**CDM(Couser that detects Motion)**

CDM system can be implemented using the Python programming language, along with the OpenCV and MediaPipe libraries, and PyAutoGUI module.

OpenCV is a library for computer vision tasks, such as object detection, image processing, and video analysis. MediaPipe is a library that provides pre-trained models for various computer vision tasks, including facial landmarks detection and hand tracking. PyAutoGUI is a module that allows for automating mouse and keyboard actions.

To implement an CDM system, the first step is to use OpenCV to capture video from the camera. Next, MediaPipe can be used to detect facial landmarks, which can be used to determine the location of the eyes. Finally, PyAutoGUI can be used to move the mouse cursor to the location of the detected eyes.

The report includes the step-by-step instructions for installing and setting up the required libraries and modules, as well as the code for implementing the eye detection system. Additionally, the report should include a discussion of the performance of the system, including any limitations or challenges encountered.

Overall the report covers the following points:

* Introduction to Eye detection and its importance
* Tools and libraries used
* Installation and setup instructions
* Code snippets and their explanation
* Results and performance evaluation
* Limitations and future work.

**Introduction to Eye detection and its importance:**

Eye detection is a process in which a computer program is able to locate and identify eyes in an image or video. This technology is important for a variety of applications, including security and surveillance, human-computer interaction, and assistive technology for individuals with visual impairments. Eye detection can be used to authenticate a person's identity, track their gaze to understand their attention, and even control devices through eye movements. Additionally, it can be used in robotics and autonomous systems to help them understand and interact with their environment. Overall, Eye detection plays a critical role in many areas of technology and its importance continues to grow as the technology develops.

**Tools and libraries used:**

Tools are used for this project are given below;

1. Pycharm
2. Webcam

CDM system has been implemented using the libraries that given below;

1. OpenCV
2. MediaPipe
3. PyAutoGUI

**Installation and setup instructions:**

Installing Python on a computer can be done in several ways, depending on the operating system and desired setup. Here are the basic steps for installing Python on Windows computer:

1. Download the latest version of Python from the official website: https://www.python.org/downloads/
2. Run the installer for windows operating system.
3. During the installation process, make sure to select the option to add Python to the system PATH. This will allow you to run Python from the command line or terminal.
4. Once the installation is complete, open a command line or terminal and type "python" to start the Python interpreter.
5. To check the version of python installed, type "python --version" or "python -V"
6. To install additional packages and libraries, you can use pip package manager by running "pip install <package\_name>"
7. To start coding, we used pycharm to write python code and then run it .

It's important to note that Python 2 and Python 3 are different versions of the language and may not be fully compatible with one another. It’s need to decide which version to install based on the specific requirements of project or application.

**OpenCV** (Open Source Computer Vision) is a library of programming functions primarily aimed at real-time computer vision. It can be used for a wide range of applications, including object detection, image recognition, video analysis, and more. Here are the basic steps to set up OpenCV in Python:

1. First, make sure you have Python and pip installed on your system. You can check this by running "python --version" and "pip --version" in the command line or terminal.
2. To install OpenCV, you can use pip by running the following command: "pip install opencv-python"
3. To install openCV with contrib modules, you can use pip by running the following command: "pip install opencv-python-contrib"
4. After the installation is complete, you can import the OpenCV library in Python by running "import cv2"

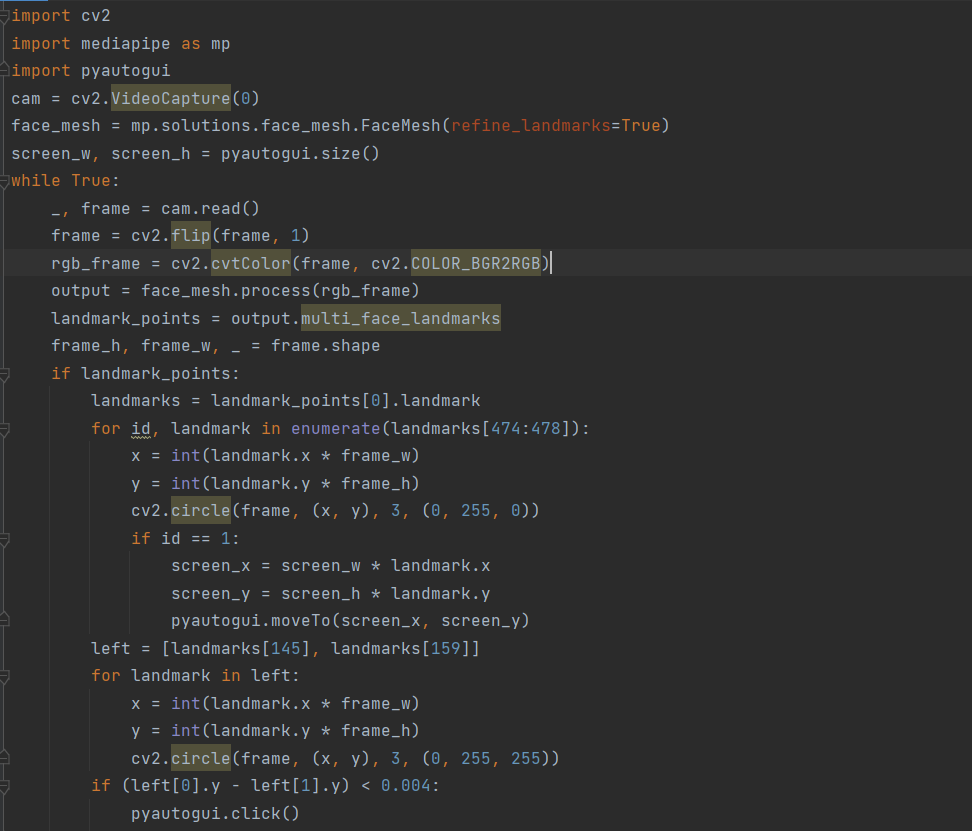
**MediaPipe** is a cross-platform framework for building pipelines to process multimedia data. It is developed by Google and can be used for various computer vision and machine learning tasks such as object detection, facial landmark detection, and hand tracking. Here are the basic steps to set up MediaPipe in Python:

1. First, make sure you have Python and pip installed on your system. You can check this by running "python --version" and "pip --version" in the command line or terminal.
2. MediaPipe is based on Bazel, a build tool developed by Google, it is necessary to install Bazel on your system before installing MediaPipe
3. MediaPipe also requires to have protobuf and numpy library installed, it can be installed by running "pip install protobuf numpy"
4. To install MediaPipe, you can use pip by running the following command: "pip install mediapipe"
5. After the installation is complete, you can import the MediaPipe library in Python by running "import mediapipe"

**PyAutoGUI** is a Python library for automating mouse and keyboard actions. It allows you to control the mouse and keyboard to perform actions such as clicking, typing, and scrolling. Here are the basic steps to set up PyAutoGUI in Python:

1. First, make sure you have Python and pip installed on your system. You can check this by running "python --version" and "pip --version" in the command line or terminal.
2. To install PyAutoGUI, you can use pip by running the following command: "pip install pyautogui"
3. After the installation is complete, you can import the PyAutoGUI library in Python by running "import pyautogui"
4. PyAutoGUI uses the screen resolution and color depth of your primary monitor, so you should be aware of these settings before using it. You can use pyautogui.size() to check the screen resolution, and pyautogui.screenshot() to take a screenshot of the whole screen.
5. PyAutoGUI also requires to have pillow library installed, it can be installed by running "pip install pillow"

**Code snippets and their explanation:**



**Results and performance evaluation:**

In our test cases CDM works pretty well. It detected eyes very preciously. It didn’t fail for single time in our test cases which is more than 50 times. The person need to be very focused to operate this system. Due to it’s very precise action we need to be very careful to using it. It will perform click by the blink of the user’s eye. Even a eye blink can do a faulty action. Due to it’s very precise action it is a very powerful system tool, but misuse of this tool can be costly.

**Limitations and future work:**

There are several limitations to current eye detection methods in Python. One limitation is that many existing algorithms rely on specific assumptions about the appearance of eyes in an image, such as their location relative to the face or their size and shape. This can make it difficult for the algorithm to detect eyes in images with unusual or difficult lighting conditions or in images of individuals with unique facial features.

Another limitation is that many existing algorithms require a large amount of training data to achieve good performance, which can be difficult to obtain in some cases. Additionally, current methods are not robust to variations in head pose, lighting conditions and occlusions.

Future work in this area could focus on developing algorithms that are more robust to variations in image conditions, as well as exploring alternative feature representations and machine learning techniques. Additionally, more research is needed on how to effectively incorporate contextual information, such as the presence of other facial features or the overall scene context, into the detection process.