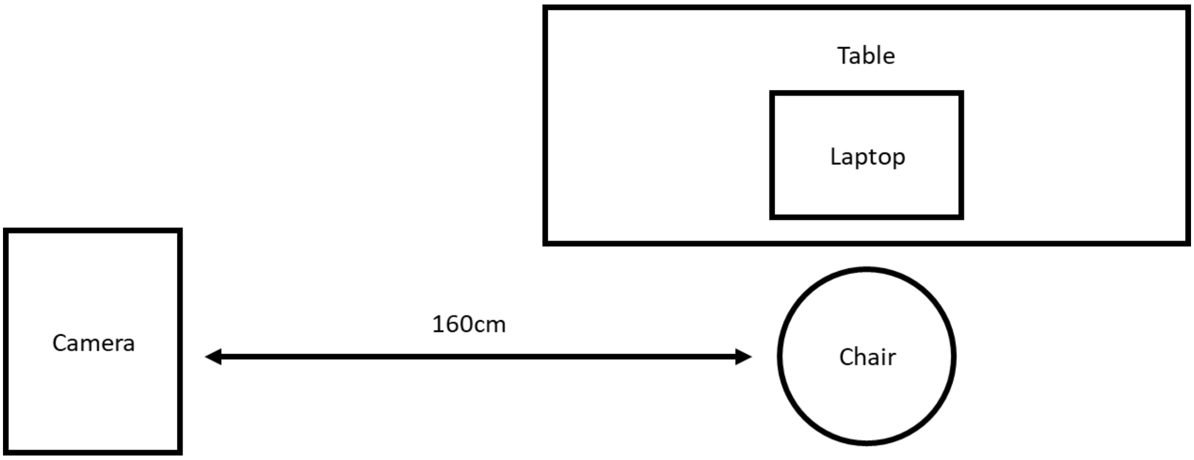



**Figure 3.3.1.1 Visualization of Posture Experiment Setup**

The camera will be placed 160 centimetres away from the chair and will be positioned at a 90° angle facing the person's left side. This provides sufficient distance for the software to detect and capture the person's entire body for the posture estimation. Variables such as lighting will be fixed for all tests.

A survey will be asked to the participant after they had finished with their experiment. The survey will ask questions before and after the experiment has been conducted.

Pre-Assessment Survey

1. Name
2. Age
3. Are you a student or a worker?
4. Do you experience the listed symptoms frequently (1+ per month)?
5. Do you have any of the following symptoms prior to the assessment?

Post-Assessment Survey

1. What kind of posture you were instructed to maintain?

2. Rate the level of headache severity.
3. Rate the level of back pain severity.
4. Rate the level of eye strain severity.
5. Rate the level of neck/shoulder pain severity.
6. Rate the level of wrist pain severity.
7. Rate the level of musculoskeletal problems severity.
8. Do you have any other symptoms not mentioned from the list?
9. What are the activities you did during the assessment?
10. Any other comments/feedback?

The pre-assessment survey will help understand the participant better by understanding their background and to see whether the participant have no symptoms prior to the experiment. The post-assessment survey involves finding out whether the person perceived any health symptoms which requires the person to rate the level of severity of the common symptoms based on a Likert scale ranging from 1 to 5 where 1 represent no emphasis and 5 represent high emphasis. Even though this survey is only related to postures, this experiment can also take the opportunity to see whether other symptoms can be inherited from prolong use of computers.

### 3.3.2 Eye Distance Test

For this experiment, the hypothesis is to find out whether the person using the computer for 3 hours will have more chances of receiving any symptoms related to computer vision syndrome. The participant will be involved in 2 separate eye distance tests. The first test will involve the person to use their computer at an appropriate distance of 50 or more centimetres as per OSHA requirements. The second test will involve the person to use their computer at a much closer distance of less than 50 centimetres. Variables specifically, the setup, position of laptop, lighting and seating position is fixed at all times while the independent variable will be the distance of the person looking at the laptop screen and will change depending on the which kind of test the person is doing.

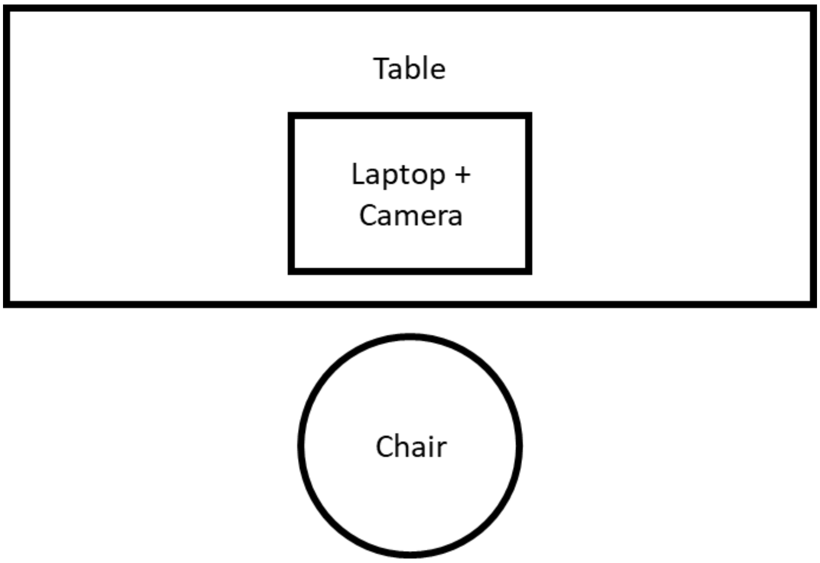



**Figure 3.3.2.1 Visualization of Eye Distance Experiment Setup**

The setting of this experiment will require a chair to be placed directly in front of the camera and having the person to interact with the laptop. The laptop screen will be angled at 90° so that it does not affect the distance estimation from the software.

Similar to the posture experiment, a survey will be asked to the participant after they had finished with their experiment to gather further information regarding the person's background and identify any symptoms relating to computer vision syndrome. The survey will ask questions before and after the experiment has been conducted.

Pre-Assessment Survey

1. Name
2. Age
3. Are you a student or a worker?
4. How many hours do you usually sent on your phone and computer?
5. Do you have any of the following symptoms prior to the assessment?

Post-Assessment Survey

1. What test were you instructed to do?
2. Rate the level of eye strain severity.
3. Rate the level of dry eyes severity.
4. Rate the level of headache severity.
5. Do you have any other symptoms not mentioned from the list?
6. What are the activities you did during the assessment?
7. Any other comments/feedback?

# CHAPTER 4

# RESULTS AND DISCUSSIONS

The result and discussions focus on the results acquired from the project work. All findings and outcomes will be visualized and analysed to extrapolate insights that supports the theories, hypothesis and meeting the project's main objectives.

## 4.1 Software

The primary objective of this final year project is to develop a software that is capable of detecting individual's posture and the effectiveness of the workspace ergonomics through the implementation of machine learning and computer vision technologies. This objective had been achieved as the prototype has been develop and managed to deliver the core functionalities as intended using computer vision and machine learning approaches. The software itself comprised of three core functionalities namely detecting body posture, provide face distance estimation and detecting workspace object's angle orientation.

### 4.1.1 Posture Detection

The first functionality of this software features body posture detection of the targeted person and classify accordingly whether the posture maintained was poor or optimal. This was achieved by implementing MediaPipe library for pose estimation and finding the coordinates of relevant key points. Since the camera is facing the left side of the person, the relevant key points are the left shoulder, left hip and the left knee and it can be interchangeable depending on the selected side. After successful detection of the relevant key points, the system will retrieve the angle using the key points. Thus, if the angle is between the range of 90° to 120°, then the system will classify it as 'optimal posture' or otherwise, 'poor posture'. Additionally, the system will display the type of posture and the angle in the information section.

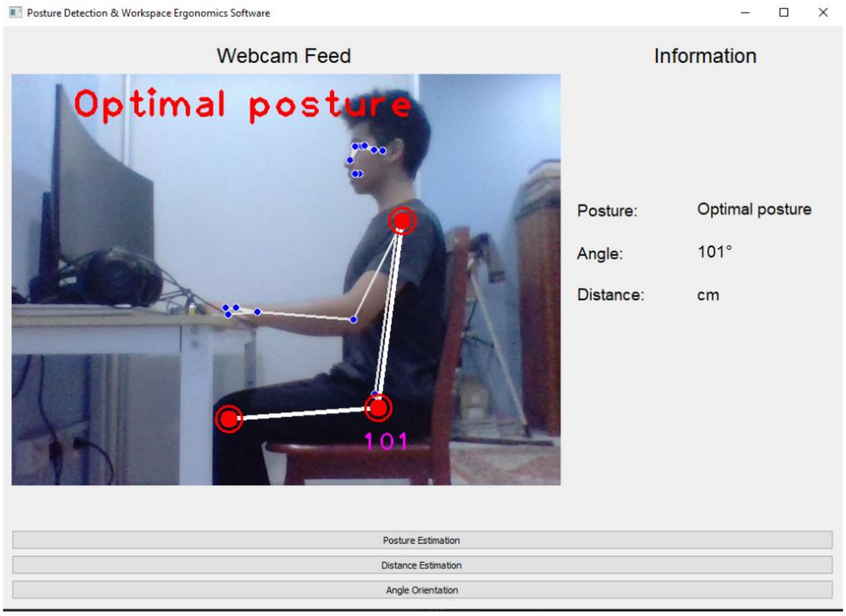



**Figure 4.1.1.1 Maintaining Optimal Posture from System**

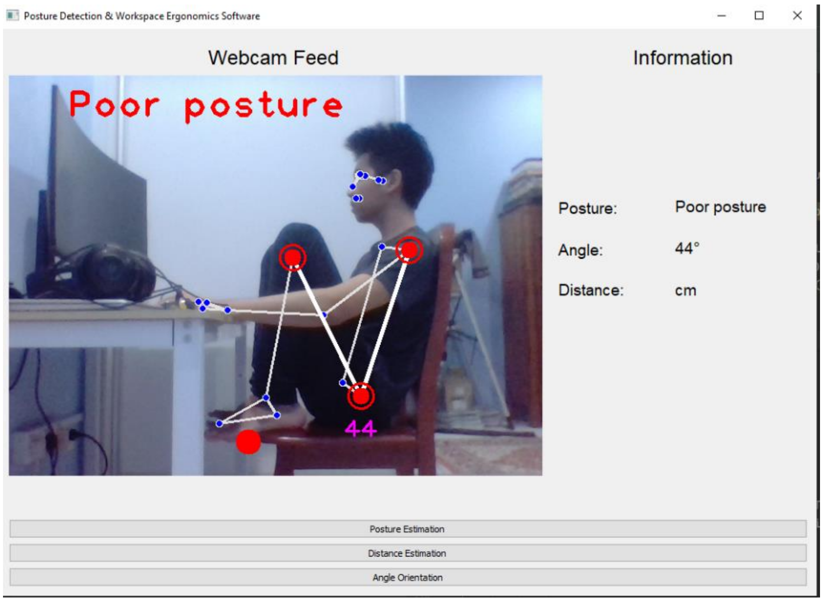



**Figure 4.1.1.2 Maintaining Poor Posture from System**

### 4.1.2 Face Distance Estimation

The second function of this software is to estimate the distance between the person and the laptop's screen. The purpose of this is to monitor a person's sitting distance to ensure that the person is not sitting proximately close to the laptop's screen as it can affect the individual's eye bringing computer vision syndromes. This is one of the aspects concerning workspace ergonomics relating to eye health. Since it is relatively difficult to detect eyes compared to faces due to its smaller size, measuring the distance between the screen and the face is another workaround for estimating the distance between the person's eyes. In order to make accurate estimations, the system undertook a supervised learning process where a reference image containing a person's face and the width of the person's face was provided. This helps in determining the focal length of the camera which helps in achieving the face distance estimations. Furthermore, the system also utilized a frontal face classifier to detect a person's face. Hence, by applying both concepts help in achieving the expected outcome.

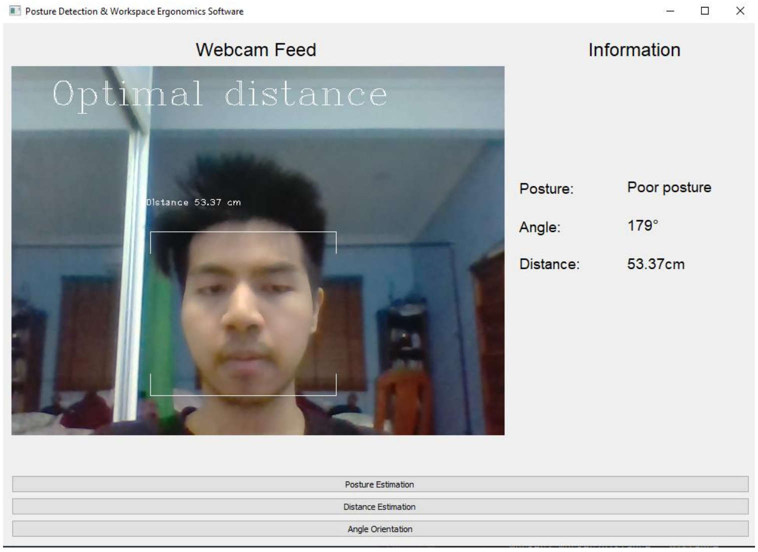



**Figure 4.1.2.1 Maintaining Optimal Distance from Software**

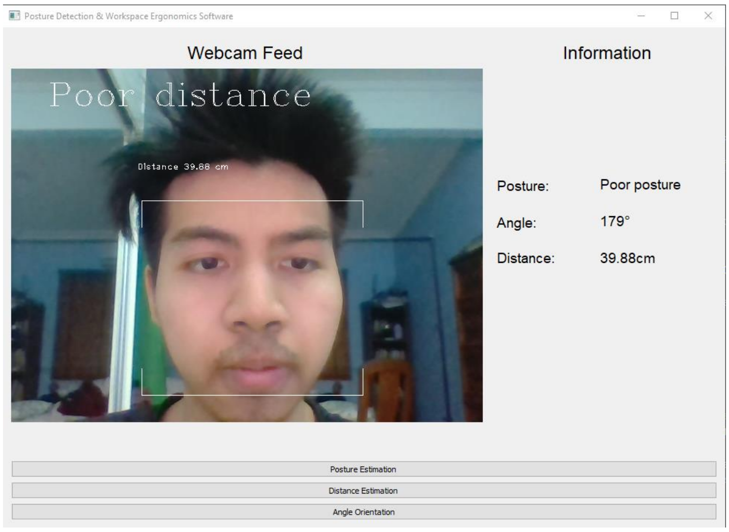



**Figure 4.1.2.2 Maintaining Poor Distance from Software**

As shown in Figure 4.1.1, it can be seen that the person is sitting roughly 53.37 centimetres away from the laptop's screen which resulted in an optimal distance. In Figure 4.1.2, the person is sitting roughly 39.88 centimetres away resulting in a poor distance.

Integrating computer vision has its flaws as the values derived is solely an estimated value based on machine learning.


**Figure 4.1.2.3 Measurement Received from Software**

**Figure 4.1.2.4 Measurement Using Measuring Tape**

To assess the accuracy of the face distance estimation, a comparison was made to observe what the software had estimated which shows that the person sitting was 56.65 centimetres as perceived by the software shown in Figure 4.1.2.3, but in reality, the person was actually sitting between 54 to 56 centimetres as shown in Figure 4.1.2.4.

A test was further performed to measure the accuracy of the face distance estimation by having a person sitting at precisely 60 centimetres away from the screen and compare with the system's input data for a span of 60 seconds. Further calculations to arrive the accuracy of the face distance estimation involves applying standard deviation and computing the margin of error. The results are shown in Table 4.1.2.1 and Table 4.1.2.2 implying that there are a roughly 2 centimetres deviation off from its precise distance giving close to accurate measurements.

| Value | Frequency |
|---|---|
| 58.11 | 2 (3.3333333333333%) |
| 58.42 | 2 (3.3333333333333%) |
| 58.74 | 2 (3.3333333333333%) |
| 59.06 | 5 (8.3333333333333%) |
| 59.38 | 6 (10%) |
| 59.545 | 1 (1.6666666666667%) |
| 59.7 | 8 (13.333333333333%) |
| 59.705 | 1 (1.6666666666667%) |
| 60.03 | 8 (13.333333333333%) |
| 60.06 | 1 (1.6666666666667%) |
| 60.36 | 8 (13.333333333333%) |
| 60.38 | 2 (3.3333333333333%) |
| 60.7 | 3 (5%) |
| 61.04 | 3 (5%) |
| 61.36 | 1 (1.6666666666667%) |
| 61.38 | 5 (8.3333333333333%) |
| 61.54 | 1 (1.6666666666667%) |
| 62.07 | 1 (1.6666666666667%) |

**Table 4.1.2.1 Values Derived from Software and its Frequencies**

| Confidence Level | Margin of Error |
|---|---|
| 68.3%, $\sigma_{\bar{x}}$ | 60 ±0.114 (±0.19%) |
| 90%, 1.645$\sigma_{\bar{x}}$ | 60 ±0.188 (±0.31%) |
| 95%, 1.960$\sigma_{\bar{x}}$ | 60 ±0.224 (±0.37%) |
| 99%, 2.576$\sigma_{\bar{x}}$ | 60 ±0.295 (±0.49%) |
| 99.9%, 3.291$\sigma_{\bar{x}}$ | 60 ±0.377 (±0.63%) |
| 99.99%, 3.891$\sigma_{\bar{x}}$ | 60 ±0.445 (±0.74%) |
| 99.999%, 4.417$\sigma_{\bar{x}}$ | 60 ±0.505 (±0.84%) |
| 99.9999%, 4.892$\sigma_{\bar{x}}$ | 60 ±0.56 (±0.93%) |

**Table 4.1.2.2 Confidence Level and Margin of Error**

### 4.1.3 Workspace Objects' Angle Orientation

Furthermore, the third main functionality is to identify workspace objects and determine its associated angle orientation. As previously mentioned, workspace ergonomics can be a driving force as to how a person habitually maintains their body posture. For example, in the office environment, if the seat that the person is sitting is not properly angled upright, it can potentially affect how the person sits which can affect their body posture. This operation is accomplished by utilizing computer vision built-in functions in order to find contours of the target object that satisfies the specified thresholds and proceed to group together and establish a bounding box surrounding the object's contours. Afterwards, after finding the contours, only then the system can determine the object's rotation angle by seeing how much the bounding box had been rotated off of from the 90° axis. Moreover, this process also utilizes object detection through the integration of MobileNetV3 object detection model and classify the given objects into their respective classes. This enables the system to see what kind of object it being shown and measure its orientation.

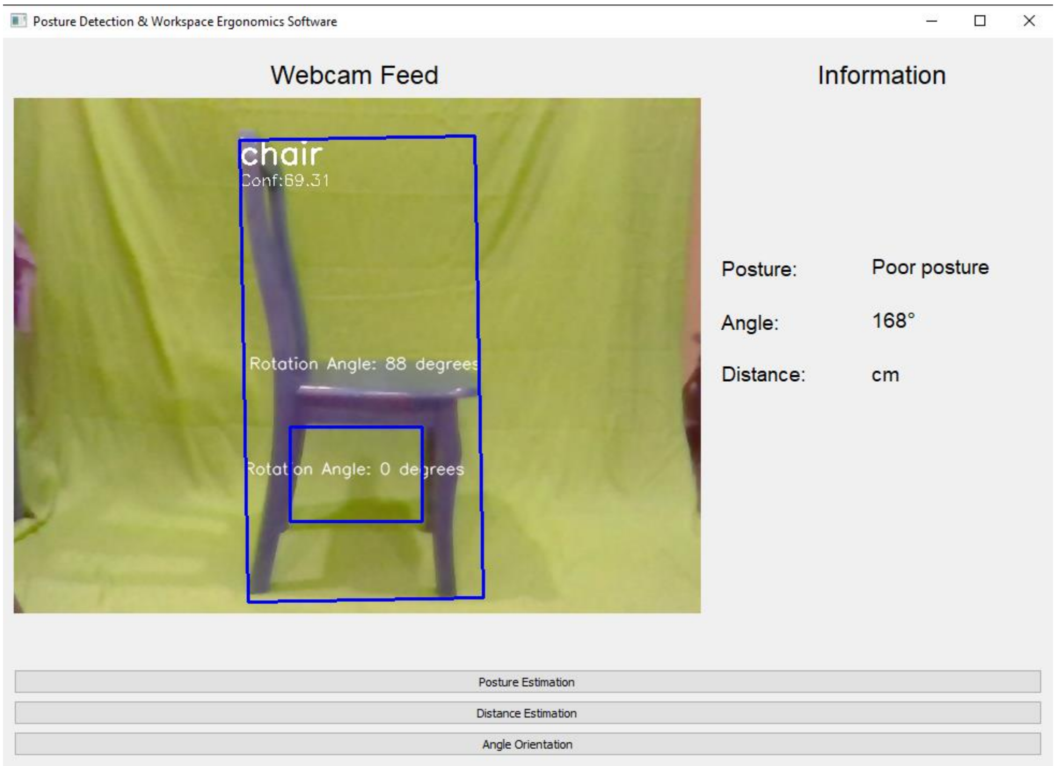



**Figure 4.1.3.1 Detecting Objects and its Angle Orientation from Software**


**Figure 4.1.3.2 Detecting Objects and its Angle Orientation Comparison**

The system works efficiently when the background is a complete solid colour that is contrasting towards the target object because it isolates the targeted object from the background and aids in distinguishing the necessary contours. This is practically demonstrated in Figure 4.1.3.2.

### 4.2 Experiment

The second objective of this project is to evaluate the effectiveness of implementing this software to individual's health. This is achieved through performing experiments on participants and retrieving the results.

For the posture experiment, participants were required to perform two different tests. The first test will involve the participant to maintain good posture for three hours whereas the second test will require the participant to maintain bad posture for three hours. The purpose of this experiment is to observe the symptoms that can possibly be derived from prolong hours maintaining good and bad postures.

The first participant had consistently maintained good posture 98.81% of the time for the first test and poor posture 99.57% of the time for the second test. This data is illustrated in the line graph shown in Figure 4.2.1.

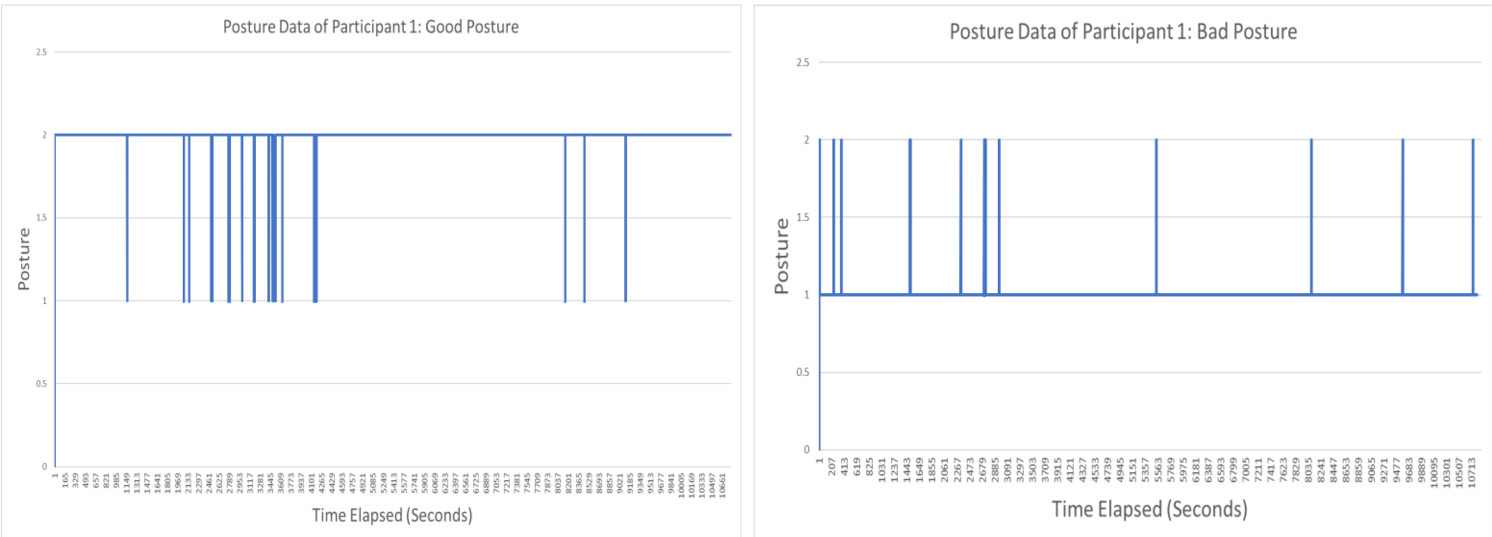
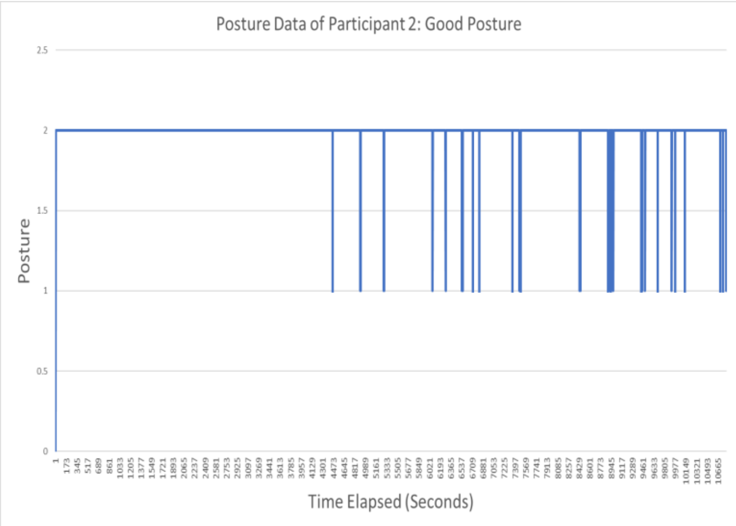



**Figure 4.2.1 Participant 1 Posture Data for Both Tests**

The second participant had maintained good posture 99.13% of the time for the first test and bad posture 98.6% of the time for the second test. This data is illustrated in the line graph shown in Figure 4.2.2.



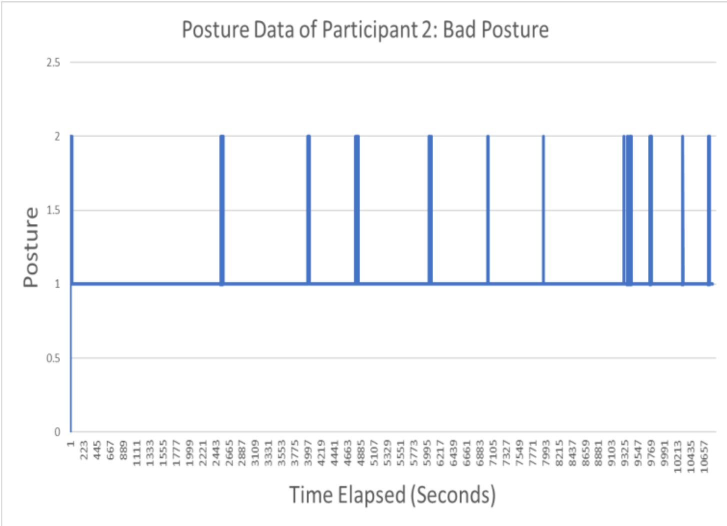




**Figure 4.2.2 Participant 2 Posture Data for Both Tests**

| Name | What kind of posture you were instructed to maintain? | Rate the level of headache severity. | Rate the level of back pain severity. | Rate the level of eye strain severity. | Rate the level of neck/shoulder pain severity. |
|---|---|---|---|---|---|
| Muhammad Haziq Firdaus Bin Faizal Hisham | Good posture | 1 | 1 | 1 | 1 |
| Muhammad Haziq Firdaus Bin Faizal Hisham | Bad posture | 1 | 2 | 1 | 1 |
| Muhammad Haris Fahmi Bin Faizal Hisham | Good posture | 1 | 1 | 3 | 1 |
| Muhammad Haris Fahmi Bin Faizal Hisham | Bad posture | 1 | 3 | 2 | 1 |

**Table 4.2.1 Posture Survey Results**

According to the survey, it is revealed that participants who maintained good posture for 3 hours' time did not experience back pain or any other posture related symptoms while on the other hand, participants who maintained poor postures experienced back pain. Another interesting finding that also occurred during this experiment is that Participant 1 had experienced eye strains regardless on both tests. Whereas Participant 2 did not experience any eye strain at all. This result is evident that maintaining poor posture for long periods of time can bring detrimental health consequences like back pain.

For the face distance experiment, the first participant had consistently maintained good distance 94.49% of the time for the first test and poor distance 99.88% of the time for the second test as shown in Figure 4.2.3.

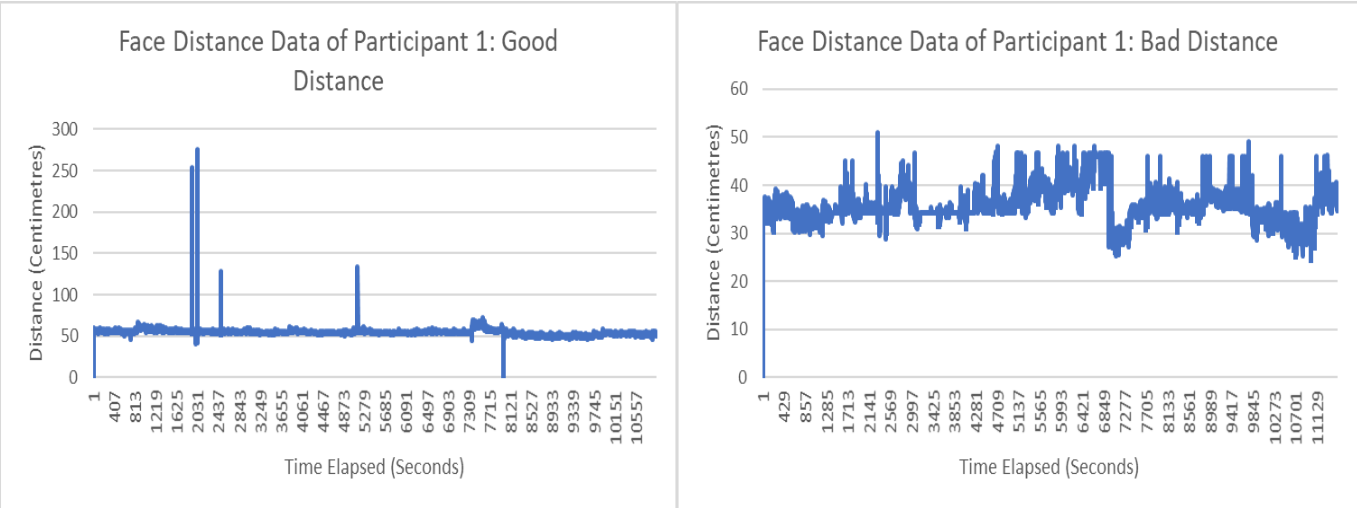



**Figure 4.2.3 Participant 1 Face Distance Data for Both Tests**

For participant 2, the participant had consistently maintained good distance 100% of the time for the first test and poor distance 100% of the time for the second test as shown in Figure 4.2.4.

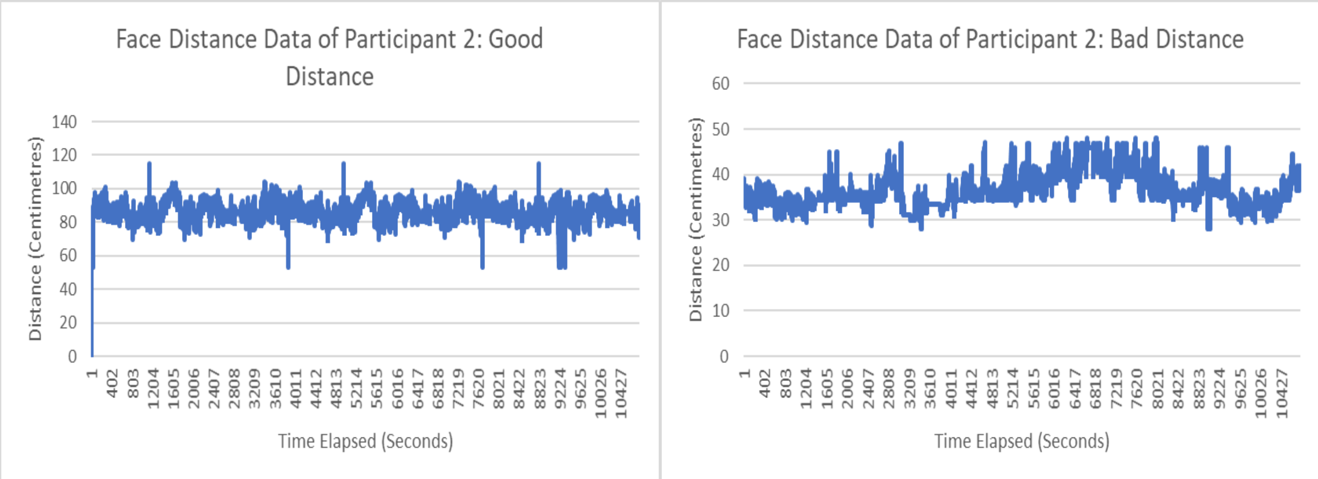



**Figure 4.2.4 Participant 2 Face Distance Data for Both Tests**

| Name | What test were you instructed to do? | Rate the level of eye strain severity. | Rate the level of dry eyes severity. |
|---|---|---|---|
| Muhammad Haris Fahmi Bin Faizal Hisham | Optimal distance test | 1 | 2 |
| Muhammad Haris Fahmi Bin Faizal Hisham | Poor distance test | 2 | 1 |
| Muhammad Haziq Firdaus Bin Faizal Hisham | Optimal distance test | 1 | 1 |
| Muhammad Haziq Firdaus Bin Faizal Hisham | Poor distance test | 2 | 2 |

**Table 4.2.2 Face Distance Survey Results**

Based on the results from the survey, participants who undergo the optimal distance test did not experience eye strain, however, the first participant did experience mild dry eyes. Participants who undergo poor distance test both experience mild eye strains and mild dry eyes. This proves the hypothesis that sitting at a close distance near the monitor or laptop screen can lead to computer vision syndromes.

There are various notable limitations that can potentially affect the final outcome and needs to be highlighted. First and foremost, positioning the system to monitor individual's posture on the side-view angle is one of the effective ways of observing postures. However, it restricts how the system views the whole person. To elaborate further, the system can properly detect someone's posture on one side that is being exposed however on the other side of the person will not be apparent from the camera because the side that was being exposed covers the other side. Therefore, it can gradually affect how the system interpret the person's posture and potentially derive inaccurate posture classification.

Secondly, poor body detection can occur. This suggests that the system could not assign the body key points in its proper body part. For example, the system can append the key point of a person's leg, but it was wrongly positioned elsewhere. This is a result of misinterpretation of the system and this limitation occurs as a result from having a poor-quality camera without depth sensing capabilities, the presence of objects that obstructs the camera's view, the colour of the person's apparels during the experiment and even environmental factors which includes poor lighting and undistinguishable background.

Thirdly, it is difficult to derive insights and effectively prove the hypothesis based on the experiment. In a way, some people may experience symptoms as a direct consequence from prolong use of computers such as back pain from maintaining long periods of unhealthy posture and eye strain from viewing a computer screen at a close distance while on the other hand, some people may not experience such symptoms regardless of how long they spent on using computers. In this case, more data is required, factors needs to be critically examined and controlled, and more time spent on computer is needed to effectively prove the hypothesis.

Furthermore, it can be a struggle for participants to express the severity of their symptoms that they experienced after the experiment has been conducted based on their personal ratings. People may rate their symptoms differently because each individuals have different judgment on how emphasized such symptoms are occurring. For example, a person may perceive that they experience a minimal back pain whereas another person may find that an equivalent level of back pain is considered as more intense. Thus, there's no proper, accurate and standardized measurement in quantifying the severity of symptoms which may insinuate that procured results could be ambiguous and debatable.

Moreover, it is also important to consider the activities the person had made prior to the experiment. A person may had spent hours on their computer before performing the test and did not experience any symptoms yet but after doing the experiment and spent more time on the computer, they started to experience negative symptoms whereas if they had not spent the time on the computer before the experiment, they may not experience those negative symptoms. This can potentially have an influence in the final results.

# CHAPTER 5

# CONCLUSION AND RECOMMENDATIONS

## 5.1 Conclusion

Computer vision is a useful technology used in different industries and can pioneer technological innovations. By implementing computer vision technologies to detect poor posture and workspace ergonomics can easily help users know the effective ways of maintaining good posture and effective workspace ergonomics which will gradually help individuals minimize or even prevent the negative health implications arise from it. Improvements in camera technologies and advancements of computer vision technologies can potentially improve capabilities, performance and accuracy which can potentially bring more precise outcomes.

This dissertation aims to investigate how computer vision can assist in detecting workspace ergonomics and poor posture and the related effects. Numerous articles from reliable sources originated from professionals had supported the argument that maintaining poor postures and having poor workspace ergonomics can eventually lead to negative health effects in the long term. Moreover, this dissertation also proves the implementation of computer vision together with machine learning and integrate into the PyQt5 framework can create a software that acts as a tool that is capable of detecting individual's posture and the effectiveness of the workspace ergonomics through the implementation of machine learning and computer vision technologies and respective users are able to observe and monitor how their posture looks like during their online activities and make necessary adjustments.

Lastly, this dissertation also aims to evaluate the effectiveness of implementing this software to individual’s health. To conclude this dissertation, workspace ergonomics and unhealthy posture can bring detrimental health effects especially to students and employees who ordinarily undergo their daily activities using a computer. It can be concluded that negative health complications constituted from continuous extensive use of computers through performing experiments for 3 hours. In addition, computer vision technology and together with machine learning, can detect objects through a camera for human posture recognition and object detection to provide preventive and reductive measures on the extent to which the effects of unhealthy posture and poor workspace ergonomics can cause.

## 5.2 Recommendations

There are various recommendations that can potentially improve the outcomes of this project. For more comprehensive coverage, it is highly recommended to involve more participants to participate in the experiment and to further identify and analyse the variety of symptoms as a direct and indirect consequence to prolong use of computers. Moreover, with increased participation can further validate the prototype in terms of accuracy. In regards to the experiment conducted, it is recommended for the setup to be properly setup with lighting, background to enable the camera to easily recognize objects for the monitoring process in order to bring much accurate findings.

Additionally, even though computer vision is effective in performing the necessary operations to arrive for posture detection and workspace ergonomics evaluation with small margin of errors, it can be even more accurate to utilize depth-sensing cameras with better specifications rather than standard webcam sensors for the reason that it can recognize depth within its surroundings and able to observe at higher resolutions resulting in acquiring more decisive readings and enabling better recognition.

Finally, in terms of the system's design and user interface, it is highly recommended for the system to have a modern yet minimalistic appearance that comply with Nielson's usability heuristics to give a more appealing and intuitive look for users.

# REFERENCES


1. Dockrell, S., Earle, D., & Galvin, R. (2010). Computer-related posture and discomfort in primary school children: The effects of a school-based ergonomic intervention. *Computers & Education*, *55*(1), 276-284.

2. Joshi, N. (2019, June 27). *The present and future of Computer Vision*. Forbes. Retrieved October 9, 2021, from https://www.forbes.com/sites/cognitiveworld/2019/06/26/the-present-and-future-of-computer-vision/?sh=70acf42b517d.

3. Burns, E. (2021, March 30). *What is machine learning and why is it important?* SearchEnterpriseAI. Retrieved October 10, 2021, from https://searchenterpriseai.techtarget.com/definition/machine-learning-ML.

4. Bowman, P. J., Braswell, K. D., Cohen, J. R., Funke, J. L., Landon, H. L., Martinez, P. I., & Mossbarger, J. N. (2014). Benefits of laptop computer ergonomics education to graduate students. *Open Journal of Therapy and Rehabilitation*, *2014*.

5. Kaelin, M. (2017, June 6). *10 ways to immediately improve workstation ergonomics*. TechRepublic. Retrieved October 14, 2021, from https://www.techrepublic.com/article/10-ergonomic-tips-to-immediately-improve-your-workspace/.

6. Middlesworth, M. (2020, January 31). *5 proven benefits of ergonomics in the workplace*. ErgoPlus. Retrieved October 15, 2021, from https://ergo-plus.com/workplace-ergonomics-benefits/.

7. Khan, R., Surti, A., Rehman, R., & Ali, U. (2012). Knowledge and practices of ergonomics in computer users. *JPMA-Journal of the Pakistan Medical Association*, *62*(3), 213.
8. Pressman, A. (2021, January 11). *PC sales have surged for at-home workers and learners during the pandemic*. Fortune. Retrieved November 20, 2021, from

https://fortune.com/2021/01/11/covid-computer-sales-lenovo-hp-dell-apple/.

9. Thomsen, J. F., Gerr, F., & Atroshi, I. (2008). Carpal tunnel syndrome and the use of computer mouse and keyboard: a systematic review. *BMC musculoskeletal disorders*, *9*(1), 1-9.

10. Ali, K. M., & Sathiyasekaran, B. W. C. (2006). Computer professionals and carpal tunnel syndrome (CTS). *International Journal of Occupational Safety and Ergonomics*, *12*(3), 319-325.

11. Culjak, I., Abram, D., Pribanic, T., Dzapo, H., & Cifrek, M. (2012, May). A brief introduction to OpenCV. In *2012 proceedings of the 35th international convention MIPRO* (pp. 1725-1730).

12. Hecht, M. (2020, November 19). *Types of posture: How to correct bad posture*. Healthline. Retrieved November 26, 2021, from https://www.healthline.com/health/bone-health/the-4-main-types-of-posture#other-types.

13. Ryan, C. L., & Lewis, J. M. (2017). *Computer and internet use in the United States: 2015*. Washington, DC: US Department of Commerce, Economics and Statistics Administration, US Census Bureau.

14. Borhany, T., Shahid, E., Siddique, W. A., & Ali, H. (2018). Musculoskeletal problems in frequent computer and internet users. *Journal of family medicine and primary care*, *7*(2), 337.

15. Troy Bedinghaus, O. D. (2021, December 3). *An overview of eye strain (asthenopia)*. Verywell Health. Retrieved February 4, 2022, from https://www.verywellhealth.com/do-you-suffer-from-asthenopia-or-tired-eyes-3421982#:~:text=Causes%20of%20Eye%20Strain&text=Ciliary%20muscles%20change%20the%20shape,periods%20of%20using%20these%20muscles.

16. Chawla, A., Lim, T. C., Shikhare, S. N., Munk, P. L., & Peh, W. C. (2019). Computer vision syndrome: darkness under the shadow of light. *Canadian Association of Radiologists Journal*, *70*(1), 5-9.

17. Goplani, A., & Haral, P. (2018). Ergonomic awareness about use of electronic devices in IT professionals. *Int J Health Sci Res*, *8*, 62-70.

18. Ganne, P., Najeeb, S., Chaitanya, G., Sharma, A., & Krishnappa, N. C. (2021). Digital eye strain epidemic amid COVID-19 pandemic–a cross-sectional survey. *Ophthalmic epidemiology*, *28*(4), 285-292.

19. Tian, H., Wang, T., Liu, Y., Qiao, X., & Li, Y. (2020). Computer vision technology in agricultural automation—A review. *Information Processing in Agriculture*, *7*(1), 1-19.

20. Sharma, S. (2020, April 2). *A short guide to pose estimation in computer vision*. Medium. Retrieved February 21, 2022, from https://medium.com/@siddrrsh/a-short-guide-to-pose-estimation-in-computer-vision-3ea708dd9155

21. Alavi, A. (2020, October 3). *A review of Google's new mobile-friendly AI framework: Mediapipe*. Medium. Retrieved February 21, 2022, from https://medium.com/swlh/a-review-of-googles-new-mobile-friendly-ai-framework-mediapipe-25d62cd482a1#:~:text=MediaPipe%20is%20Fast&text=The%20first%20thing%20that%20stood,GPU%20acceleration%20and%20multi%2Dthreading.

22. Kulikajevas, A., Maskeliunas, R., & Damaševičius, R. (2021). Detection of sitting posture using hierarchical image composition and deep learning. *PeerJ computer science*, *7*, e442.

23. Saha, S. (2018, December 17). *A comprehensive guide to Convolutional Neural Networks‐the eli5 way*. Medium. Retrieved February 28, 2022, from https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53.

24. Chen, K. (2019, December). Sitting posture recognition based on OpenPose. In *IOP Conference Series: Materials Science and Engineering* (Vol. 677, No. 3, p. 032057). IOP Publishing.

# APPENDIX

| Project Activities | Week 1 | Week 2 | Week 3 | Week 4 | Week 5 | Week 6 | Week 7 | Week 8 | Week 9 | Week 10 | Week 11 | Week 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Selection of Project Topic | ■ | | | | | | | | | | | |
| Preliminary Research Work | | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| Performing Research on Workspace Ergonomics | | | ■ | ■ | ■ | | | | | | | |
| Performing Research on Postures | | | | | ■ | | | | | | | |
| Performing Research on Health Effects | | | | | ■ | | | | | | | |
| Performing Research on Computer Vision | | | ■ | ■ | ■ | | | | | | | |
| Performing Research on Proof of Concept | | | | ■ | ■ | | | | | | | |
| Submission of Progress Assessment 1 | | | | | | ■ | | | | | | |
| Proposal Defence | | | | | | | ■ | ■ | | | | |
| Conducting Requirements Gathering | | | | | | | | | | ■ | | |
| Submission of Interim Draft Report | | | | | | | | | | | ■ | |
| Submission of Progress Assessment 2 | | | | | | | | | | | | ■ |
| Submission of Interim Report | | | | | | | | | | | | ■ |

**Figure 5.1 Gantt Chart of Project Activities**

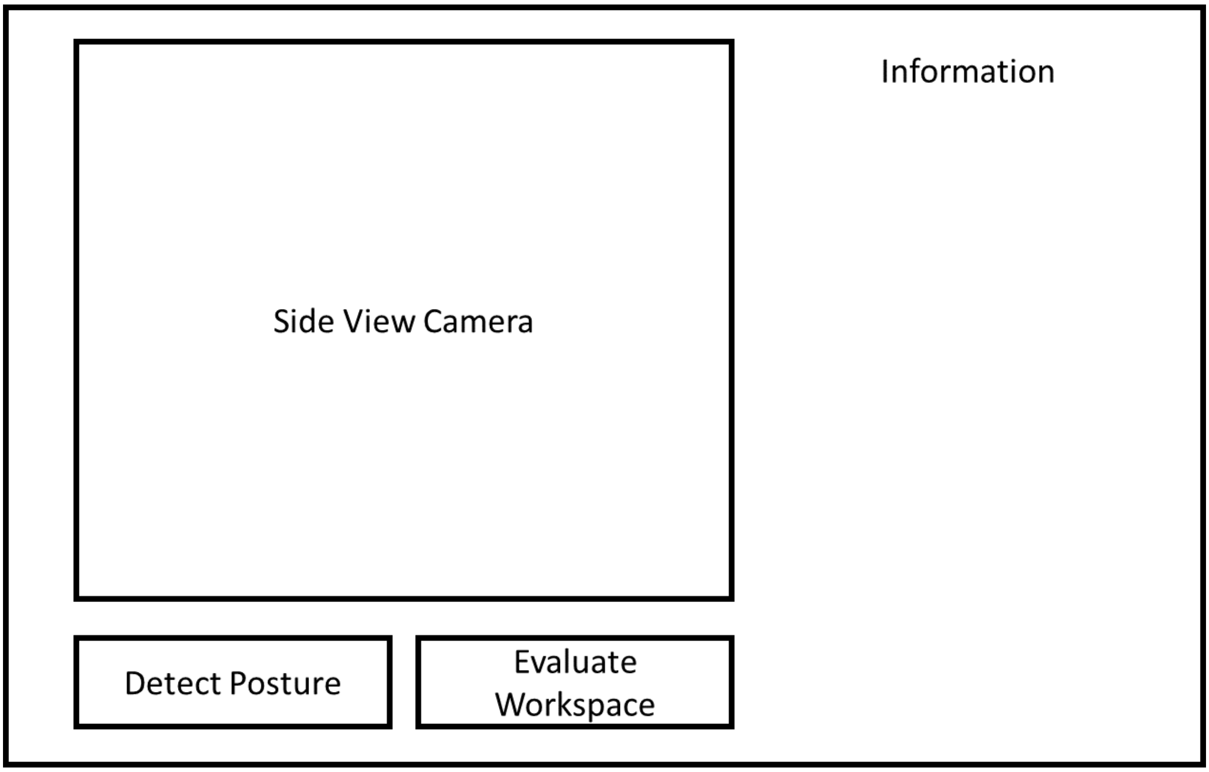



**Figure 5.2 Software Graphical User Interface**

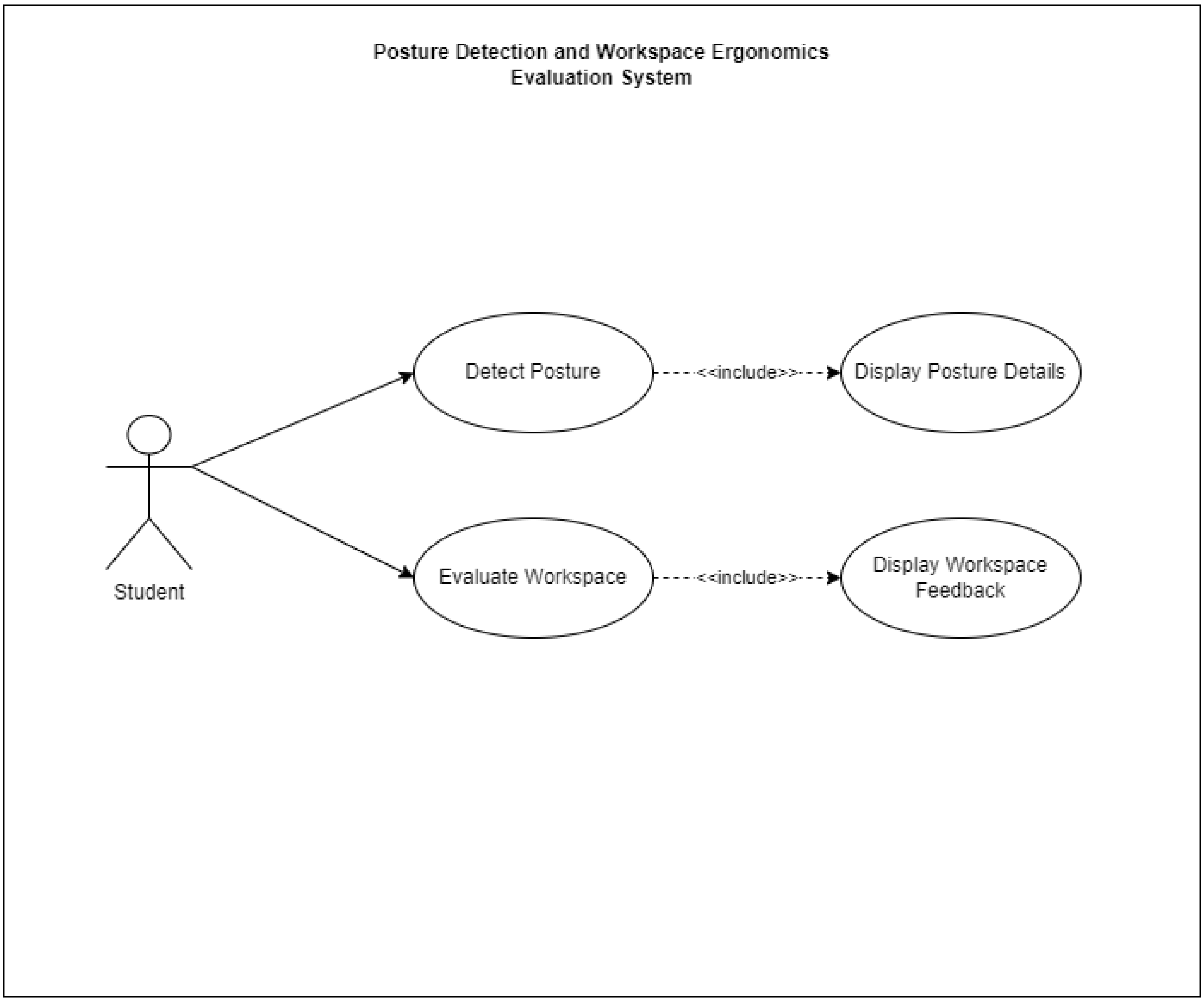



**Figure 5.3 UML Use Case Diagram of System**

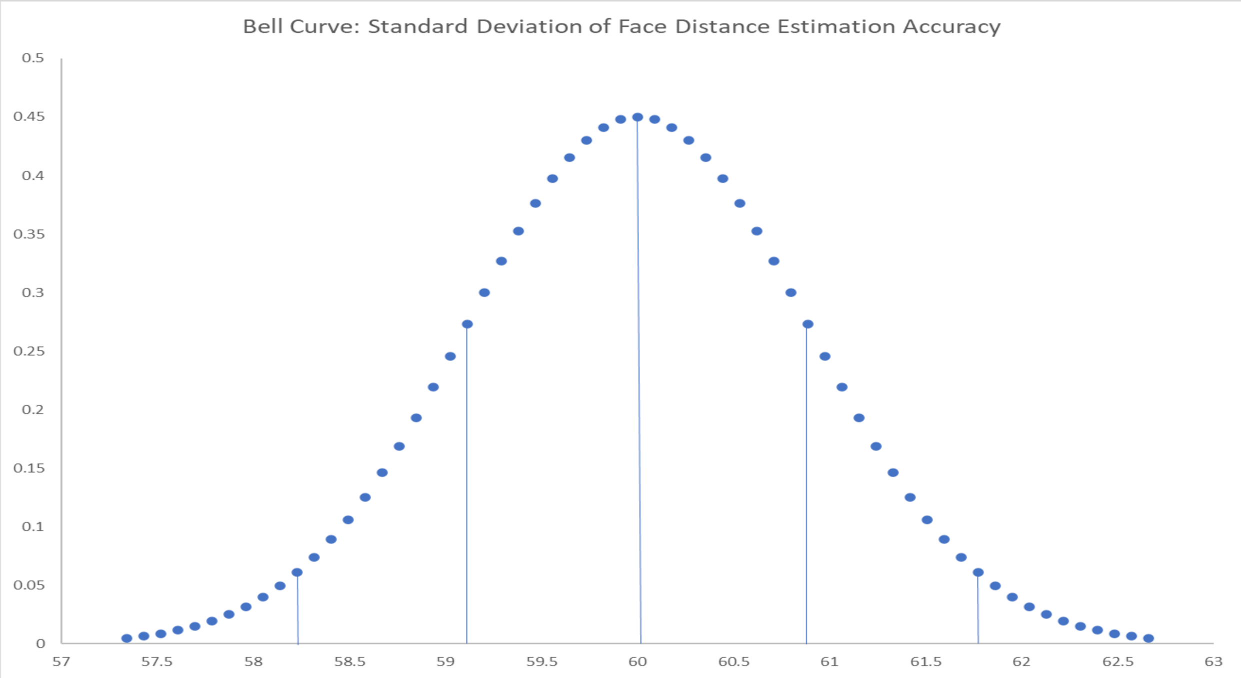
**Functional Requirements**

1. There are 2 buttons where the user can only interact which is the 'Detect Posture' and 'Evaluate Workspace' buttons.
2. When the 'Detect Posture' button is clicked, the system should be able to begin detecting the posture of the person seen from the camera sensor.
3. When the 'Evaluate Workspace' button is clicked, the system should be able to begin measuring the distance between the person to the monitor and identify the angle of orientation between the chair, table.
4. Results of either one of the configurations are shown under the 'information' section.
5. Software should be able to have access to a camera sensor in order to be functional.
6. The 'Side View Camera' area from the user interface will display the camera input in real time.

**Non-Functional Requirements**

1. System should be able to perform and react swiftly as user interacts with the system. Immediate effect should occur once any action is being triggered.
2. Performance of overall system should not exceed more than 3 seconds delay for any event.
3. Design of interface should be minimal but allows easy cognition.
4. Design must comply with the 10 usability heuristics, so users are able to interact with the software easily, effectively, and efficiently.

**Figure 5.4 List of Functional & Non-Functional Requirements**




```python
import sys
from PyQt5 import QtCore
from PyQt5.QtGui import *
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
import cv2
import mediapipe as mp
import numpy as np
import math
import time

start = time.time()
lstTimeElapsed = []

Black = (0, 0, 0)
Known_distance = 30  # Inches
Known_width = 5.7  # Inches

GREEN = (0, 255, 0)
RED = (0, 0, 255)
BLACK = (0, 0, 0)
YELLOW = (0, 255, 255)
WHITE = (255, 255, 255)
CYAN = (255, 255, 0)
MAGENTA = (255, 0, 242)
GOLDEN = (32, 218, 165)
LIGHT_BLUE = (255, 9, 2)
PURPLE = (128, 0, 128)
CHOCOLATE = (30, 105, 210)
PINK = (147, 20, 255)
ORANGE = (0, 69, 255)

fonts = cv2.FONT_HERSHEY_COMPLEX
fonts2 = cv2.FONT_HERSHEY_SCRIPT_SIMPLEX
fonts3 = cv2.FONT_HERSHEY_COMPLEX_SMALL
fonts4 = cv2.FONT_HERSHEY_TRIPLEX

class MainWindow(QWidget):
    def __init__(self):
        super(MainWindow, self).__init__()

        self.setWindowTitle("Posture Detection & Workspace Ergonomics Software")
        self.setGeometry(0,0,980,680)

        self.labelTitle = QLabel('Webcam Feed', self)
```

```
        self.labelTitle.setGeometry(QtCore.QRect(70, 80, 250, 50))
        self.labelTitle.setFont(QFont('Arial', 18))
        self.labelTitle.move(250, 10)

        self.label1 = QLabel('Information', self)
        self.label1.setGeometry(QtCore.QRect(70, 80, 250, 50))
        self.label1.setFont(QFont('Arial', 18))
        self.label1.move(760, 10)

        self.label2 = QLabel('Posture:', self)
        self.label2.setFont(QFont('Arial', 14))
        self.label2.move(670, 200)
        self.labelPosture = QLabel('Poor posture', self)
        self.labelPosture.setGeometry(QtCore.QRect(0, 0, 550, 25))
        self.labelPosture.setFont(QFont('Arial', 14))
        self.labelPosture.move(810, 200)

        self.label3 = QLabel('Angle:', self)
        self.label3.setFont(QFont('Arial', 14))
        self.label3.move(670, 250)
        self.labelAngle = QLabel('0', self)
        self.labelAngle.setGeometry(QtCore.QRect(0, 0, 250, 25))
        self.labelAngle.setFont(QFont('Arial', 14))
        self.labelAngle.move(810, 250)

        self.label4 = QLabel('Distance:', self)
        self.label4.setFont(QFont('Arial', 14))
        self.label4.move(670, 300)
        self.label4.setGeometry(QtCore.QRect(670, 300, 200, 25))
        self.labelTime = QLabel('Test', self)
        self.labelTime.setGeometry(QtCore.QRect(0, 0, 250, 25))
        self.labelTime.setFont(QFont('Arial', 14))
        self.labelTime.move(810, 300)

        self.VBL = QVBoxLayout()
        self.FeedLabel = QLabel()
        self.VBL.addWidget(self.FeedLabel)

        self.PostureBTN = QPushButton("Posture Estimation")
        self.PostureBTN.clicked.connect(self.BeginPostureEstimation)
        self.VBL.addWidget(self.PostureBTN)

        self.DistanceBTN = QPushButton("Distance Estimation")
        self.DistanceBTN.clicked.connect(self.BeginDistanceEstimation)
        self.VBL.addWidget(self.DistanceBTN)

        self.OrientationBTN = QPushButton("Angle Orientation")
        self.OrientationBTN.clicked.connect(self.BeginAngleOrientation)
        self.VBL.addWidget(self.OrientationBTN)

        self.Worker1 = Worker1()
        self.Worker2 = Worker2()
        self.Worker3 = Worker3()
        self.Worker1.start()
        self.Worker1.ImageUpdate.connect(self.ImageUpdateSlot)
        self.setLayout(self.VBL)

        self.my_timer = QtCore.QTimer()
        self.my_timer.timeout.connect(self.update_label)
        self.my_timer.start(10)

    def update_label(self):
        lblPosture = Worker1.worker1posture
        self.labelPosture.setText(str(lblPosture))

        lblAngle = Worker1.worker1angle
        self.labelAngle.setText(str(lblAngle) + u"\N{DEGREE SIGN}")

        lblDistance = Worker2.worker2Distance
        self.labelTime.setText(str(lblDistance) + "cm")

    def ImageUpdateSlot(self, Image):
        self.FeedLabel.setPixmap(QPixmap.fromImage(Image))

    def CancelFeed(self):
        self.Worker1.stop()
```

```
    def BeginPostureEstimation(self):
        self.Worker2.stop()
        self.Worker3.stop()
        self.Worker1.start()
        self.Worker1.ImageUpdate.connect(self.ImageUpdateSlot)

    def BeginDistanceEstimation(self):
        self.Worker1.stop()
        self.Worker3.stop()
        self.Worker2.start()
        self.Worker2.ImageUpdate.connect(self.ImageUpdateSlot)
        # self.my_timer.stop()

        # self.updateLabel2()

    def BeginAngleOrientation(self):
        self.Worker1.stop()
        self.Worker2.stop()
        self.Worker3.start()
        self.Worker3.ImageUpdate.connect(self.ImageUpdateSlot)

    # def updateLabel2(self):
    #     lblDistance = Worker2.worker2Distance
    #     self.labelTime.setText(str(lblDistance))



class Worker1(QThread):
    ImageUpdate = pyqtSignal(QImage)
    worker1posture = ''
    worker1angle = ''

    def run(self):

        def findAngle(frame, p1, p2, p3):

            if results.pose_landmarks:

                x1, y1 = poseList[p1][1:]
                x2, y2 = poseList[p2][1:]
                x3, y3 = poseList[p3][1:]

                angle = math.degrees(math.atan2(y3 - y2, x3 - x2) - math.atan2(y1 - y2, x1 -
x2))

                if angle < 0:
                    angle += 360

                angle = abs(angle - 360)

                cv2.line(frame, (x1, y1), (x2, y2), WHITE, 3)
                cv2.line(frame, (x2, y2), (x3, y3), WHITE, 3)
                cv2.circle(frame, (x1, y1), 10, (255, 0, 0), cv2.FILLED)
                cv2.circle(frame, (x1, y1), 15, (255, 0, 0), 2)
                cv2.circle(frame, (x2, y2), 10, (255, 0, 0), cv2.FILLED)
                cv2.circle(frame, (x2, y2), 15, (255, 0, 0), 2)
                cv2.circle(frame, (x3, y3), 10, (255, 0, 0), cv2.FILLED)
                cv2.circle(frame, (x3, y3), 15, (255, 0, 0), 2)
                cv2.putText(frame, str(int(angle)), (x2 - 20, y2 + 50),
cv2.FONT_HERSHEY_PLAIN, 2, (255, 0, 255), 2)



            return angle

        self.ThreadActive = True
        Capture = cv2.VideoCapture(0)

        mpDraw = mp.solutions.drawing_utils
        mpPose = mp.solutions.pose
        pose = mpPose.Pose()
        posture = 'undefined'


        while self.ThreadActive:
            ret, frame = Capture.read()
            if ret:
```

```
                Image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
                results = pose.process(Image)
                poseList = []

                # To calculate the current time elapsed. Each iteration will display the
posture.
                currentTime = time.time()
                timeElapsed = currentTime - start
                lstTimeElapsed = timeElapsed

                if results.pose_landmarks:
                    mpDraw.draw_landmarks(Image, results.pose_landmarks,
mpPose.POSE_CONNECTIONS)

                    for id, lm in enumerate(results.pose_landmarks.landmark):
                        h, w, c = Image.shape
                        cx, cy = int(lm.x * w), int(lm.y * h)
                        poseList.append([id, cx, cy])
                    cv2.circle(Image, (cx, cy), 15, (255, 0, 0), cv2.FILLED)

                if len(poseList) != 0:
                    angle = findAngle(Image, 11, 23, 25)

                    if 95 <= angle <= 120:
                        posture = 'Optimal posture'
                    else:
                        posture = 'Poor posture'

                    cv2.putText(Image, str(posture), (70, 50), cv2.FONT_HERSHEY_PLAIN, 3,
(255, 0, 0), 3)

                lstTimeElapsed = "{:.2f}".format(lstTimeElapsed)
                print(lstTimeElapsed, posture, poseList)

                Worker1.worker1posture = posture
                Worker1.worker1angle = int(angle)

                ConvertToQtFormat = QImage(Image.data, Image.shape[1], Image.shape[0],
QImage.Format_RGB888)
                Pic = ConvertToQtFormat.scaled(640, 480, Qt.KeepAspectRatio)
                self.ImageUpdate.emit(Pic)


    def stop(self):
        self.ThreadActive = False
        self.quit()

class Worker2(QThread):
    ImageUpdate = pyqtSignal(QImage)
    worker2Distance = ''

    def run(self):

        self.ThreadActive = True
        Capture = cv2.VideoCapture(0)

        Distance_level = 0
        face_detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")

        def FocalLength(measured_distance, real_width, width_in_rf_image):
            focal_length = (width_in_rf_image * measured_distance) / real_width
            return focal_length

        def Distance_finder(Focal_Length, real_face_width, face_width_in_frame):
            distance = (real_face_width * Focal_Length) / face_width_in_frame
            return distance

        def face_data(Image, Distance_level):
            face_width = 0
            face_center_x = 0
            face_center_y = 0
            gray_image = cv2.cvtColor(Image, cv2.COLOR_BGR2GRAY)
            faces = face_detector.detectMultiScale(gray_image, 1.3, 5)
            for (x, y, w, h) in faces:
                LLV = int(h * 0.12)

                cv2.line(Image, (x, y + LLV), (x + w, y + LLV), WHITE, 1)
```

```
                cv2.line(Image, (x, y + h), (x + w, y + h), WHITE, 1)
                cv2.line(Image, (x, y + LLV), (x, y + LLV + LLV), WHITE, 1)
                cv2.line(Image, (x + w, y + LLV), (x + w, y + LLV + LLV),WHITE, 1)
                cv2.line(Image, (x, y + h), (x, y + h - LLV), WHITE, 1)
                cv2.line(Image, (x + w, y + h), (x + w, y + h - LLV), WHITE, 1)

                face_width = w
                face_center = []
                # Drwaing circle at the center of the face
                face_center_x = int(w / 2) + x
                face_center_y = int(h / 2) + y
                if Distance_level < 10:
                    Distance_level = 10

            return face_width, faces, face_center_x, face_center_y

        ref_image = cv2.imread("Ref_image1.jpg")
        ref_image_face_width, _, _, _ = face_data(ref_image, Distance_level)
        Focal_length_found = FocalLength(Known_distance, Known_width, ref_image_face_width)

        while self.ThreadActive:
            ret, frame = Capture.read()
            if ret:
                Image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
                face_width_in_frame, Faces, FC_X, FC_Y = face_data(Image, Distance_level)
                # finding the distance by calling function Distance finder
                for (face_x, face_y, face_w, face_h) in Faces:
                    if face_width_in_frame != 0:
                        Distance = Distance_finder(Focal_length_found, Known_width,
face_width_in_frame)
                        Distance -= 6
                        Distance *= 2.54
                        Distance = round(Distance, 2)
                        # Drwaing Text on the screen
                        Distance_level = int(Distance)

                        if Distance >= 50:
                            txtDistance = 'Optimal'
                        else:
                            txtDistance = 'Poor'

                        cv2.putText(Image, f"Distance {Distance} cm", (face_x - 6, face_y -
6), cv2.FONT_HERSHEY_PLAIN, 0.8, (WHITE), 1)
                        cv2.putText(Image, f"{txtDistance} distance", (50, 50),
cv2.FONT_HERSHEY_COMPLEX, 1.5, (WHITE), 1)

                ConvertToQtFormat = QImage(Image.data, Image.shape[1], Image.shape[0],
QImage.Format_RGB888)
                Pic = ConvertToQtFormat.scaled(640, 480, Qt.KeepAspectRatio)
                self.ImageUpdate.emit(Pic)

            Worker2.worker2Distance = Distance

    def stop(self):
        self.ThreadActive = False
        self.quit()

class Worker3(QThread):
    ImageUpdate = pyqtSignal(QImage)

    def run(self):

        self.ThreadActive = True
        Capture = cv2.VideoCapture(0)

        while self.ThreadActive:
            ret, img = Capture.read()
            if ret:
                gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
                _, bw = cv2.threshold(gray, 50, 255, cv2.THRESH_BINARY | cv2.THRESH_OTSU)
                contours, _ = cv2.findContours(bw, cv2.RETR_LIST, cv2.CHAIN_APPROX_NONE)

                classNames = []
                classFile = 'coco.names'
                with open(classFile, 'rt') as f:
                    classNames = f.read().rstrip('\n').split('\n')
```

```
                configPath = 'ssd_mobilenet_v3_large_coco_2020_01_14.pbtxt'
                weightsPath = 'frozen_inference_graph.pb'

                net = cv2.dnn_DetectionModel(weightsPath, configPath)
                net.setInputSize(320, 320)
                net.setInputScale(1.0 / 127.5)
                net.setInputMean((127.5, 127.5, 127.5))
                net.setInputSwapRB(True)

                classIds, confs, bbox = net.detect(img, confThreshold=0.5)
                print(classIds, bbox)

                for classId, confidence, box in zip(classIds.flatten(), confs.flatten(),
bbox):
                    cv2.putText(img, classNames[classId - 1], (box[0] + 10, box[1] + 30),
cv2.FONT_HERSHEY_SIMPLEX, 1,
                               WHITE, 2)
                    cv2.putText(img, "Conf:" + str(round(confidence * 100, 2)), (box[0] + 10,
box[1] + 50),
                               cv2.FONT_HERSHEY_SIMPLEX, 0.5,
                               WHITE, 1)

                for i, c in enumerate(contours):

                    # Calculate the area of each contour
                    area = cv2.contourArea(c)

                    # Ignore contours that are too small or too large
                    if area < 3700 or 100000 < area:
                        continue

                    rect = cv2.minAreaRect(c)
                    box = cv2.boxPoints(rect)
                    box = np.int0(box)

                    # Retrieve the key parameters of the rotated bounding box
                    center = (int(rect[0][0]), int(rect[0][1]))
                    width = int(rect[1][0])
                    height = int(rect[1][1])
                    angle = int(rect[2])

                    if width < height:
                        angle = abs(90 - angle)
                    else:
                        angle = abs(-angle)

                    label = "  Rotation Angle: " + str(angle) + " degrees"
                    # cv.rectangle(img, (center[0] - 35, center[1] - 25), (center[0] + 200,
center[1] + 10), (255, 255, 255), -1)
                    cv2.putText(img, label, (center[0] - 120, center[1]),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, WHITE, 1,
                               cv2.LINE_AA)
                    cv2.drawContours(img, [box], 0, (0, 0, 255), 2)

                ConvertToQtFormat = QImage(img.data, img.shape[1], img.shape[0],
QImage.Format_RGB888)
                Pic = ConvertToQtFormat.scaled(640, 480, Qt.KeepAspectRatio)
                self.ImageUpdate.emit(Pic)

    def stop(self):
        self.ThreadActive = False
        self.quit()

if __name__ == "__main__":
    App = QApplication(sys.argv)
    Root = MainWindow()
    Root.show()
    sys.exit(App.exec())
```