

School of Built Environment, Engineering and Computing

|  |  |
| --- | --- |
| Student ID | 77389273 |
| Student Name | Yatharth Mehta |
| Module Name & CRN | COMP737 - Masters Final Project/Dissertation - 18684 - LTP - 202410 |
| Level | 7 |
| Assessment Name & Part No. | Dissertation |
| Project Title | AI Driven Work From Home Assistant |
| Date of Submission | 6th May, 2025 |
| Course | M.Eng Computer Science |
| Academic Year | 2024-25 |

AI Driven Work From Home Assistant

Table of Contents

[1) Introduction 5](#_Toc197482006)

[2) Aim and Objectives 5](#_Toc197482007)

[3) Literature Review 5](#_Toc197482008)

[(A) The Trend of Work From Home Sector 5](#_Toc197482009)

[(B) Challenges regarding the Work From Home Environment 6](#_Toc197482010)

[(C) Current technologies assisting Remote Working 7](#_Toc197482011)

[(D) Shortcomings of the assisting technologies 7](#_Toc197482012)

[4) Methodology 8](#_Toc197482013)

[(A) Why Python ? 9](#_Toc197482014)

[(B) Libraries Utilized 9](#_Toc197482015)

[(C) Working 10](#_Toc197482016)

[I) Posture Recognition System 10](#_Toc197482017)

[(a) WHY ? (Introduction) 10](#_Toc197482018)

[(b) HOW ? (Working) 10](#_Toc197482019)

[(c) Flowchart 10](#_Toc197482020)

[II) Distraction Tracking System 11](#_Toc197482021)

[(a) WHY ? (Introduction) 11](#_Toc197482022)

[(b) HOW ? (Working) 11](#_Toc197482023)

[(c) Flowchart 11](#_Toc197482024)

[III) Background Noise Suppression System 12](#_Toc197482025)

[(a) WHY ? (Introduction) 12](#_Toc197482026)

[(b) HOW ? (Working) 12](#_Toc197482027)

[(c) Flowchart 12](#_Toc197482028)

[5) Evaluation 12](#_Toc197482029)

[(A) Metrics 13](#_Toc197482030)

[(B) Distraction Tracking (Eye Tracking) Evaluation 14](#_Toc197482031)

[I) Comparative Analysis 14](#_Toc197482032)

[(a) Tabular Data of 5 Runs 14](#_Toc197482033)

[(b) Final Average Data 15](#_Toc197482034)

[II) Advantages over the compared code 16](#_Toc197482035)

[III) Technical Rationale of Differences 16](#_Toc197482036)

[IV) Graphical Representation 17](#_Toc197482037)

[V) Multi-environmental testing 19](#_Toc197482038)

[(a) Why ? 19](#_Toc197482039)

[(b) How ? 19](#_Toc197482040)

[(c) Why CPU Load ? 19](#_Toc197482041)

[(d) Results 19](#_Toc197482042)

[(i) Tabular Data 19](#_Toc197482043)

[10% CPU Load Data 20](#_Toc197482044)

[45% CPU Load Data 21](#_Toc197482045)

[90% CPU Load Data 22](#_Toc197482046)

[(ii) Graphical Representation 24](#_Toc197482047)

[Accuracy 24](#_Toc197482048)

[Latency 25](#_Toc197482049)

[Memory Usage 26](#_Toc197482050)

[Security 27](#_Toc197482051)

[Coding Practice 28](#_Toc197482052)

[Summary 28](#_Toc197482053)

[(C) Posture Recognition Evaluation 28](#_Toc197482054)

[I) Comparative Analysis 29](#_Toc197482055)

[II) Advantages over the compared code 29](#_Toc197482056)

[III) Technical Rationale of Differences 29](#_Toc197482057)

[IV) Graphical Representation 31](#_Toc197482058)

[(D) Background Noise Reduction 33](#_Toc197482059)

[I) Comparative Analysis 33](#_Toc197482060)

[II) Advantages over the compared code 34](#_Toc197482061)

[III) Technical Rationale of Differences 34](#_Toc197482062)

[IV) Graphical Representation 35](#_Toc197482063)

[6) Potential Enhancements 37](#_Toc197482064)

[7) Ethical Considerations 38](#_Toc197482065)

[8) Conclusion 38](#_Toc197482066)

[9) References 38](#_Toc197482067)

[10) Appendix 40](#_Toc197482068)

[(A) Posture Recognition Cod 40](#_Toc197482069)

[(B) Distraction Tracking Code 42](#_Toc197482070)

[(C) Background Noise Suppression Code 45](#_Toc197482071)

# 1) Introduction

Work from Home is a new trend nowadays in the corporate world. The number of employees opting for work from home is increasing day by day as work from home environment provides the comfort , the facilities and peace of one’s home yet getting the same amount of remuneration and workload. Even being so popular and highly adapted , the Work From Home method is not perfect and still needs upgradation for betterment as all the professional standards and environments do. As beneficial as remote working is in today’s work cycle, there are a few drawbacks like distraction problems, health problems and many more. Also Artificial Intelligence is now ruling the development industry. There are hardly any sectors that does not use AI for it’s betterment. Hence the new and needy for development sector like Working From Home or Remote Working can utilize the help and functionalities that AI has to offer. Thus , this report will discuss about an AI driven Work from home Assistant that can be used to increase the productivity of the working professionals.

# 2) Aim and Objectives

Aim :

To Develop a software that helps and assists in a work-from-home environment and enhances the user experience and productivity.

Objectives :

1. To conduct a thorough literature review about the Work from home environment and the existing assistance systems.
2. To critically analyse the functionalities like posture monitoring , background noise suppression and distraction monitoring that are to be added to achieve the aim.
3. To develop the system using appropriate methods and tools.
4. To Evaluate the system by comparing the current existing tools with the developed tools.
5. To provide future potential upgradations in this system.

# 3) Literature Review

## (A) The Trend of Work From Home Sector

Although remote working or work from home was existing the job market since quite a time but it gained it’s highest popularity in the time of Covid-19 pandemic. Due to the extensive duration of Lockdowns the industry and job market had to adapt remote working more rapidly and widely. In 2019 , around 10% percent of the employees worked remotely fully, while 20% would work part-time but in the pandemic nearly 100% of the people , those who could, worked fully remotely (Faulds, D.J. and Raju, P.S., 2021). Due to the flexibility and ease to work nature more and more people started adapting it but as the trend got the boom rapidly for the management this was quiet a hassle. Adding to that Galanti, T.M. et al. (2021) says that in order for employers to successfully support employees' productivity and guarantee that they have a better work-life balance, adopting this flexible method of working has typically been portrayed as a deliberate decision that involves a time of planning, preparation, and adaption. 4–6 but, the COVID-19 pandemic has significantly compelled the majority of businesses to embrace this mode of operation, sometimes without equipping staff members with the skills needed for remote work.

That period of necessary remote working made the Work From Home so popular and comfortable that the people became a kind of fan of it. It was the first time for the people that they could have a comfort zone while being in a workplace. Hence , the amount of people who still choose and prefer working from home in almost every sector has been monumental till this date post the Covid Pandemic. Kong, X. et al. (2022) did an extensive data analytical study and got the conclusion that being compelled to work from home during the COVID-19 pandemic has a beneficial impact on the decision to work from home following the epidemic. In a journal article after doing highly mathematical and detailed study of the work from home adaptation and substitution post Covid, Bick, A. et al. (2023) confirms that while WFH adoption includes possible increases in worker wellbeing and reduced labor costs for businesses, WFH substitution entails poorer worker welfare and higher production costs for enterprises.

## (B) Challenges regarding the Work From Home Environment

In spite of being highly adapted the work from home environment has some challenges of it’s own. Primarily there comes the managing challenges. As in the remote working the monitoring of work and managing the employees gets a little trickier and harder than the on-site management. Lack of face-to-face contacts can cause misunderstandings, lower team morale, and difficulty tracking project progress (Chitta, S. et al., 2021). In addition to that increased amount of screentime , continuous sitting and increased exposure of radiations from constant use of laptop and mobile phones had adverse negative impacts on the health of the working professionals. Ugemuge, P.T. et al. (2022) states that although research has examined the effects of working from home on a number of criteria, including quality of life, health, safety, and productivity, prolonged periods of sedentary labor, particularly when done with poor posture, increase the risk of cervical and lower back discomfort. Putting more light on the health affected due to work from home Arlinghaus, A. and Nachreiner, F. (2014) says that increased use of information and communication technologies, like computers and smartphones, allows for greater temporal and spatial flexibility in the workplace, which may result in more supplemental work and that is, working beyond contractually agreed-upon hours or being available constantly which further extends work hours and results in work hours on the weekends and in the evenings, interfering with social and biological rhythms for social interaction, sleep, and recuperation.

Apart from management and health there raised another problem in the work from home environment and that is lack of concentration or increasing in distraction. Since their coworkers are not bothering them, people who work from home are often anticipated to encounter less distractions than those who work in an office, however if a homemaker shares their workstation with other family members, the number of distractions and mental health problems may actually rise (Bergefurt, L. et al., 2022). Related to distractions and mental health Xiao, Y. et al. (2021) states that since working parents were spending more time at home with their children, it makes sense that their physical and mental health was improving. However, the work-life strain brought on by more distractions and a lack of support from daycare facilities or babysitters during working hours is probably the cause of the concurrent rise in new physical and mental problems.

## (C) Current technologies assisting Remote Working

Other than WhatsApp and other social medias, there are some existing technologies or software that are making the remote working easier for the professionals. Some of the best examples are Google Meet , Zoom and Microsoft Teams. The COVID-19 epidemic and the subsequent stay-at-home orders have resulted in substantial changes in the way people work and one of these developments is the rising usage of video conferencing as a way of communication or conducting work meetings (Karl, K. A. et al., 2022). Except these video conferencing software there are some more technology that assists in remote working environment. Discord was used prominently for the communication and managing the interaction between managers and employees and also between employees to employees. Although discord was more used by the managers to create a friendly environment between the employees for better non-work related interactions. The User Experience Questionnaire (UEQ) application revealed that Discord excels in non-task-related features and offers an above-average UX (Mora-Jimenez et al., 2022).

## (D) Shortcomings of the assisting technologies

Though existing technologies help a lot in the remote working environment but there are certain gaps or shortcomings that it can not fulfil. The technologies are purely focused on managerial perspective or connecting people through chatting platforms or video conferencing but there are no major softwares or applications that takes the following in account of the professional while working from home :

1. Body Posture : As people have to sit in a desired locations for hours on end.
2. Distraction : As environment of a home can be distracting and stimulating.
3. Background noise : As controlling the house environmental factors are impossible.

Butte, K.T. et al. (2023) states that working remotely can lead to bad posture and lower back pain (LBP) due to increased sitting duration and less suitable ergonomics. There are no technologies that are available easily in the market that can help or assist to this problem for the remotely working professionals. Above that there are no tech that exists that can calculate the amount of time the person was distracted from the work. Another factor that any video conferencing or other softwares don’t measure is fatigue. Working from home gives a person a mental confidence that he/she is working from the best setup possible that is home and that gives a sense of relief and comfort and due to that people can sometime overdo the work and overdone work with a bad or inappropriate posture can result into an increasing level of fatigueness in their body , leading to health deterioration. Also with this abrupt transition, workers and the media began debating videoconference weariness, a potentially new phenomena of feeling fatigued and drained due to a videoconference (Bennett, A.A. et al., 2021). On top of that indoor noises are a major problem to a professional working from home. It can disrupt the professional flow of a meeting or an online conference. It was discovered that greater indoor noise disturbance during the pandemic era led to higher degrees of noise annoyance, implying that the degree of noise disruption is closely linked to persons' experience of noise annoyance, which can impair their job satisfaction (Park, Shin and Kim, 2023). Luebstorf et al. (2023) addresses the issue of the background noises by stating that people were having difficulty maintaining focused and being distracted by their pets, neighbours, or background noises. In some situations, background noises were created by employees' families, resulting in having issues communicating with coworkers owing to missing social cues, a lack of fast response, and less flexibility contributed to videoconferences feeling less natural. Thus , the work from home environment can use some assistance to fill these gaps.

# 4) Methodology

So in this report a software to fill the above mentioned gaps is developed. The software is primarily focused on assisting the personal in work from home environment. The software is basically divided into 3 functionalities. 1) Posture Recognition System 2) Distraction Tracking System 3) Background Noise Suppression System. All of the functionalities are developed in Python coding language as it is a high level language and largely adaptable.

## (A) Why Python ?

Python is chosen for the coding of the systems in this software for multiple beneficial reasons. The reasons and advantages to choosing python are as follows :

* High-Level Language
* Easy to code
* Easy to adapt
* Provides huge range of Libraries to work with
* Easy to test and compare
* More advanced than other high level languages in terms of simplicity and flexibility
* Comparatively more versatile
* Requires bare minimum changes for Cross Platform workings
* Works best in case of AI and ML development

Hence , python is the selected language and for the IDE , “Visual Studio Code” is used as it is more suitable , dynamic and has so many helping features for the person coding in that IDE.

## (B) Libraries Utilized

As python language provides a vast community of libraries to help with , a handful of them are used to develop the abovesaid functionalities. Below is the list of the libraries that are utilized in this project :

|  |  |  |
| --- | --- | --- |
| Library | Functionality | Used for |
| OpenCV (cv2) | Image and video processing and manipulation | Face Detection, Object Tracking and Camera Capture |
| Mediapipe | Customizable Machine Learning cross-platform solutions | Face , Hand and Pose Detection in Real time |
| Ctypes | To interact C compatible shared libraries | For alert pop-ups |
| Time | Perform time related tasks | Measure the time taken for an activity |
| Librosa | Audio analysis and operations | Background noise suppression |
| Soundfile | Reads and Writes Sound files | Importing and Exporting the music file |

Table 1.1 – Python Libraries Utilized

## (C) Working

### I) Posture Recognition System

#### (a) WHY ? (Introduction)

This system as the name suggests is specifically designed to recognize the posture of the user sitting in front of the camera in the work from home settings. As the personnel in the work from home environment has to stick to the desk for the majority of their work. Hence this system will make sure to assist them in maintaining a proper posture while working so the risk of spine and back problems can be reduced drastically.

#### (b) HOW ? (Working)

For the working , with a use of python library , the webcam or camera of the users computer will be captured and on that video stream , a Machine Learning model is implemented to check the alignment and distance and other particulars of the shoulders , ears and other required landmarks on the body. On that raw data , conditions and thresholds are implemented in order to calculate the results of the posture of the user. The outcome will have 2 results , whether the user has good posture or bad posture (slouching or anything else). Now, it is understandable that real-time analysis can be rigid at times so , the alert pop up is not designed to be thrown at real time but the AI will hold on some given time limit and check whether the posture has been bad over that time limit that has been pre-set. If that box is ticked then only a pop-up will be shown to the user alerting him/her that the posture has been bad.

In order to perform these tasks 2 main libraries are used in the coding section. CV2 and Media pipe are used for Video Capturing and for landmarking the body parts. Ctypes and Time is also used for alerts and time based calculations.

#### (c) Flowchart

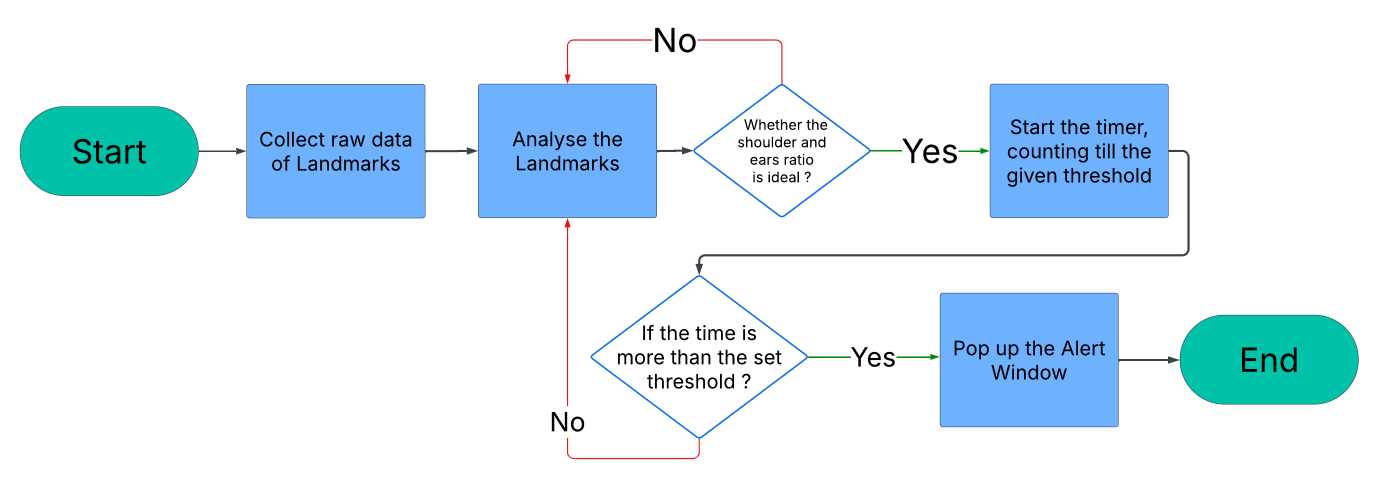


Image 1.1 – Flow chart of the posture recognition system

### II) Distraction Tracking System

#### (a) WHY ? (Introduction)

Now working from as comfortable setting as literally home , the chances of getting distracted by either chores or merely sliding into that comfort zone of not doing anything and just relaxing on the chair or even using the mobile phone to blow some steam off is gargantuan. This functionality is build to assist the user to be attentive to work and remind or alert the user of him/her being distracted and to continue focusing on the work they are supposed to do rather than getting engaged in the other activities.

#### (b) HOW ? (Working)

So practically , like the posture monitoring system , this system also requires the webcam or the camera of the user. Likewise a Machine Learning model is implemented but this time the landmarks of the whole body is not taken into consideration but the eyes , particularly iris is in the main focus. The system constantly monitors the eyes (iris) of the user and based on that a condition is checked that whether the eyes are facing the screen or not. If the eyes are getting off the screen and is looking at any other directions an assumed decision is made that the user is distracted from the work. Now again like mentioned previously , the real time alerts and decisions can be a little rigid and inflexible. Thus , a specific threshold is also set for the alert to be popped. The system waits for a certain time getting thorough that the user is in fact distracted and when the system is perfectly sure i.e. if the eyes are off the screen for the time that is above the threshold, then and only then an alert will be popped on the screen notifying the personnel that he/she is distracted.

The libraries used for this system are OpenCV (CV2) and media pipe for capturing the webcam and locating the eyes (iris) landmarks. The library ctypes is used to integrate an alert popup and Time library is used for time related calculations.

#### (c) Flowchart

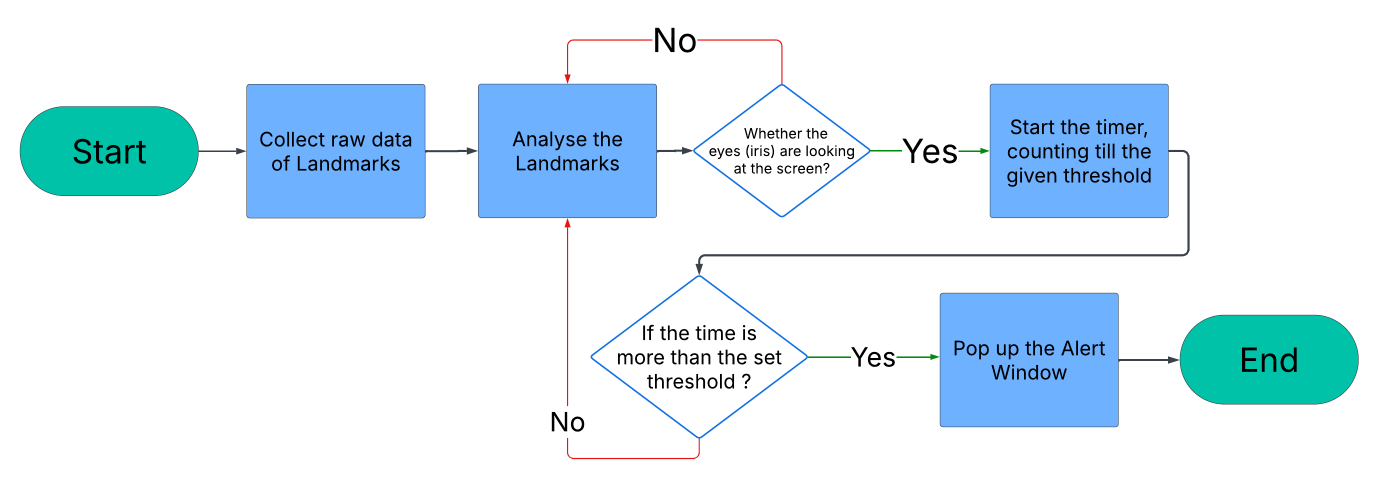


Image 1.2 – Flow chart of the distraction tracking system

### III) Background Noise Suppression System

#### (a) WHY ? (Introduction)

Remote working has introduced a trend of pre-recorded meetings in which any meetings can be recorded and can be further used by the employees to refer back to. In this scenario , a meeting or conference acts like a reference point for employees that they can refer to or revisit for clarity rather than being just a one-time communication event , which increases the functional importance of a meeting or a conference. The on-site offices have meeting rooms , conference rooms or cubicles to maintain the decorum and silence for professionals to communicate properly and clearly , while the level of that peace and decorum can not be promised when a person is conducting a conference or parting important details in his/her won home i.e working from home environment. Hence this Noise suppression system provides the functionality to remove or erase the background noise significantly resulting onto clearing the vocals to make sure that voices and the information parted in the recorded meeting or conference is smooth and clear.

#### (b) HOW ? (Working)

The user has to pass the raw audio file to the system and then using multiple python libraries the system works on the raw file , analyzing it and then removing all the background noises and disturbance like the sound of a fan , barking of dogs , crying of a child , sound of a doorbell and many more up to a significant and notable level. After cleansing the audio file , the system then exports the cleaned audio , the refined audio to specified folder.

For the libraries , librosa is used for the cleaning and refining the audio and soundfile library is used for the file, import export operations of the file.

#### (c) Flowchart

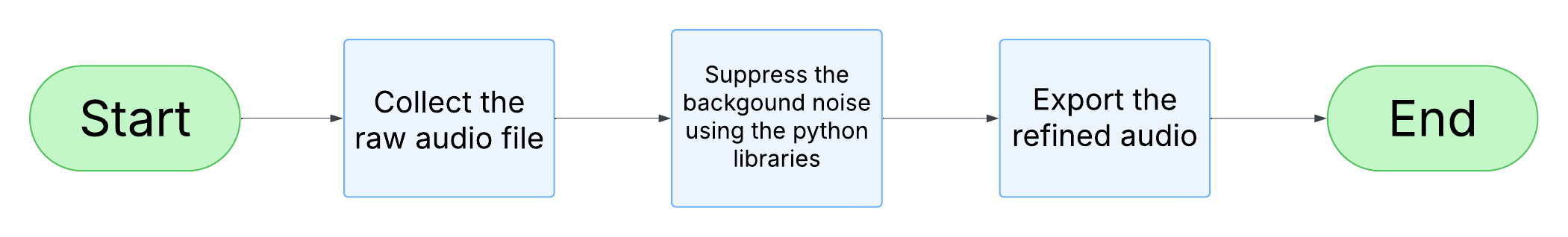


Image 1.3 – Flow chart of the background noise suppression system

# 5) Evaluation

Evaluation of the software is done via 2 methods primarily traditional tools are used to evaluate the Distraction tracking functionality. In that various python libraries are used to compare and evaluate the results and displaying them in percentage values. For the rest of the 2 facilities AI tools are being used to evaluate more smartly and by inculcating the use of AI tools gives a change of evaluation algorithm than using the traditional libraries on all of the functionalities. Although , regardless the evaluation algorithm , the metrics are same for every computation.

## (A) Metrics

The evaluation is conducted on various metrics. These are the metrics that can be used on all of the 3 developed functionalities those metrics are :

*Accuracy*

Accuracy is one of the primary factor when evaluating any code. This metrics is measured to see how accurate the code is functionalizing i.e how well the logics that are embedded in the code are executing when run. To calculate the accuracy , True Positive and False Positives are also calculated and then the :

**Accuracy = (TP/TP+FP) ​× 100**

Is implemented to find the percentage value of the Accuracy.

*Speed (Time)*

The speed metric will evaluate how quick and responsive the code is when ran. It will evaluate the time taken into Milliseconds. This factor majorly depends on the IDE and the system that the code is being ran on.

*Security*

Security is an important aspect of any code and if there are facilities like utilizing the webcam, like there are in this software , it becomes integral that it is not misused or is not much vulnerable.

*Memory Usage*

For any software , the memory usage is extremely important. The allocation and consumption of memory resources has to be kept as low as possible in order for the computer to run multiple applications smoothly.

*Coding Practices*

Even though the logic of a coding block or structure might be same but an important aspect is also to code it properly , understandably and also by using the best alternatives and practices to achieve the same goal.

## (B) Distraction Tracking (Eye Tracking) Evaluation

For the eye tracking evaluation traditional evaluating tools are used like python libraries , in-built functions and many more. Python has a vast range of libraries that are veery useful when it comes to evaluating the metrics like speed , memory usage etc. The libraries used to test the metrics are :

|  |  |  |
| --- | --- | --- |
| **Library** | **Functionality** | **Evaluated Metrics** |
| Mediapipe | Detects facial landmarks | Accuracy |
| Time | Measuring the durations | Latency/Speed |
| Bandit | Measure Security parameters | Security |
| psutil | Detects the memory used for a given process | Memory Usage |
| inspect | Detects the coding structures | Coding Practices |

Table 2.1 – Python Libraries Utilized for testing metrics

### I) Comparative Analysis

Below is the quantitative comparative analysis of the eye tracking code used in this project and another open source eye tracking code. That code is from a website named “GeeksforGeeks”. As keeping the limited time for research and the dynamics of this project in the mind this was the one of the best options to compare the code with. Provided with additional time , further exploration could have yielded a more refined alternative. Accuracy is measured by the logic of face detection , so for that both codes are tested under common situations that is 10 seconds of the face present on the screen and 10 seconds of face going out of the screen and then again 10 seconds of the face being present. As traditional non-AI methods are used for this evaluation , it is unfair to utilize data from a single run , hence both the codes are run 5 times each and the average data is used and compared for evaluation.

#### (a) Tabular Data of 5 Runs

First Run:

|  |  |  |
| --- | --- | --- |
| Metric | Project Code | Compared Code |
| Accuracy | 93.24 % | 55.79 % |
| Latency/ Speed | 0.0265 seconds/frame | 0.0599 seconds/frame |
| Security | 70 % | 70 % |
| Memory Usage | 162.83 MB | 337.23 MB |
| Coding Practice | 60 % | 60 % |

Table 2.2 – First run of Distraction Tracking

Second Run:

|  |  |  |
| --- | --- | --- |
| Metric | Project Code | Compared Code |
| Accuracy | 94.25 % | 56.82 % |
| Latency/ Speed | 0.0265 seconds/frame | 0.0509 seconds/frame |
| Security | 70 % | 70 % |
| Memory Usage | 162.39 MB | 336.5 MB |
| Coding Practice | 60 % | 60 % |

Table 2.3 –Second run of Distraction Tracking

Third Run:

|  |  |  |
| --- | --- | --- |
| Metric | Project Code | Compared Code |
| Accuracy | 92.79 % | 63.95 % |
| Latency/ Speed | 0.0263 seconds/frame | 0.0642 seconds/frame |
| Security | 70 % | 70 % |
| Memory Usage | 162.42 MB | 335.10 MB |
| Coding Practice | 60 % | 60 % |

Table 2.4 –Third run of Distraction Tracking

Fourth Run:

|  |  |  |
| --- | --- | --- |
| Metric | Project Code | Compared Code |
| Accuracy | 91.82 % | 54.51 % |
| Latency/ Speed | 0.0261 seconds/frame | 0.0623 seconds/frame |
| Security | 70 % | 70 % |
| Memory Usage | 162.32 MB | 335.43 MB |
| Coding Practice | 60 % | 60 % |

Table 2.5 –Fourth run of Distraction Tracking

Fifth Run:

|  |  |  |
| --- | --- | --- |
| Metric | Project Code | Compared Code |
| Accuracy | 92.4 % | 54.70 % |
| Latency/ Speed | 0.0263 seconds/frame | 0.0485 seconds/frame |
| Security | 70 % | 70 % |
| Memory Usage | 163.03 MB | 332.91 MB |
| Coding Practice | 60 % | 60 % |

Table 2.6 –Fifth run of Distraction Tracking

#### (b) Final Average Data

|  |  |  |
| --- | --- | --- |
| Metric | Project Code | Compared Code |
| Accuracy | 92.89 % | 57.15 % |
| Latency/ Speed | 0.0263 seconds/frame | 0.0572 seconds/frame |
| Security | 70 % | 70 % |
| Memory Usage | 162.60 MB | 335.43 MB |
| Coding Practice | 60 % | 60 % |

Table 2.7 - Average of 5 runs of Distraction Tracking

As the table 2.7 depicts, the project code is trumps the geeksforgeeks code by a high 35.74%. The project code also has a lower latency rate than the compared code making it the faster code. Both the codes share the common ground in the matter of Security and coding practices. Lastly the compared code uses more than double amount of the memory making the projects’ code more lightweight and memory-efficient.

### II) Advantages over the compared code

* More Accurate
* Lower latency , faster execution
* Lesser memory utilization
* Equally Secured

**Additional Facility:**

* Includes an Alert for distraction when a thee user is not looking at the screen for constant 10 seconds and that particular facility is not present in the compared geeksforgeeks code. This facility makes the code more helpful and more goal-oriented fulfilling the objective of assisting the working professionals from work from home.

### III) Technical Rationale of Differences

Accuracy :

The higher level of accuracy achieved by the Projects code might primarily because of the specific targeted landmarks set in the code. The code utilizes only the eye landmarks while the compared code plots and uses landmarks all over the face making it more complex to calculate. Another reason might be that the project code applies the FaceMesh solution particularly targeting the face while the Geeks for geeks code applies the Holistic model that also includes models to recognize full body , hands and various poses. This includes unnecessary values and possible noise in the data directly affecting the accuracy.

Latency(Speed) :

As far as latency is considered, because the projects’ code only needs time to plot and recognize landmarks on the eye it requires comparatively lesser time while the compared code needs higher amount time to not only recognize but also to plot landmarks all over the face. Also because of higher landmarks the 3D mesh is bigger and more detailed making the render process heavy and slower.

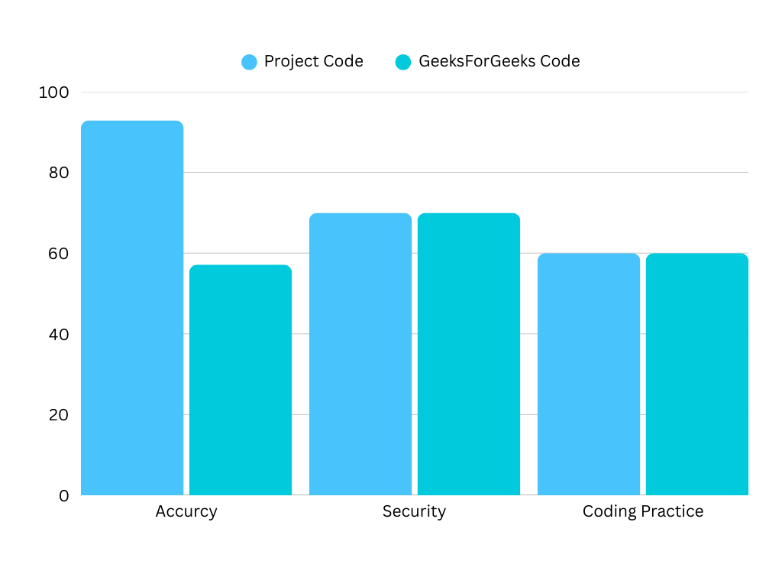
Memory Usage :

Firstly as discussed that the geeks for geeks code generated a more detailed 3D mesh, it has more facial landmarks to store in the memory which might result into increasing the memory usage. On the other hand , the projects’ code ignore the other unnecessary facial landmarks and only loads as well as stores the eyes(iris) landmarks making it more light weight and lower in memory consumption. Another possible reason might be that as the compared code operates with Holistic model , it has to load multiple heavy models for face , body , hand gestures and many more , making it consuming higher amount of memory , while the other code operates on FaceMesh solution that includes merely the facial models.

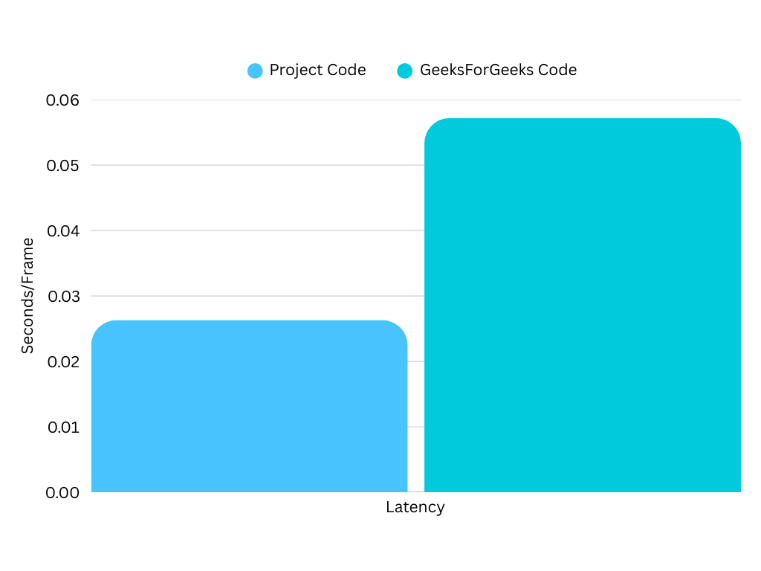
Security and Coding Practices :

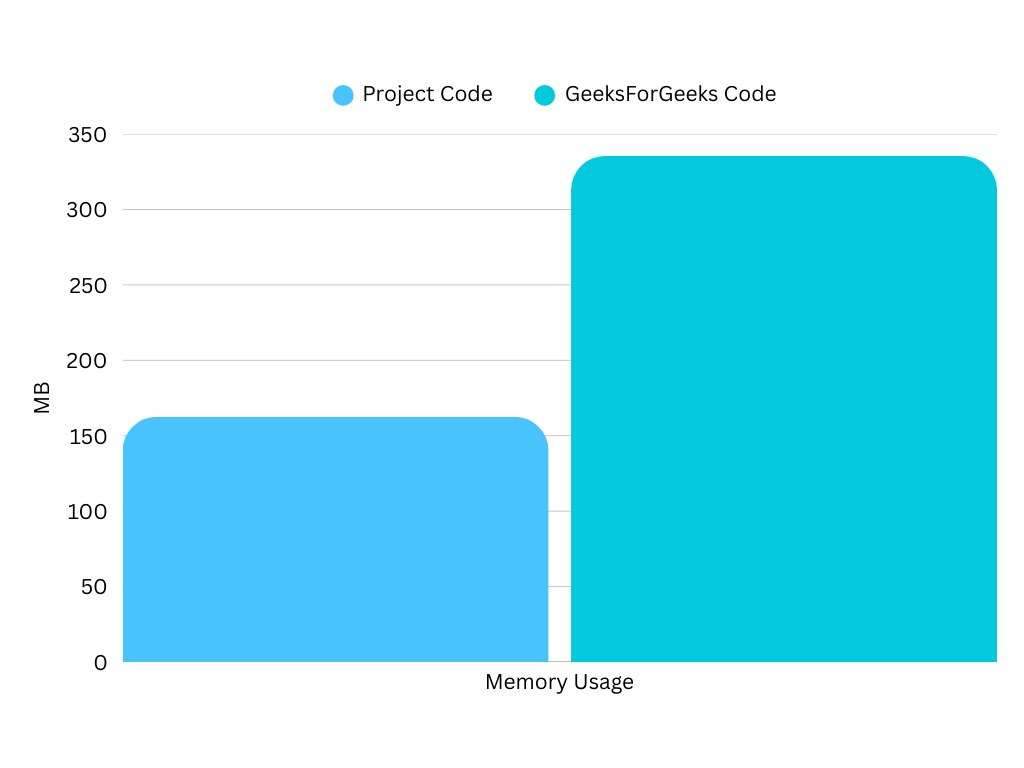
No change is seen in both the code. They both posses the same amount of security and well-coding practices.

### IV) Graphical Representation



Graph 1.1 – Graphical representation of Accuracy , Security and Coding Practices of Distraction Tracking



Graph 1.2 – Graphical representation of Latency of Distraction Tracking 

Graph 1.3 – Graphical representation of Memory Usage of Distraction Tracking

### V) Multi-environmental testing

Multi-environment testing is basically putting the code in different variation of the environment it is running. Now for the more in-depth testing the codes are put through multi environments to check if there are any visible variations or do the codes go under minimal or no change under the extremes.

#### (a) Why ?

A code might work a perfectly in a certain environment but can work drastically inverse in a different given condition. This uncertainty is nearly impossible to figure out in the planning stage or even in the coding stage. Hence running and testing the codes in multiple environment can provide more optimal , better and refined data. If there are any possible issues or bugs within the code , this form of testing provides a better probability to detect and pick out those bugs which furtherly by solving can increase the potential of the code by making it better and improving the qualities like handling edge cases.

#### (b) How ?

Basically the code would be run the similar way tested before , but there would a change of a certain environmental parameter that would be more and more extreme with every run. There would be total 3 runs including 3 changes i.e 3 different scenarios to collect data from. Yet again , it is not ideal to use a single data to properly and clearly check the differences of the environment hence , a code would be run 5 times per environment and then the average data of that would be compared with the average of the other data that has been also run 5 times per environment. For the multi-environment change , CPU Load metric is used and manipulated. The CPU Load is set at 10% initially , the environment in which the above data is collected , then the CPU Load is set at 45% increasing the intensity and then finally the CPU Load is set on 90 % to check the final extreme condition.

#### (c) Why CPU Load ?

The whole aim and purpose of the project and developing the system is to assist the working from home professionals. This system is not an independent activity like a game or an application which will have direct attention of the user while being run , this system is majorly going to be used as a background app or activity which will assist and help the user while he/she being in the working structure will have multiple applications and software running simultaneously. The CPU Load of an working entity changes and fluctuates rapidly and drastically. Thus it is important that system is able to handle the load and the fluctuations properly and not act differently when the CPU Load increases or decreases extremely. Hence keeping all the above facts in center , CPU Load is one of the major and most important metric to be used to test and evaluate for a multi-environmental change.

#### (d) Results

##### (i) Tabular Data

###### 10% CPU Load Data

First Run:

|  |  |  |
| --- | --- | --- |
| Metric | Project Code | Compared Code |
| Accuracy | 93.24 % | 55.79 % |
| Latency/ Speed | 0.0265 seconds/frame | 0.0599 seconds/frame |
| Security | 70 % | 70 % |
| Memory Usage | 162.83 MB | 337.23 MB |
| Coding Practice | 60 % | 60 % |

Table 3.1 – First run of Distraction Tracking with CPU Load 10%

Second Run:

|  |  |  |
| --- | --- | --- |
| Metric | Project Code | Compared Code |
| Accuracy | 94.25 % | 56.82 % |
| Latency/ Speed | 0.0265 seconds/frame | 0.0509 seconds/frame |
| Security | 70 % | 70 % |
| Memory Usage | 162.39 MB | 336.5 MB |
| Coding Practice | 60 % | 60 % |

Table 3.2 – Second run of Distraction Tracking with CPU Load 10%

Third Run:

|  |  |  |
| --- | --- | --- |
| Metric | Project Code | Compared Code |
| Accuracy | 92.79 % | 63.95 % |
| Latency/ Speed | 0.0263 seconds/frame | 0.0642 seconds/frame |
| Security | 70 % | 70 % |
| Memory Usage | 162.42 MB | 335.10 MB |
| Coding Practice | 60 % | 60 % |

Table 3.3 – Third run of Distraction Tracking with CPU Load 10%

Fourth Run:

|  |  |  |
| --- | --- | --- |
| Metric | Project Code | Compared Code |
| Accuracy | 91.82 % | 54.51 % |
| Latency/ Speed | 0.0261 seconds/frame | 0.0623 seconds/frame |
| Security | 70 % | 70 % |
| Memory Usage | 162.32 MB | 335.43 MB |
| Coding Practice | 60 % | 60 % |

Table 3.4 – Fourth run of Distraction Tracking with CPU Load 10%

Fifth Run:

|  |  |  |
| --- | --- | --- |
| Metric | Project Code | Compared Code |
| Accuracy | 92.4 % | 54.70 % |
| Latency/ Speed | 0.0263 seconds/frame | 0.0485 seconds/frame |
| Security | 70 % | 70 % |
| Memory Usage | 163.03 MB | 332.91 MB |
| Coding Practice | 60 % | 60 % |

Table 3.5 – Fifth run of Distraction Tracking with CPU Load 10%

**Final Average Data**

|  |  |  |
| --- | --- | --- |
| Metric | Project Code | Compared Code |
| Accuracy | 92.89 % | 57.15 % |
| Latency/ Speed | 0.0263 seconds/frame | 0.0572 seconds/frame |
| Security | 70 % | 70 % |
| Memory Usage | 162.60 MB | 335.43 MB |
| Coding Practice | 60 % | 60 % |

Table 3.6 – Average of 5 runs of Distraction Tracking with CPU Load 10%

###### 45% CPU Load Data

First Run:

|  |  |  |
| --- | --- | --- |
| Metric | Project Code | Compared Code |
| Accuracy | 91.98 % | 57.14 % |
| Latency/ Speed | 0.0265 seconds/frame | 0.0603 seconds/frame |
| Security | 70 % | 70 % |
| Memory Usage | 164.65 MB | 335.61 MB |
| Coding Practice | 60 % | 60 % |

Table 3.7 – First run of Distraction Tracking with CPU Load 45%

Second Run:

|  |  |  |
| --- | --- | --- |
| Metric | Project Code | Compared Code |
| Accuracy | 97.80 % | 51.48 % |
| Latency/ Speed | 0.0269 seconds/frame | 0.0573 seconds/frame |
| Security | 70 % | 70 % |
| Memory Usage | 163.04 MB | 335.46 MB |
| Coding Practice | 60 % | 60 % |

Table 3.8 – Second run of Distraction Tracking with CPU Load 45%

Third Run:

|  |  |  |
| --- | --- | --- |
| Metric | Project Code | Compared Code |
| Accuracy | 94.60 % | 55.81 % |
| Latency/ Speed | 0.0264 seconds/frame | 0.0586 seconds/frame |
| Security | 70 % | 70 % |
| Memory Usage | 166.02 MB | 336.5 MB |
| Coding Practice | 60 % | 60 % |

Table 3.9–Third run of Distraction Tracking with CPU Load 45%

Fourth Run:

|  |  |  |
| --- | --- | --- |
| Metric | Project Code | Compared Code |
| Accuracy | 92.31 % | 49.28 % |
| Latency/ Speed | 0.0262 seconds/frame | 0.0607 seconds/frame |
| Security | 70 % | 70 % |
| Memory Usage | 166.37 MB | 337.31 MB |
| Coding Practice | 60 % | 60 % |

Table 3.10 – Fourth run of Distraction Tracking with CPU Load 45%

Fifth Run:

|  |  |  |
| --- | --- | --- |
| Metric | Project Code | Compared Code |
| Accuracy | 97.37 % | 46.81 % |
| Latency/ Speed | 0.0261 seconds/frame | 0.0604 seconds/frame |
| Security | 70 % | 70 % |
| Memory Usage | 162.86 MB | 336.08 MB |
| Coding Practice | 60 % | 60 % |

Table 3.11 – Fifth run of Distraction Tracking with CPU Load 45%

**Final Average Data**

|  |  |  |
| --- | --- | --- |
| Metric | Project Code | Compared Code |
| Accuracy | 94.8 % | 52.1 % |
| Latency/ Speed | 0.0336 seconds/frame | 0.0595 seconds/frame |
| Security | 70 % | 70 % |
| Memory Usage | 164.59 MB | 336.19 MB |
| Coding Practice | 60 % | 60 % |

Table 3.12 – Average of 5 runs of Distraction Tracking with CPU Load 45%

###### 90% CPU Load Data

First Run:

|  |  |  |
| --- | --- | --- |
| Metric | Project Code | Compared Code |
| Accuracy | 84.75 % | 52.15 % |
| Latency/ Speed | 0.0304 seconds/frame | 0.0802 seconds/frame |
| Security | 70 % | 70 % |
| Memory Usage | 164.10 MB | 335.67 MB |
| Coding Practice | 60 % | 60 % |

Table 3.13 – First run of Distraction Tracking with CPU Load 90%

Second Run:

|  |  |  |
| --- | --- | --- |
| Metric | Project Code | Compared Code |
| Accuracy | 89.68 % | 54.52 % |
| Latency/ Speed | 0.0317 seconds/frame | 0.0806 seconds/frame |
| Security | 70 % | 70 % |
| Memory Usage | 166.27 MB | 335.97 MB |
| Coding Practice | 60 % | 60 % |

Table 3.14 – Second run of Distraction Tracking with CPU Load 90%

Third Run:

|  |  |  |
| --- | --- | --- |
| Metric | Project Code | Compared Code |
| Accuracy | 92.91 % | 61.18 % |
| Latency/ Speed | 0.0295 seconds/frame | 0.1044 seconds/frame |
| Security | 70 % | 70 % |
| Memory Usage | 164.86 MB | 335.75 MB |
| Coding Practice | 60 % | 60 % |

Table 3.15 – Third run of Distraction Tracking with CPU Load 90%

Fourth Run:

|  |  |  |
| --- | --- | --- |
| Metric | Project Code | Compared Code |
| Accuracy | 90.92 % | 49.08 % |
| Latency/ Speed | 0.0305 seconds/frame | 0.0823 seconds/frame |
| Security | 70 % | 70 % |
| Memory Usage | 160.89 MB | 336.23 MB |
| Coding Practice | 60 % | 60 % |

Table 3.16 – Fourth run of Distraction Tracking with CPU Load 90%

Fifth Run:

|  |  |  |
| --- | --- | --- |
| Metric | Project Code | Compared Code |
| Accuracy | 94.94 % | 51.94 % |
| Latency/ Speed | 0.0306 seconds/frame | 0.0821 seconds/frame |
| Security | 70 % | 70 % |
| Memory Usage | 165.68 MB | 335.88 MB |
| Coding Practice | 60 % | 60 % |

Table 3.17 – Fifth run of Distraction Tracking with CPU Load 90%

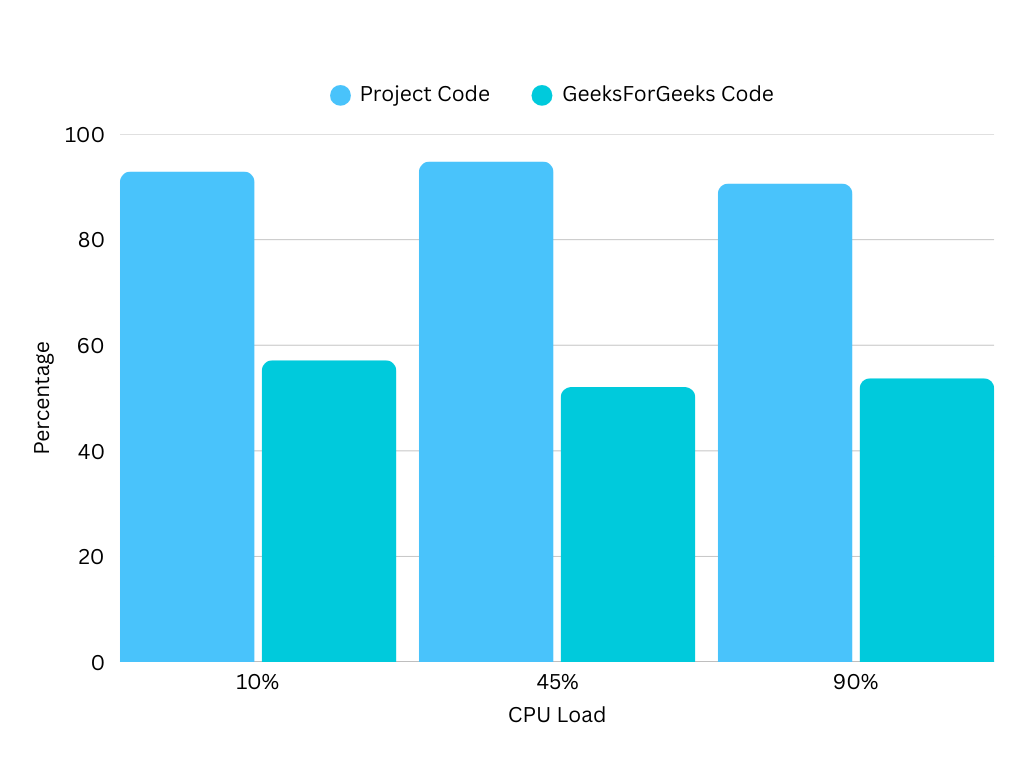
**Final Average Data**

|  |  |  |
| --- | --- | --- |
| Metric | Project Code | Compared Code |
| Accuracy | 90.64 % | 53.77 % |
| Latency/ Speed | 0.0305 seconds/frame | 0.0859 seconds/frame |
| Security | 70 % | 70 % |
| Memory Usage | 164.36 MB | 335.9 MB |
| Coding Practice | 60 % | 60 % |

Table 3.18 – Average of 5 runs of Distraction Tracking with CPU Load 90%

##### (ii) Graphical Representation

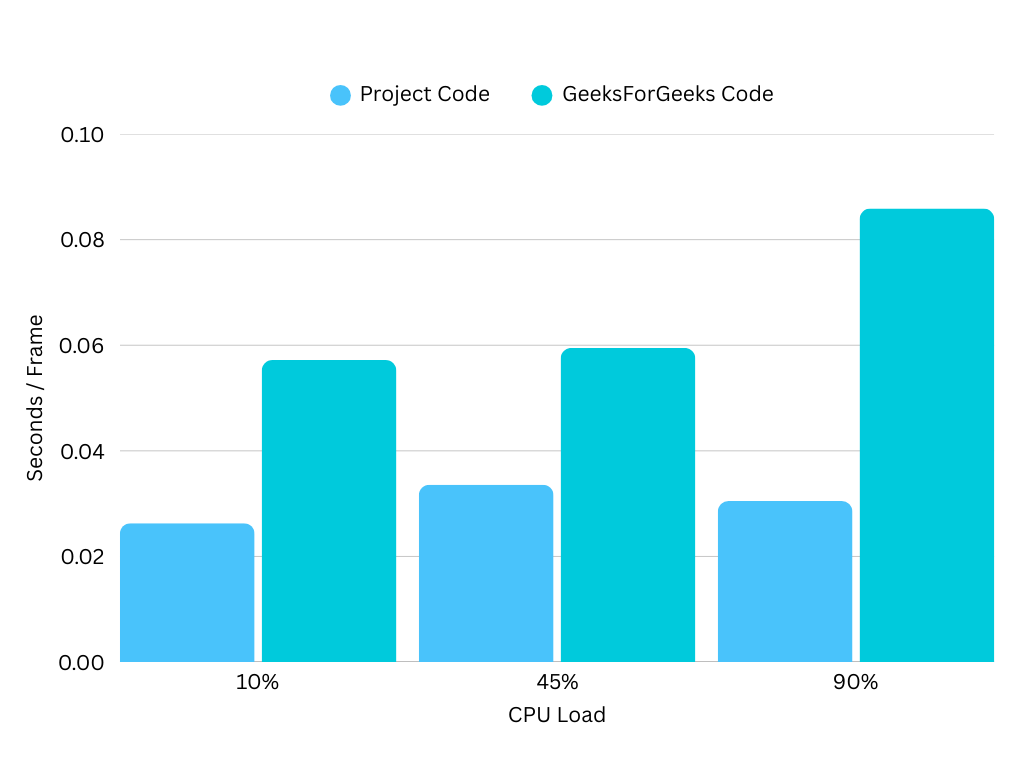
###### Accuracy



Graph 2.1 – Graphical Representation of Accuracy of all 3 CPU Loads

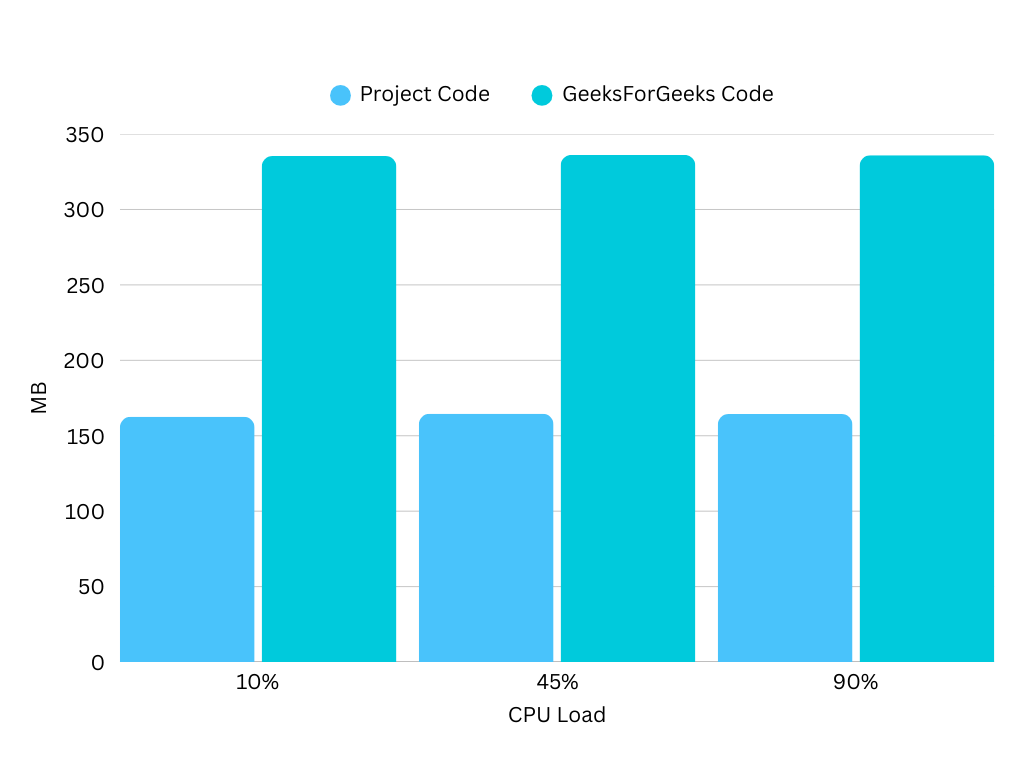
As it is visible in Graph 2.1 that some level of fluctuation is seen in accuracy but overall the system has proven reliable with the accuracy rate being in the same range without producing any outlier value under the various and extreme circumstances.

###### Latency

 Graph 2.2 – Graphical Representation of Latency of all 3 CPU Loads

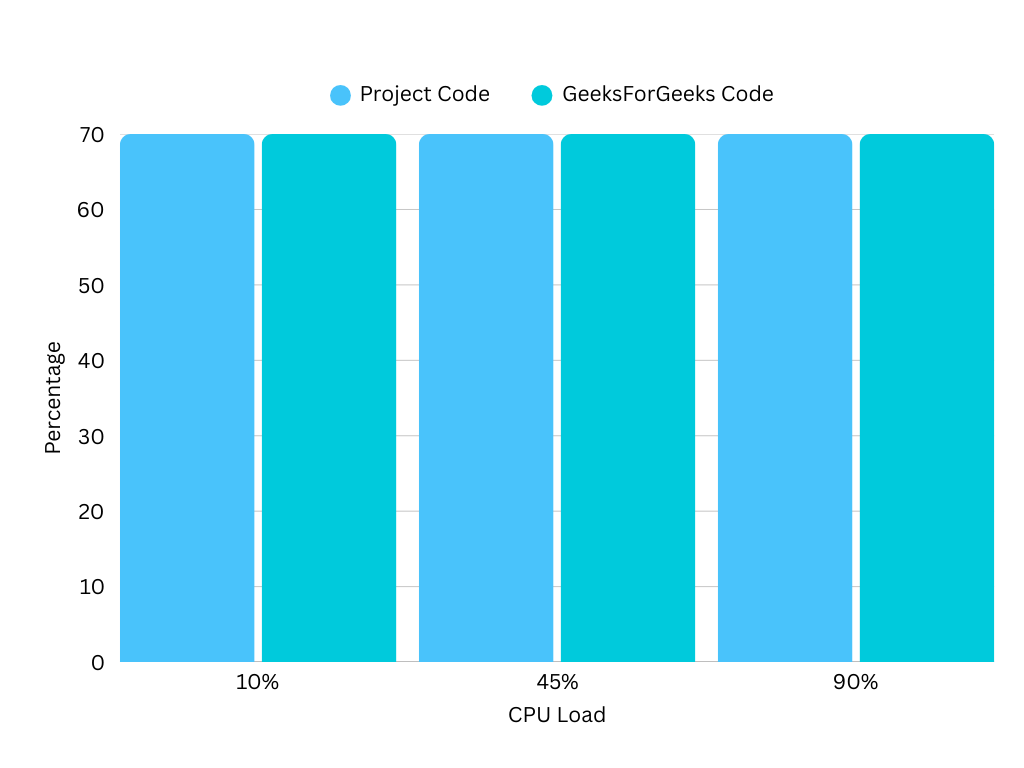
The Graph 2.2 depicts a slight increment and then further a decrement in the latency rate of the projects’ code with the increasing CPU load. While on the other hand after a slight rise between 10% and 45% CPU load the latency rate increases drastically of the geeks for geeks code , the increment of this dynamic makes the code not ideal for higher CPU Load environments. Hence at busy office hours this code might not be much reliable to produce faster results.

###### Memory Usage

 Graph 2.3 – Graphical Representation of Memory Usage of all 3 CPU Loads

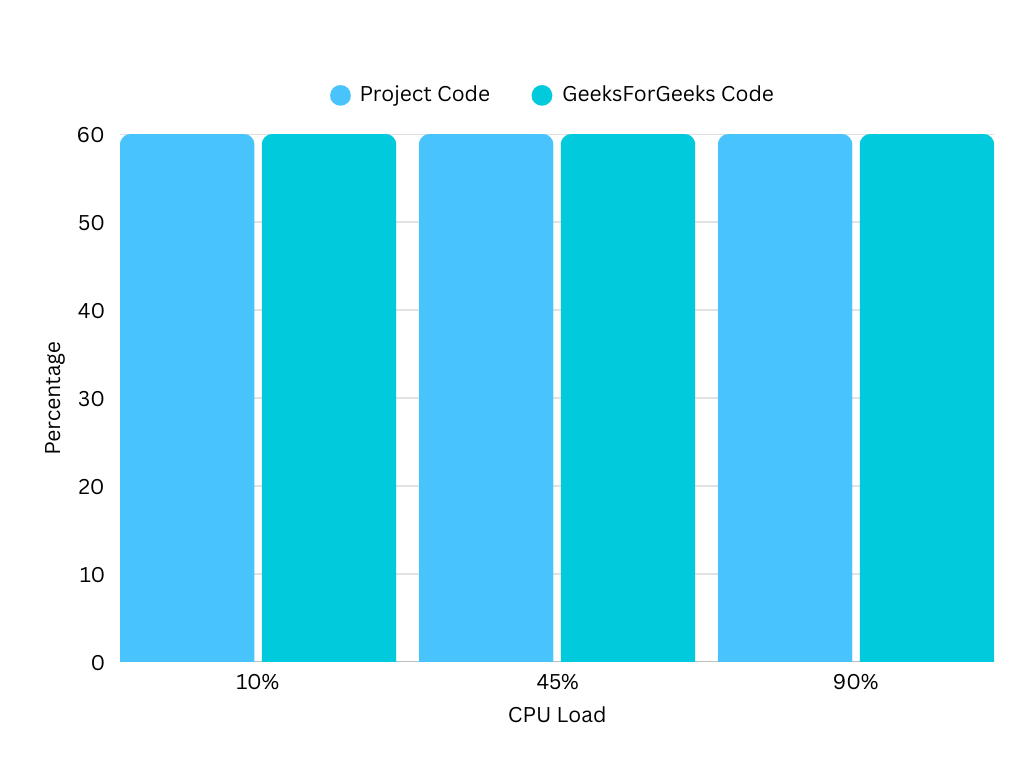
In Graph 2.3 it is clearly visible that both the codes hardly change the memory usage even on the extreme conditions. Although a minor and quiet unnoticeable increment is seen in both of the codes with the increasing CPU Load. Hence this proves that in spite of the extreme CPU Load, the projects’ system does not utilize higher memory.

###### Security

 Graph 2.4 – Graphical Representation of Security of all 3 CPU Loads

As the code is not being altered while the CPU load is set higher , the security concerns cannot be a metric in which a change , variation or even fluctuation can be seen and it that fact is clearly visible in the Graph 2.4.

###### Coding Practice

 Graph 2.5 – Graphical Representation of Coding Practice of all 3 CPU Loads

The constant values for all the CPU Load environment for the Coding Practices that is seen in the Graph 2.5 makes sense as the code is not changed or re-coded in the process of changing the environments.

###### Summary

Overall , the distraction tracking system is not only up to the professional and industry standards but can also perform perfectly under extreme conditions , making it a perfect solution to be opted by any working from home professional.

## (C) Posture Recognition Evaluation

The evaluation of this feature is done through an AI tool that evaluates 2 codes and gives the comparative metrics. The code used for the comparison is from a GitHub repository called TiffinTech/posture-corrector and below is the link to the repository :

<https://github.com/TiffinTech/posture-corrector>

This code matches the exact functionality that is explored in this project. It also provides the service to detect the posture of the person sitting in the video captured from the webcam.

### I) Comparative Analysis

For the analysis the same 5 metrics are used by the AI tool.

|  |  |  |
| --- | --- | --- |
| Metric | Project Code | Compared Code |
| Accuracy | 70 % | 90 % |
| Latency/ Speed | 0.030 seconds/frame | 0.045 seconds/frame |
| Security | 75 % | 75 % |
| Memory Usage | 92.7 MB | 183.4 MB |
| Coding Practice | 80 % | 70 % |

Table 4.1 – Tabular Data of Posture Recognition Evaluation

As it is visible in the table 4.1 , the code developed for this report has a lower accuracy rate than the code available on the market , the accuracy is lower by 20%. While in the terms of the latency , the compared code (i.e the github public code) is slower than the developed code by 0.015 %. Even though that number looks negligible but when there is the matter of real time frame capturing , every millisecond per frame is important , when comparing a larger data or comparing for a longer period of time that minor difference can make a higher impact. About security, both the codes tie up to have the exact amount of safety and vulnerabilities. The projects’ code utilizes significantly lesser memory than the other code. The difference is not minimal but almost double making the projects’ code more compact. Also , the reports’ code is structured better resulting into the higher percentage value in the coding practice metrics than the compared code. The difference it merely 10%.

### II) Advantages over the compared code

* Lower latency , faster execution
* Lesser memory utilization
* Better structured code
* Secured

**Additional Facility:**

* Includes an Alert for bad posture when a bad posture is detected for constant 10 seconds that is not present in the compared open source code. This facility makes the code more compatible and sensible to use in the direction for assisting the remote working professionals than the compared code.

### III) Technical Rationale of Differences

Accuracy :

The compared code has higher level of accuracy and the reason might be the logic of detecting the posture , as it has a more complex logic that includes a calculation of shoulder and neck angles while the projects’ code works on a comparatively simpler logic that includes the calculation of the difference between left shoulder and left ear.

Latency(Speed) :

As the projects’ code uses a simpler logic and calculation , the computation is lighter and thus the faster speed and lesser latency. On the flip side, the compared code is based on highly heavy computations and calculations which results into utilizing more time , increasing latency.

Memory Usage :

The primary reason of being the memory usage high in the compared code might be that not only it uses more data for calculations but also it uses additional features and facilities like sound utilization , plotting libraries and more. The projects’ code does neither requires higher memory for calculation not introduces any extra or additional features resulting into a lesser memory usage.

Coding Practices :

The projects;’ code seems much cleaner and simple while the compared code is tightly clustered and has so many complexity for the detection calculation. Also the projects’ code uses better coding practices like flags , functions etc. than the compared code.

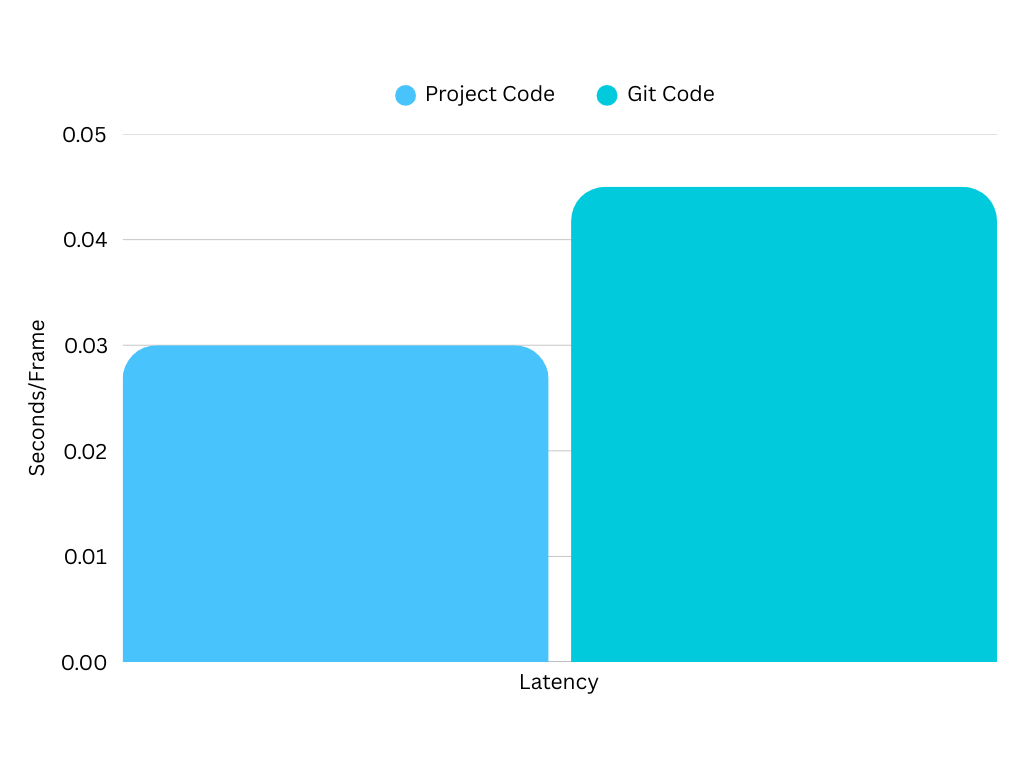
Security :

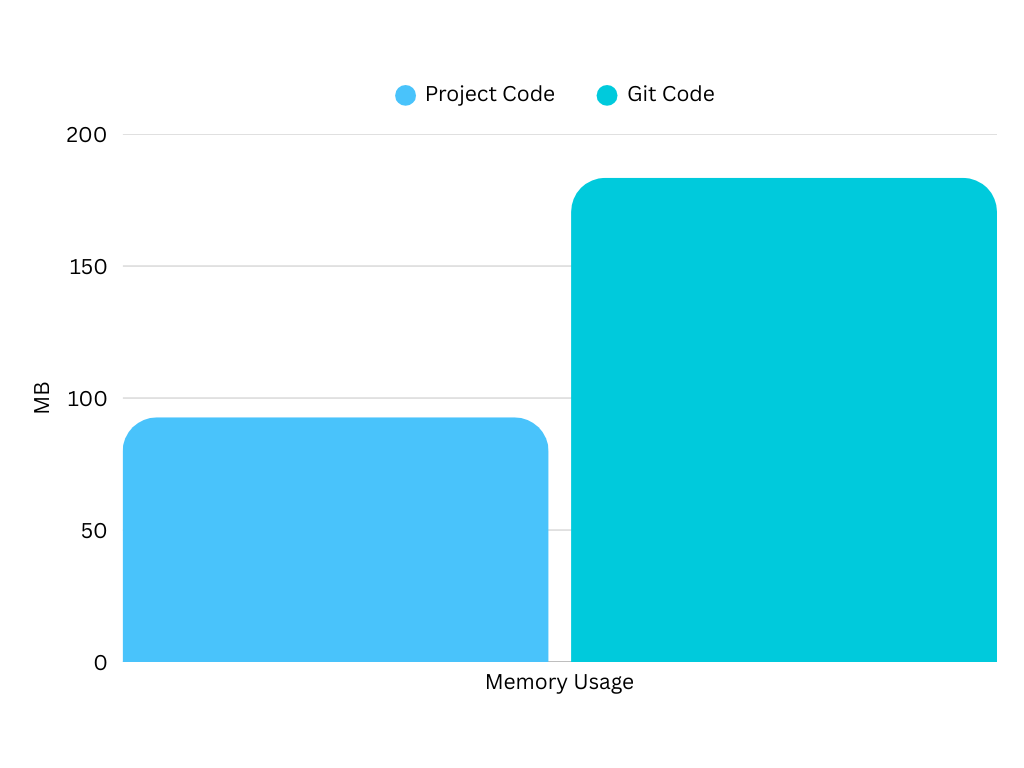
As both the codes use more or less the same libraries to function , they both are equally secured.

### IV) Graphical Representation



Graph 3.1 – Graphical Representation of Accuracy , Security and Coding Practice of Posture Recognition codes

 Graph 3.2 – Graphical Representation of latency of Posture Recognition codes



Graph 3.3 – Graphical Representation of Memory Usage of Posture Recognition codes

## (D) Background Noise Reduction

The use of AI tool is also done for the evaluation of this feature. Although the comparison metrics have remained the same. This code is compared with a similar code having the same functional value with a public Git repository of “Sa-if” , below is the link of the repository :

<https://github.com/sa-if/Audio-Denoiser>

If given more time for exploration , a better and more powerful alternative might be found but keeping all the limitations of this projects i.e the deadlines and the dynamics, this repository was the best contender for comparison.

### I) Comparative Analysis

Below is the table of the comparative analysis of both codes by an Artificial Intelligence tool , compared in the basis of the same metrics used in the previous functionalities.

|  |  |  |
| --- | --- | --- |
| Metric | Project Code | Compared Code |
| Accuracy | 74 % | 88 % |
| Latency/ Speed | 0.8 seconds/full file | 2.1 seconds/full file |
| Security | 82 % | 75 % |
| Memory Usage | 83.9 MB | 152.3 MB |
| Coding Practice | 85 % | 80 % |

Table 5.1 – Tabular Data of Background Noise Suppression Evaluation

As it is visible in the table 5.1 , in terms of accuracy the projects’ code has lesser accuracy than the compared code by 14%. On the other hand the projects’ code is faster in execution as the latency value is lesser than half of that of the compared git code. As far as security is concerned , the code developed for the project has proven 7% more secured than the code from the Git repository. Above that memory used by the git code is almost double than the memory utilized by projects’ code. Even though there is not much difference but still the project developed code trumps the number of percentage of coding practices by 5%.

### II) Advantages over the compared code

* Lower latency , faster execution
* Lesser memory utilization
* Better structured code
* Lesser code complexity
* More Secured

**Additional Facility:**

* The compared code works generally on the constant background noise like humming , while the projects’ code not only does that but deeply isolates the vocals from the background noise which can also work on irregular noises , sudden noise and even musical disturbance.

### III) Technical Rationale of Differences

Accuracy :

The compared code has higher level of accuracy than the projects’ code and the main reason of that might be because the compared code uses noisereduce libraries which is a more advanced alternative than HPSS that is being used in the projects’ code.

Latency(Speed) :

The projects’ code is faster because of one main reason that it converts the processed audio in merely one step with the help of librosa , while the compared code first converts into an array with numpy and then converts back to MP3. Also the compared code plots the data using the Matplotlib which seems a stretch and thus resulting into low speed in execution.

Memory Usage :

As discussed earlier the compared code uses numpy to convert data in to array and then back to MP3 that process is memory intensive and requires additional memory while the projects’ code uses a simple audio processing function to perform that operation. Above that the Matplotlib and visualizing the data also uses memory and not in a minimalistic way.

Security:

One of the probable reason of a slight better security rate of the projects’ code might be that it imports lesser libraries than the compared code and the more libraries the probability of more vulnerabilities found increases and also be exploited through the local machine.

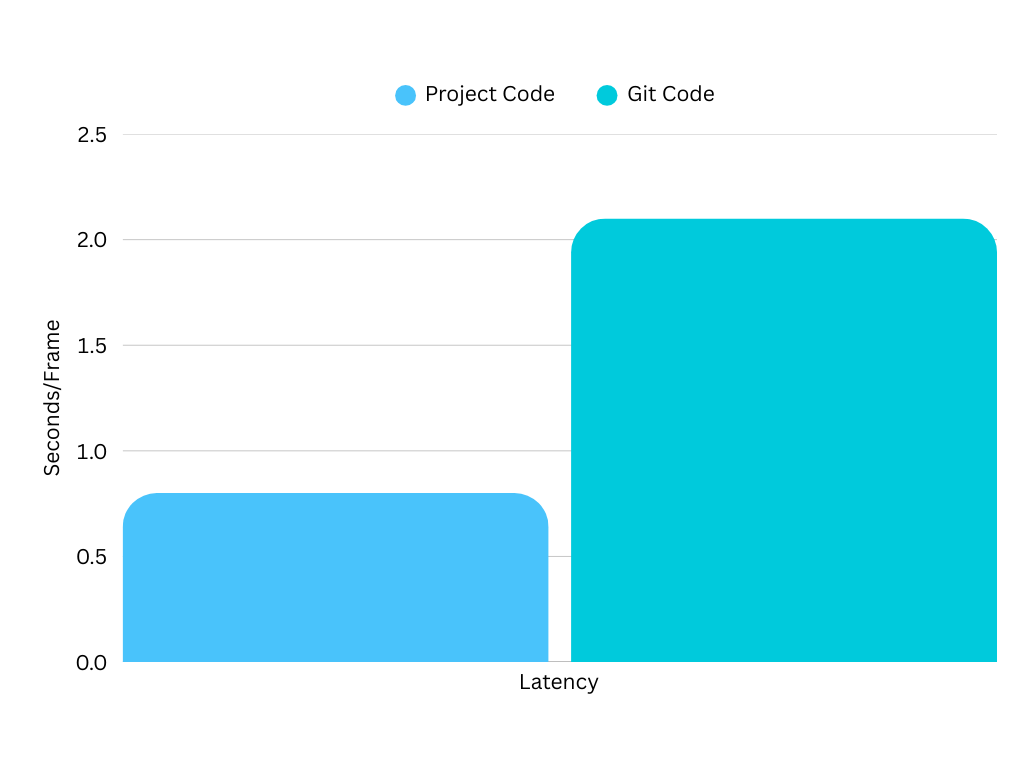
Coding Practices:

The obvious reason of the coding practice rate being higher of the projects’ code might be that it is simple and does not include any complex or unnecessary steps like plotting or visualizing or converting the data into arrays and back.

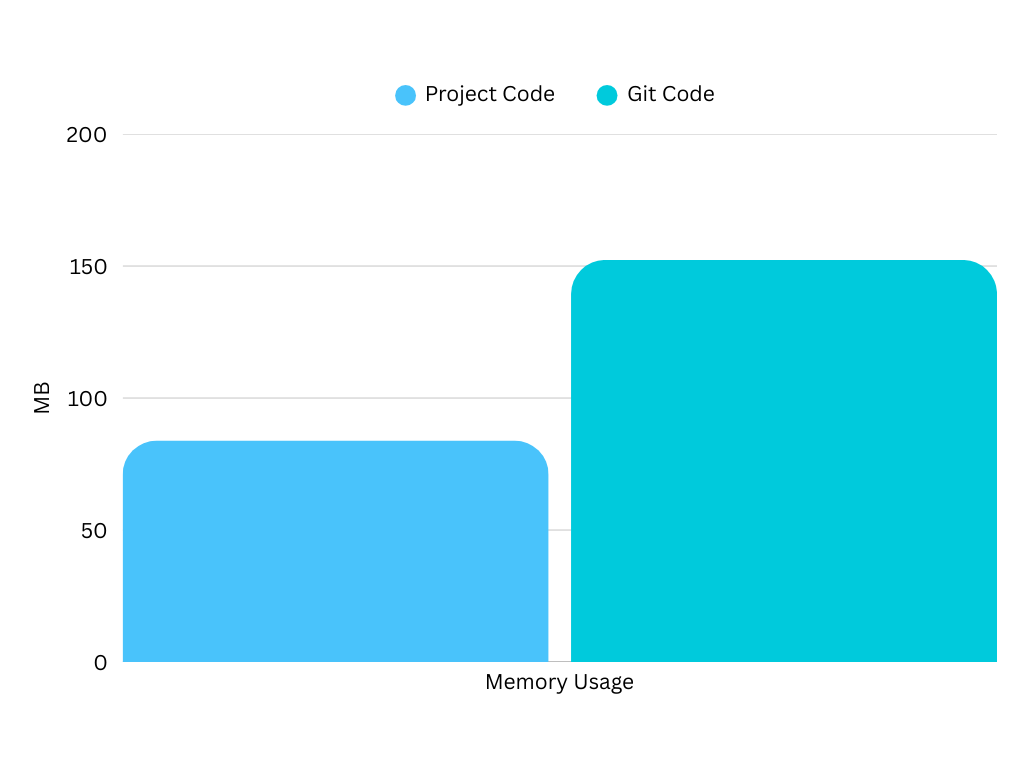
### IV) Graphical Representation



Graph 4.1 – Graphical Representation of Accuracy , Security and Coding Practice of Posture Recognition codes



Graph 4.2 – Graphical Representation of Latency of Posture Recognition codes



Graph 4.2 – Graphical Representation of Memory Usage of Posture Recognition codes

# 6) Potential Enhancements

No matter however perfect a system is there is always a scope for upgradation. Keeping in mind the aim and objectives of the system , there are a certain potential upgradations that can be done. A primary update would be a database integration , where the data of the distracted time and data of the time when the posture was incorrect can be stored. That data can be further utilized for any data analysis or Machine Learning approach. Another possible enhancement can be in depth training to the AI model for removing the background noise more clearly and make it more smooth and refined by focusing on particular sound waves to suppress more detailed noises like dog barking or child crying. More over , a user friendly GUI and dashboard can be designed to make it more attractive and user friendly. That dashboard can visually, show the metrics of the code making it more refined and easy to use. Above that a to increase the mobility of the assistance , a mobile application can be developed that inculcates the similar UI/UX strata. Also as far as eye tracking is concerned , an alert or limit can be set to detect the strain or basically give reminder to have a break after a certain point of time. In order to make it more user-interactive , gamification can be possible , in which a certain points or rewards are guaranteed when a posture is maintained for a specific time or if the average distraction time of the week is less. Audio feedback functionality can be added so that if pop-up or alerts are gone unnoticed , a voice alert can be used to get the user’s attention. Last but not the least , a multilingual support should be provided so that the software can also be enjoyed and utilized by the non-English speaking communities/people.

# 7) Ethical Considerations

It is essential to keep the ethical considerations in mind while handling a project. This projects follows the code of conduct of an esteemed professional learned society called “British Computer Society” and it follows the “Code of Conduct”. Furthermore this projects does not involve any direct or indirect contact with human participants. No sensitive or personal information is gathered or collected for any research or analysis purposes and it does not involve any risk of compromising confidentiality or anonymity. The project involves all the information and data that is available freely on the public domain. It does not cause any physical or psychological harm or negative consequences to any individual. For in depth ethical clearance the Leeds Beckett University Research Ethics form has been taken and below is the link of that form being approved.

<https://researchethics.leedsbeckett.ac.uk/print/view/148764>

# 8) Conclusion

In conclusion, this report not only discusses thoroughly through the detailed literature review about the problems faced by the individuals that are working from home but also gives an intact solution for it. The aim and objectives that were set to achieve was completely fulfilled. A system has been introduced with all the relevant features passing through any and all tests and experiments conducted on it. Hence a possible , probable and plausible solution is found for recognizing the correct posture , tracking the distraction and reducing the background noise which if used and integrated with enough resources and handles properly can reduce the issues faced by the remote working professionals at a huge rate and can boost up their productivity and comfort drastically, having the work-from-home environment a completely different way to execute.

# 9) References

1. Arlinghaus, A. and Nachreiner, F. (2014) Health effects of supplemental work from home in the European Union. Chronobiology International, 31(10), pp. 1100–1107. doi: 10.3109/07420528.2014.957297.
2. Bennett, A.A., Campion, E.D., Keeler, K.R. and Keener, S.K. (2021) ‘Videoconference fatigue? Exploring changes in fatigue after videoconference meetings during COVID-19’, Journal of Applied Psychology, 106(3), pp. 330–344. Available at: <https://doi.org/10.1037/apl0000906>.
3. Bergefurt, L. et al. (2022) The influence of distractions of the home-work environment on mental health during the COVID-19 pandemic. Ergonomics, 66(1), pp. 16–33. doi: 10.1080/00140139.2022.2053590.
4. Bick, A., Blandin, A., and Mertens, K. (2023) Work from home before and after the COVID-19 outbreak. *American Economic Journal: Macroeconomics*, 15(4), October, pp. 1–39.
5. Butte, K.T., Cannavan, D., Hossler, J. et al. (2023) ‘The relationship between objectively measured sitting time, posture, and low back pain in sedentary employees during COVID-19’, Sport Science and Health, 19, pp. 259–266. Available at: <https://doi.org/10.1007/s11332-022-01031-x>.
6. Chitta, S., Pamidi Venkata, A. K., Ahmad, T. et al. (2021) Remote project management: Best practices for distributed teams in the post-pandemic era. Journal of Machine Learning Research, 1(2) July–December, pp. 145.
7. Faulds, D.J. and Raju, P.S. (2021) The work-from-home trend: an interview with Brian Kropp. *Business Horizons*, 64(1), Jan-Feb, pp. 29-35. doi: 10.1016/j.bushor.2020.10.005.
8. Galanti, T.M. et al. (2021) Work from home during the COVID-19 outbreak: the impact on employees’ remote work productivity, engagement, and stress. *Journal of Occupational and Environmental Medicine*, 63(7), July, pp. e426-e432. doi: 10.1097/JOM.0000000000002236.
9. Karl, K. A., Peluchette, J. V. and Aghakhani, N. (2022) Virtual work meetings during the COVID-19 pandemic: The good, bad, and ugly. Small Group Research, 53(3), pp. 343–365. doi: 10.1177/10464964211015286.
10. Kong, X. et al. (2022) Work from home in the post-COVID world. *Case Studies on Transport Policy*, 10, pp. 1118-1131.
11. Luebstorf, S., Allen, J.A., Eden, E., Kramer, W.S., Reiter-Palmon, R. and Lehmann-Willenbrock, N. (2023) Digging into “Zoom Fatigue”: A Qualitative Exploration of Remote Work Challenges and Virtual Meeting Stressors. Merits, 3(1), pp. 151–166. <https://doi.org/10.3390/merits3010010>.
12. Mora-Jimenez, L.D., Ramírez-Benavides, K., Quesada, L., Lopez, G. and Guerrero, L.A. (2022) User experience in communication and collaboration platforms: a comparative study including Discord, Microsoft Teams, and Zoom. In: Rocha, Á., Ferrás, C., Méndez Porras, A. and Jimenez Delgado, E. eds. Information Technology and Systems. ICITS 2022, 12-14 February 2022, Bogotá, Colombia. Cham: Springer. Available from: <https://doi.org/10.1007/978-3-030-96293-7_6>.
13. Park, S.H., Shin, H-K. and Kim, K-W. (2023) Relationship between indoor noise perception and remote work during the COVID-19 pandemic. PLOS ONE, 18(6), e0286481. <https://doi.org/10.1371/journal.pone.0286481>.
14. Ugemuge, P.T., Shingankar, S., and Joshi, S. (2022) Work from home – a growing trend in IT companies: benefits, drawbacks, opportunities, and challenges. *Journal of Datta Meghe Institute of Medical Sciences University*, 17(4), Oct-Dec, pp. 1009-1012. doi: 10.4103/jdmimsu.jdmimsu\_456\_21.
15. Xiao, Y. et al. (2021) Impacts of working from home during COVID-19 pandemic on physical and mental well-being of office workstation users. Journal of Occupational and Environmental Medicine, 63(3), pp. 181–190. doi: 10.1097/JOM.0000000000002097.

# 10) Appendix

## (A) Posture Recognition Cod

|  |
| --- |
| import ctypes  import time  import cv2  import mediapipe as mp  drawmp = mp.solutions.drawing\_utils  posemp = mp.solutions.pose  video = cv2.VideoCapture(0)  flag = 1  alert = False  time\_of\_distraction = None    def get\_difference(x,y):      return abs(x[0]-y[0])    with posemp.Pose(min\_detection\_confidence = 0.6, min\_tracking\_confidence = 0.6) as pose :      while video.isOpened():          ret, frame = video.read()          frame = cv2.flip(frame,1)          if not ret:              break          image = cv2.cvtColor(frame,cv2.COLOR\_BGR2RGB)          image.flags.writeable = False          results = pose.process(image)          image.flags.writeable = True          image = cv2.cvtColor(image, cv2.COLOR\_RGB2BGR)            try:              landmarks = results.pose\_landmarks.landmark              left\_shoulder = [landmarks[posemp.PoseLandmark.LEFT\_SHOULDER.value].x,landmarks[posemp.PoseLandmark.LEFT\_SHOULDER.value].y]              left\_ear = [landmarks[posemp.PoseLandmark.LEFT\_EAR.value].x,landmarks[posemp.PoseLandmark.LEFT\_EAR.value].y]              difference = get\_difference(left\_ear,left\_shoulder)\*frame.shape[1]                if difference > 105:                  posture = "Slouching"                  color = (0,0,255)                  flag = 0                  if time\_of\_distraction is None:                      time\_of\_distraction = time.time()                  else:                      if time.time() - time\_of\_distraction > 5 and not alert:                          ctypes.windll.user32.MessageBoxW(0, "The Posture is not Correct !", "Alert", 0x60 | 0x1)                          alert = True              else:                  posture = "Good Posture"                  color = (0,255,0)                  flag = 1                  time\_of\_distraction = None                  alert = False                cv2.putText(image, f'Posture: {posture}', (30, 50),cv2.FONT\_HERSHEY\_SIMPLEX, 1, color, 2, cv2.LINE\_AA)              cv2.putText(image, f'Ear-Shoulder Offset: {int(difference)}px', (30, 90),cv2.FONT\_HERSHEY\_SIMPLEX, 0.9, color, 2, cv2.LINE\_AA)              drawmp.draw\_landmarks(image, results.pose\_landmarks, mp.POSE\_CONNECTIONS)          except:              pass            cv2.imshow('Posture Recognition',image)          if cv2.waitKey(10) & 0xFF == ord('q'):              break  video.release()  cv2.destroyAllWindows() |

## (B) Distraction Tracking Code

|  |
| --- |
| import time  import cv2  import mediapipe as mp  import ctypes  camera = cv2.VideoCapture(0)  faceMesh = mp.solutions.face\_mesh.FaceMesh(refine\_landmarks = True)  flag = 1  alert = False  distracted\_time = None  while True:      ret, frame = camera.read()      frame = cv2.flip(frame,1)      rgbframe = cv2.cvtColor(frame,cv2.COLOR\_BGR2RGB)      frame\_height, frame\_width, \_ = frame.shape      results = faceMesh.process(rgbframe)      landmark\_points = results.multi\_face\_landmarks        #         x = int(landmark.x \* frame\_width)      #         y = int(landmark.y \* frame\_height)      #         print(x,y)      #         cv2.circle(frame,(x,y),2,(0,0,255),-1)      if landmark\_points :          for face\_landmarks in landmark\_points:                right\_eye\_x\_axis = int((face\_landmarks.landmark[474].x + face\_landmarks.landmark[475].x + face\_landmarks.landmark[476].x + face\_landmarks.landmark[477].x) / 4 \* frame\_width)              right\_eye\_y\_axis = int((face\_landmarks.landmark[474].y + face\_landmarks.landmark[475].y + face\_landmarks.landmark[476].y + face\_landmarks.landmark[477].y) / 4 \* frame\_height)              left\_eye\_x\_axis = int((face\_landmarks.landmark[469].x + face\_landmarks.landmark[470].x + face\_landmarks.landmark[471].x + face\_landmarks.landmark[472].x) / 4 \* frame\_width)              left\_eye\_y\_axis = int((face\_landmarks.landmark[469].y + face\_landmarks.landmark[470].y + face\_landmarks.landmark[471].y + face\_landmarks.landmark[472].y) / 4 \* frame\_height)              right\_eye\_boundary\_left\_side = face\_landmarks.landmark[469]              right\_eye\_boundary\_right\_side = face\_landmarks.landmark[471]              left\_eye\_boundary\_left\_side = face\_landmarks.landmark[472]              left\_eye\_boundary\_right\_side = face\_landmarks.landmark[470]                cv2.circle(frame, (right\_eye\_x\_axis, right\_eye\_y\_axis), 3, (0, 255, 0), -1)              cv2.circle(frame, (left\_eye\_x\_axis, left\_eye\_y\_axis), 3, (0, 255, 0), -1)                            right\_eye\_top = int(face\_landmarks.landmark[475].y \* frame\_height)              right\_eye\_bottom = int(face\_landmarks.landmark[477].y \* frame\_height)              right\_eye\_center\_y = right\_eye\_y\_axis              left\_eye\_top = int(face\_landmarks.landmark[470].y \* frame\_height)              left\_eye\_bottom = int(face\_landmarks.landmark[472].y \* frame\_height)              left\_eye\_center\_y = left\_eye\_y\_axis              # right\_center\_top = right\_eye\_center\_y - right\_eye\_top              # right\_center\_bottom = right\_eye\_bottom - right\_eye\_center\_y              # left\_center\_top = left\_eye\_center\_y - left\_eye\_top              # left\_center\_bottom = left\_eye\_bottom - left\_eye\_center\_y              r1 = right\_eye\_center\_y - right\_eye\_top              r2 = right\_eye\_bottom - right\_eye\_center\_y              l1 = left\_eye\_center\_y - left\_eye\_top              l2 = left\_eye\_bottom - left\_eye\_center\_y                          if (r1 < 6.2 or r2 < 6.2 or l1 < 6.2 or l2 < 6.2):                  flag = 0                  cv2.putText(frame, "DISTRACTED", (50, 100),cv2.FONT\_HERSHEY\_SIMPLEX, 2, (0, 0, 255), 4)                  if distracted\_time is None:                      distracted\_time = time.time()                  else:                      if time.time() - distracted\_time > 5 and not alert:                          ctypes.windll.user32.MessageBoxW(0, "You are distracted for more than 10 seconds!", "Alert", 0x40 | 0x1)                          alert = True              else:                  flag = 1                  distracted\_time = None                  alert = False        cv2.imshow('video frame', frame)      cv2.waitKey(1) |

## (C) Background Noise Suppression Code

|  |
| --- |
| import librosa  import soundfile as sf  # from scipy.signal import medfilt  x, sr = librosa.load('raw.wav', sr=None)  har\_, per\_ = librosa.effects.hpss(x)  clean\_x = har\_  sf.write('clean\_audio.wav', clean\_x, sr)  print("Background noise reduced!") |