



Bilkent University

Department of Computer Engineering

Senior Design Project

Third Eye

Final Report

Alkim Önen, İbrahim Eren Tilla, Göktuğ Öztürkcan, Berke Oğuz, Safa Alperen Oruç

Supervisor: Fazlı Can

Jury Members: Can Alkan and Çiğdem Gündüz Demir

Final Report

Apr 30, 2021

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

Contents

1. Introduction	2
2. Requirement Details	3
2.1. Functional Requirements	4
2.2. Non-Functional Requirements	5
3. Final Architecture and Design Details	6
3.1. Overview	6
3.2. Subsystem Decomposition	6
3.3. Hardware/Software Mapping	7
4. Development/Implementation Details	8
4.1. Human Detection	9
4.2. Distance Detection	10
4.3. Mask Detection	10
4.4. Human Tracking	10
4.5. User Interface	11
5. Testing Details	11
5.1. Verification	12
5.2. Validation	13
6. Maintenance Plan and Details	14
6.1. Bug Fixing	14
6.2. Hardware Maintenance and Optimization	14
6.3. Updates	14
7. Other Projects Elements	15
7.1. Consideration of Various Factors in Engineering Design	16
7.2. Ethics and Professional Responsibilities	17
7.3. Judgements and Impacts to Various Contexts	18
7.4. Teamwork Details	19
7.4.1. Contributing and functioning effectively on the team	19
7.4.2. Helping creating a collaborative and inclusive environment	19
7.4.3. Taking lead role and sharing leadership on the team	20
7.4.4. Meeting objectives	20
7.5. New Knowledge Acquired and Applied	21
8. Conclusion and Future Work	22
8.1. Conclusion	22
8.2. Future Plans	23
9. User Manual	31
10. Glossary	35
References	36

1. Introduction

A year ago, most of us did not know about a term which we are using in everyday life nowadays: social distancing. With the emergence of the COVID-19, all our lives had changed, and we had to adapt accordingly [1]. Although the virus itself affected the whole world immensely, both in an economic and in a social sense, the world keeps spinning and people keep on living with their lives [2]. With it comes the need for social distancing.

Social distancing, in its most basic form, is to refrain from getting close with other people in crowded spaces. Research about social distancing proved it to be useful in ceasing the spread of the virus to a crucial point. It is believed that 3-4 months of moderate social distancing can save up to 1.7 million lives and \$8 trillion or \$60,000 per US household [3]. Thus, it is of utmost importance to realize the role of social distancing in fighting COVID-19.

Although the benefits of social distancing are undoubtable, there are still many people that do not care about it and keep on violating spacing rules and advice in crowded spaces [4]. This brings the need for a regulation system for the crowds and that is where *Third Eye* steps in. We, as a group, were not pleasant with this situation and wanted to do more than complying to the regulations which is why we started this project. With *Third Eye*, we aim to put this problem on the table and come up with a viable solution that will minimize this issue. Hence, we came up with a product for business owners to track and apply the regulations.

Third Eye's main feature is detection of social distancing violations. It uses human detection methods and linear interpolation to locate and predict real life distances between humans. Then, it alerts the users if the distances between some people violate the regulations. However, the program also analyzes data to figure which people may have relations. This way families and friends are differentiated from strangers. *Third Eye* also equips a mask detection system so that users could detect whether people are wearing masks or not.

However, *Third Eye* does not only aim to regulate social distancing but to provide useful data to the users as well. It will analyze the crowds and where the people's interest peaks. Then it will inform the users how the crowd moves, and which parts of the market are trending. For example, a shopping center overseer will be able to check out the data that *Third Eye* offers and see the trending sections of the mall.

In this report, the final state of *Third Eye* will be presented, providing the details of its architecture and its design choices. First, we will state the requirement details of our project, where the finalized functional and non-functional requirements of the program will be given. Later, the final architecture of the project will be fleshed out using a subsystem decomposition and the hardware/software mapping will be presented. Then, the development details of the project will be disclosed. These will include the main focal points of our project: human detection, distance detection, mask detection and human tracking system. After that, the testing details will be given regarding their types. We will continue with examining the details of the maintenance plan. This will be continued with the listing of the other project elements, such as consideration of various factors in engineering design (public health, public safety, global factors etc.), ethics and professional responsibilities, judgements and their impacts, teamwork details (contribution and functioning effectively) and the new knowledge acquired/applied throughout the project. Afterwards, we will present the conclusion and the future work for *Third Eye*. Finally, a “User Manual” will be given at the end of the report which contains the screenshots and the workflow of our project.

2. Requirement Details

In this section, both the functional and non-functional requirements of *Third Eye* will be provided. The functional requirements are the specifications of the intended behavior of the system, where we have divided our project into two main categories: system and user functionality. The non-functional requirements, on the other hand, help us to ensure the usability and effectiveness of the whole system, where we have listed many attributes to cover.

2.1. Functional Requirements

2.1.1. System Functionality

The system should:

- receive the camera feed accordingly to where it will be used.
- adjust the “social distancing” distance according to the camera’s height.
- actively process the input feed and be ready to alarm the user if the calculated social distancing rules are violated.
- compare the data (via object tracking system) if multiple connected camera feeds are given to the system.

- analyze and store the data where the user will be able to see which parts of the covered area is being populated most.
- be able to work with possible external alarming systems (such as a megaphone in a mall).
- work on a Windows OS (Possibly Unix variants etc.).

2.1.2. User Functionality

The user will be able to:

- select whether to analyze through a pre-recorded footage or a live feed.
- select whether to regulate social-distancing or masking requirements.

2.2. Non-Functional Requirements

2.2.1. Reliability

The system should:

- correctly identify people and the number of people in a crowd in order to generate correct outputs.
- work with a variety of different video formats.
- work with different color ranges [5].
- be able to differentiate the human body from the environment in different situations or places [5]. It shouldn't confuse other living and non-living beings with the human body.
- not collect and store the imagery without the knowledge of the crowd. If saving the data is the case, there should be warnings about the situation or consent from the crowd should be taken by the users.
- not confuse already formed groups (families, couples, etc.) with social distance violations and show them in a distinctive way.

2.2.2. Extensibility

The system should:

- be sustainable and maintainable, in other words, should be open to updates.

2.2.3. Usability

The system should:

- be user-friendly. Users should be able to perform the actions easily.
- give clear outputs. The outputs should be self-explanatory. Users should not be confused with the results.
- work with different formats of video input or convert given formats into the desired one. This way the user is not bothered with that.

2.2.4. Accessibility

The system should:

- be manually installed to computers according to the needs of the customer.

2.2.5. Efficiency

The system should:

- work in real-time in order to identify the violations of social distancing in an area so that the authorized people can act on the situation immediately.
- analyze the data and return results in a short time frame so there will be minimal delay between the results and the real-time.
- use minimum resources. It shouldn't allocate too many resources.

3. Final Architecture and Design Details

3.1. Overview

In this part of the final report, the subsystem decomposition, hardware/software mapping, design decision and change details will be explained in detail. Subsystem decomposition, layers and structure of the project will be explained.

3.2. Subsystem Decomposition

As it is planned at the high level design phase of the project, we implemented the project according to 3-tier architecture. Therefore, we could break the project into structures

and this made the implementation phase more easier and faster for our team. The subsystems that we mainly focus on were application-tier and presentation-tier since the data tier only includes a local storage for statistical storage. In the application layer we focused on human tracking, mask detection etc... At the presentation tier, we designed the GUI of the project. *Third Eye* is a desktop application, therefore, the design of the UI was developed with the QtDesigner which is a Python based GUI platform.

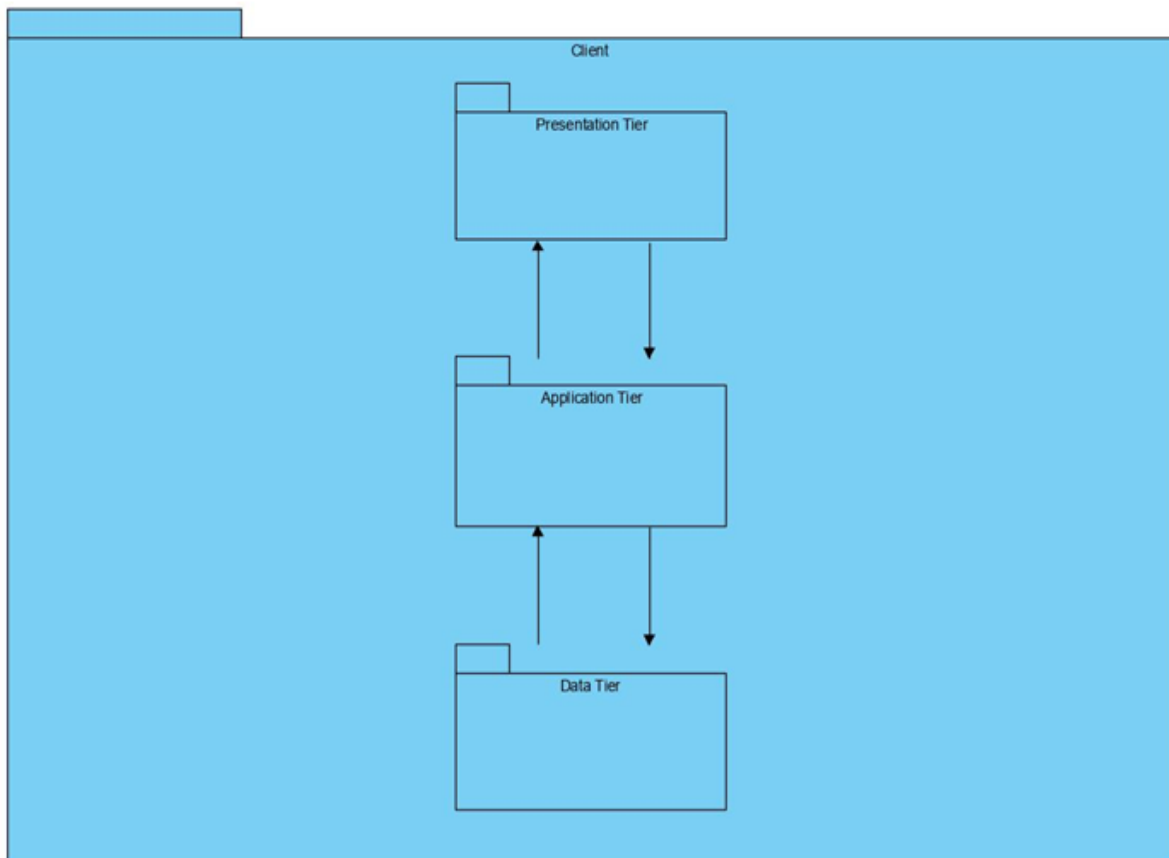


Figure 1: Subsystem decomposition diagram

3.3. Hardware/Software Mapping

In the software part, we stick to our promises on the high level design phase and implement our desktop application with Python. We used YOLOv4 for object detection and tracking which is a well known, open source computer vision library of Python. Moreover, we integrated it with the “darknet” which is explained more detailed in the implementation details section.

We have planned to use Tkinter for GUI, however, as a design decision change we used QtDesigner which is again a python based GUI platform. However, as a reason for this design decision change, we thought that QtDesigner provides more professional options. QtDesigner provides various UI components ,and also, it has coding flexibility compared to Tkinter.

Another design decision update is about hardware. We have planned to use JetsonNano as a hardware device which is an Nvidia product that provides accurate object detection. However, we achieve an accurate object detection with the open source software libraries and the algorithms that we have developed. Therefore, we decided not to use JetsonNano in the project.

The project can work on the already existing video and image, however, hardware device it is welcome to use real-time camera feed. It can also work by connecting it with an external camera. The project can work on MacOS, Linux and Windows operating systems on desktop.

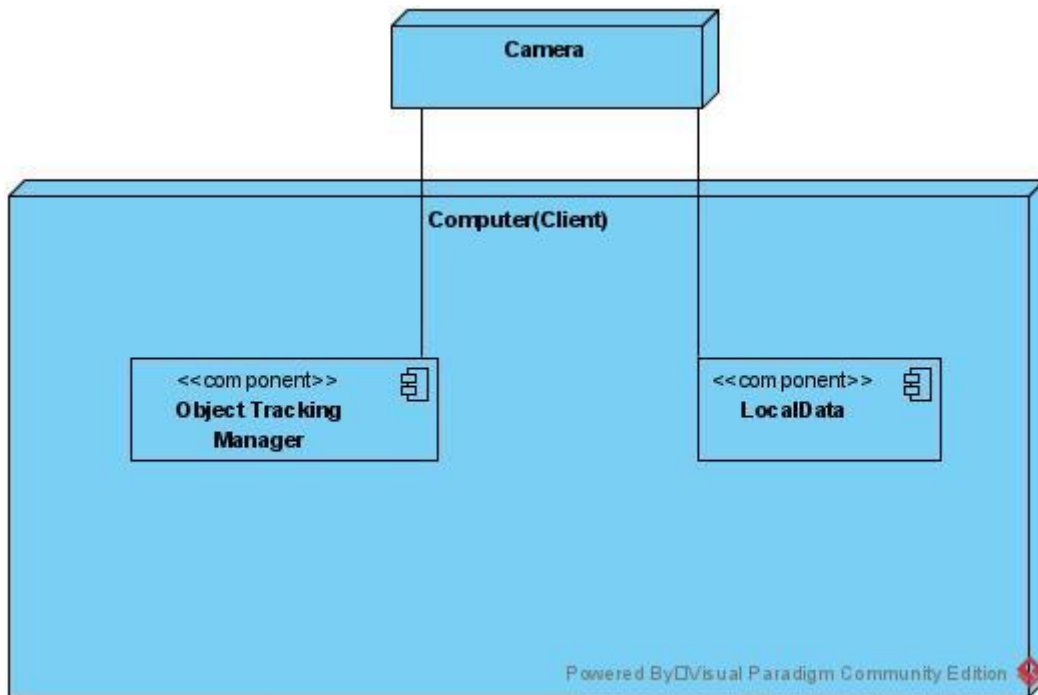


Figure 2: Hardware and software connection diagram

4. Development/Implementation Details

Third Eye is implemented using Python 3. The backend of *Third Eye* uses YOLOv4/Darknet [9]. Darknet is an open source neural network framework written in C and CUDA. Our implementation uses YOLOv4 weights provided by Darknet along with Opencv and tensorflow for human detection. Earlier iterations were done using the YOLOv3 however during the development stage YOLOv4 was released then we decided to upgrade the project with the newer weights. Front-end of *Third Eye* works with PyQt5 which also uses python. Firstly, the frontend was designed in respect to the mockups that were delivered in the Analysis Report. However slight changes and alterations to the overall theme and the looks of the app was done over time.

In the implementation stage of *Third Eye* first the backend and the frontend was constructed separately. For the backend first C was being used since we believed it would work the fastest. However after testing in Python and seeing it worked faster we decided to switch to Python. Also mask detection and human detection was implemented separately since human detection was implemented using pre-trained weights however mask detection was done/trained from scratch.

During the implementation Github was used to coordinate the works by merging and branching.

Development of *Third eye* consisted of 4 main components which can be reduced to the following subsections,

- Human Detection
- Distance Detection
- Mask Detection
- Human Tracking

4.1. Human Detection

For human detection *Third Eye* uses YOLOv4/Darknet's pre-trained weights. Firstly OpenCVs human detection was going to be used however it fell a little bit short and it was scratched. During development with YOLOv3, YOLOv4 weights was released and we decided to upgrade to YOLOv4.

Tracking humans in a still picture is doable with a CPU all it takes is a few seconds to process the frame. However, for a video it takes painfully long since a video usually has 30 frames per second. Which means it would take around 2 minutes just to process 1 second of information in real time. To overcome this we decided to use a GPU for human detection with CUDA. Which is much faster and allows us to process up to 20 frames per second, with an average of 16 frames per second (Using a GTX 1080).

To use the pre-trained YOLOv4 weights with python we used Tensorflow to process them. This allowed us to work in Python which also worked faster than the version we used on C.

Third Eye can run social distance detection in both live footage and with pre-recorded video. For the live version, the application reads camera feeds from the system and processes the footage and works in real time. For pre-recorded footage, users can select video in their system. However, if the footage resolution is too high, processing the footage may give lower FPS (around 15) than usual. In order to solve this problem, we implemented a frame skip option which calculates how much time each frame takes to process and skips the next frame or frames accordingly.

4.2. Distance Detection

In order to calculate the distances between humans in videos, simply using pixel distances would have resulted in faulty results as the scale of people closer to camera and people further to the camera are not equal. We had to create a calculation system that determines the real positions. For this purpose, we used linear interpolation techniques. First we get the following information from the user: camera height, camera look angle, camera view angles in horizontal and vertical axis. As we also have every frame's width and height information, we create an imaginary coordinate system by taking the center point of the image as origin. Then we calculate the distance of origin to the camera. Afterwards, we use given variables with trigonometry and simple linear interpolation to predict the real life location of a detected human. We locate each human in a frame and calculate the distances to each other using Euclidean norm. However, this method is designed for flat horizontal surfaces. If the surface is tilted or there are serious height differences in the ground, errors may increase.

There is a possibility that the user may not be familiar with camera's settings. If camera settings are not adjusted well, detecting positions with interpolation may not give

correct results. In this case, *Third Eye* predicts the distance between two people from the heights of their bounding boxes.

4.3. Mask Detection

For the mask detection algorithm, we used YOLOv4/Darknet structure and trained the program using a dataset for mask detection that we downloaded from Kaggle. This dataset included three directories: test, validate, train. They all contained the respective images and their corresponding text (.txt) files that include the configurations for YOLOv4 to process through. The training operation was done on a Google Colab VM and lasted around 5 hours. In the process, we have produced many weights and selected the best one (the one with the lowest average loss and highest precision rate) for our program.

4.4. Human Tracking

Human tracking systems are important for holding statistics about human intensity , and also detecting people who are relatives. System tracks the people on the camera so that it does not count the same person as a new human for every frame change.

Human tracking uses YOLOv4s human detection and as an algorithm it uses DeepSORT. It is an algorithm that shows remarkable results for real-time multi object tracking. It works mainly in two phases which is detection and association. At detection phase, it detects humans in a frame. In the association part, it does similar detections to the previous frame so that in the next frames humans from the previous frame are predicted and this procedure followed through the whole video to get the result.

4.5. User Interface

GUI part of the project implemented with PyQt5. QtDesigner provides various options and easy drag and drop opportunities. However, determining the colors of the button was a little bit tricky for the free version. Therefore, we had to implement the button colors with the code segments that are available at the preferences option for every button.

Moreover, we use photoshop to add images like camera and eye which gives a more fancy view to the UI page. Also, the background image is implemented with photoshoping.

5. Testing Details

5.1. Verification

We have done the verification before the execution of the code in order to find the bugs development cycle. The fundamental steps of the verification are followed as it is given below.

5.1.1. Inspection

As team members we have taken different technical elective courses which makes it easier to share the work, and also the work that is done becomes more relevant. After those works are shared accordingly, we planned some meetings to fix the works assigned to all team members.

Each team member detected the simple bugs and recorded them in the parts they know better which makes the implementation with less errors and bugs.

5.1.2. Analysis

Analysis part includes the mathematical model and calculations that we reviewed before the coding. Reviewing the FPS calculation formula and the distance calculation methods from the camera feed is the example of this step.

5.1.3. Demonstration

It is the less detailed data gathering part of the verification. In this part, the system is manipulated as it is intended to be used to verify that the results are as planned or expected. As an example, before we started to implement human detection, we assumed that human tracking would work well with the human detection algorithm.

5.1.4. Test

The test inputs are given to the program. Our aim is to work the program on the real-time video systems. However, for this step we considered some photos as inputs.

5.2. Validation

In the validation part we have done some different tests to the application that is explained in detail with the subtitles below.

5.2.1. Performance Testing

Third Eye works on real time camera feed, therefore, performance is one of the most important parts of the project. FPS of the camera feed was the part that we care about the most. We did the testing with a code segment that we have implemented to the frame part. It calculates the frame per second for us and we obtained the results. At the first phases of implementation, we realized that FPS of the camera feed went down to 3 FPS. Therefore, we decided to review and refactor our detection algorithms to get higher performance. After some refactoring, we have done the performance test again to the camera feed, which is working with the detection algorithm, and we found out that FPS of the feed could go up to 20-25 FPS.

5.2.2. Requirements Testing

Based on the requirements decisions listed by the team members from the last semester at the project design phase, we have tested the requirements whether they are working without errors. Moreover, we have checked the requirements documentation to make sure that important requirements are implemented in the project as we wish. Besides the main human tracking, some side functionalities like statistics and mask detection is an example of some requirements that we have considered.

5.2.3. Unit Testing

One of the black-box testing that we have used in the project. We test some functions of the software as units like the “object tracking” algorithm function. We detect the bugs and defects on the code and eliminate them in order to get the correct results from different functions.

As an example, as we first wrote the object tracking algorithm, the tracking system was recognizing the trees as humans. After focusing on the particular function implementation, we eliminate that bug.

5.2.4. Integration Testing

Another useful black-box testing that we have applied in the project. When the code parts are shared among team members and the unit functions are integrated, it is inevitable that the system has some bugs. Therefore, after integrating those parts, we have manually tested those parts in order to improve the coding phase cleaner.

5.2.5. Testing the UI

UI buttons, page redirects and functionalities are tested manually by team members. Also, UX concerns are considered and we tried to implement the application as much as user friendly. Therefore, after testing, we have made some changes on it like improving the button sizes, button places...etc.

5.2.6. User Acceptance Testing

In order to evaluate the project from users perspective, we have shown it to our friends and family for their ideas. Moreover, at the end of the implementation, we will present the project to our supervisor and jury members. Their feedback on the project will determine whether the project is ready for deployment or it needs some changes or improvement before the demo on CSFair.

6. Maintenance Plan and Details

Since cameras and hardwares are changing day by day, maintenance is a must for *Third Eye*. Our maintenance plan is based on three subtitles, these are, bug fixing, hardware maintenance and optimization and updates.

6.1. Bug Fixing

Bug fixing is a continuous process for our project because after every new version, there will be bugs. After the deployment, if there is a bug found by the users or the customers, we as a development team will make some implementations for fixing these bugs.

6.2. Hardware Maintenance and Optimization

For human detection, we use GPU human detection with CUDA which is much faster than CPU and allows us to process up to 20 frames per second. We tested *Third Eye* with a Gtx-1080. There are various types of cameras so we need to get feedback from the user. According to that feedback, we can change the hardware and optimize the hardware with respect to the camera. For greater frames per second, the system requires fairly decent hardware so that we need to find an optimal hardware for the user's camera. If the user's camera changes, we may need to change the hardware too.

6.3. Updates

Change is always a part of the software projects so that there will be updates according to the environments and libraries that are used. For example, we are using YOLOv4/Darknet for the backend of *Third Eye* but we will update it to the YOLOv5 in the future. Also we will do updates according to the users feedbacks, we can add new features, switch our features or change our user interface.

7. Other Projects Elements

7.1. Consideration of Various Factors in Engineering Design

There were limiting factors in our project that required an extensive discourse. Below are some of those factors that limited our design and brief discussions on the potential effects of these factors.

7.1.1. Public Health

Public health is the key factor in our project. The idea behind *Third Eye* is to detect and prevent social distancing violations in order to resist the fast paced spread of COVID-19. Similarly, wearing masks also reduces the mobility of the virus [10]. Also, it will be a good option to use in possible future virus pandemics. This factor does not have any limitations to the usage of the application. Instead, it diverts people into using *Third Eye*.

7.1.2. Public Safety

There is an indirect effect of *Third Eye* to public safety. While *Third Eye* gives warning for people too close to each other, it is possible to predict and reduce the potential crime practices like pick-pocketing. However, there is no limitation for the application itself.

7.1.3. Public Information Security

Third Eye does not collect any private information about the people who are captured by the camera while tracking the social distance violations. It also detects face mask usage instead of applying face detection or recognition. *Third Eye* does not record any information. The statistics about commercial causes will be about human count which is only about the numerical crowd statistics. Therefore, it can be said that the application respects personal privacy while detecting humans and collecting statistics.

Apart from that, information security also refers to rules, regulations and norms that nearly every country comes to mutual agreement on. Such factors include well-defined standards like General Data Privacy Regulation (GDPR) [11]. However, for safety and security issues, businesses and governments have the authority to surveil the area as long as they notify the people [12]. For this case, *Third Eye* doesn't take any accountability as this is the responsibility of the users themselves.

7.1.4. Global Factors

COVID-19 is a global issue which makes our project a global solution. It can be used in every place in the world to detect the social distance between people. Nonetheless, the current design of *Third Eye* doesn't have multi language support and is only limited to english.

7.1.5. Public Welfare

Third Eye aims to access people from anywhere with security cameras such as shopping malls, airplanes, and hospitals. It is important for us to offer this application to users in all kinds of economical level businesses without any economical drawbacks. Hence, *Third Eye* will not impose economic insecurities and will be free of use.

7.1.6. Social Factors

As we mentioned, *Third Eye* is a social distance tracking project. It aims to calculate the distance between people and give a warning about it. However, even if the importance of social distance is well known by people, they can choose to hang out with their family members or friends. People prefer to maintain their distance from strangers. We considered this issue and planned to bring a solution to this. AI will detect the relation between people from their gestures and evaluate them as a group. Therefore, *Third Eye* will not give a warning to those people who are willing to choose to hang out together.

	Effect Level	Effect
Public Health	10/10	Reduce the mobility of COVID-19 virus
Public Safety	3/10	Increase security violations
Public Information Security	7/10	Application concerns about the data and identification privacy
Global Factors	8/10	It will be a solution to the fast paced spread of global pandemics such as COVID-19
Public Welfare	5/10	<i>Third Eye</i> can be built into any system free of charge.
Social Factors	4/10	<i>Third Eye</i> aims to prevent all social distancing violations, however people who prefer to hangout close to each other like families and friends will not be directly warned.

Table 1: Factors that can affect analysis and design

7.2. Ethics and Professional Responsibilities

Third Eye is a smart surveillance tool that uses surveillance cameras for enforcing social distancing between people and analyzing the crowds where the people's interest peaks for commercial purposes. Keeping people under surveillance is a sensitive subject instead of privacy. It is our ethical responsibility to protect people's privacy. In our project, human detection algorithms detect humans anonymously, not by identification so this prevents many possible privacy problems for the project.

In many countries, the security of personal information is protected by laws. In Turkey, there is a protection of personal data law which is known as KVKK, according to this law, it is a professional responsibility for us to keep personal data private and not to share this data with any third party organization. We also considered personal data security for ethical responsibility. We do not share any personal data with a third party in any condition. When we need statistical data, the information we keep can be used by the user only and it is important that this information cannot contain any personal data.

As a team, it was our Professional responsibility to hold meetings weekly where we discussed further plans and went over what has been done in the past weeks. The workload and the future of the project was discussed democratically in these meetings and the workload which we decided in these meetings was distributed fairly. Similarly, key decisions about the techniques and tools used in the project were all decided as a team in important breakpoints of the project.

It was our ethical responsibility to check licenses for the APIs, libraries, and tools before usage. All of the libraries and techniques we used were either our own or from open_source projects with open licenses. Also, we did adhere to the Code of Ethics and General Data Protection Regulation (GDPR) [11]. Since there is surveillance through cameras wherever *Third Eye* is being used, as an ethical responsibility, there need to be warning signs put up and cameras placed accordingly to the regulations which is the responsibility of the users[12].

7.3. Judgements and Impacts to Various Contexts

Judgement Description	Running system in real time with great performance	
	Impact Level	Impact Description
Impact in Global Context	5/5	Since the system is designed for real time use, it is important for real time human detection and interpolation.
Impact in Economic Context	4/5	For greater frames per second, the system requires fairly decent hardware.
Impact in Environmental Context	0/5	This judgement of <i>Third Eye</i> doesn't have any environmental impact.
Impact in Societal Context	2/5	<i>Third Eye</i> has a medium impact in societal context as it searches for violations in distancing while it can also be used for safety measures.

Table 2: First judgement and its impacts

Judgement Description	The system should be free and available to everyone	
	Impact Level	Impact Description
Impact in Global Context	5/5	More uses of the system can reduce the spread speed of COVID-19.
Impact in Economic Context	4/5	All businesses is different sizes could use the system without the cost
Impact in Environmental Context	0/5	<i>Third Eye</i> doesn't have any environmental impact.
Impact in Societal Context	2/5	This judgement has a medium impact on societal context as the system could increase safety in several places.

Table 3: Second judgement and its impacts

7.4. Teamwork Details

7.4.1. Contributing and functioning effectively on the team

Teamwork was critical in the development process of *Third Eye* as correct teamwork can provide more successful project outcomes and higher profits. Thus, we put great emphasis on the distribution of workload and do our best to circulate the roles and responsibilities for different parts of the project evenly. Since we worked on our project in the quarantine period, the work distribution was even more important for our mental and physical health.

We assigned a leader role for the different parts of the project to each member in the project plan so that it increases team member's contributions and functions effectively on the team. For contributing and functioning effectively, team members focused and trained during the implementation process. Each team member is trained for their corresponding position specifically for the project. The training includes the research of algorithms and methods specifically designed to solve problems that are in the scope of this project. After training, every team member contributed the project and function effectively on their part. If any team member had an issue we set up a meeting and did collaborative work.

7.4.2. Helping creating a collaborative and inclusive environment

In order to create a collaborative and inclusive environment, we used a number of technologies at our disposal.

- Jira: We used Jira for the collaboration and to manage the workload of our project, as well as addressing any possible problems through the process.
- Github: We used GitHub as the repository of our project as well as our version control system. This made our implementation stage more organized and provided a more collaborative codebase.
- Zoom: We arranged our meeting on the Zoom platform. We utilized it mostly on discussions and feedback parts using the platform's share screen and annotate features.
- Discord: During the implementation, besides Zoom we used Discord in order to communicate with each other. Its advanced message system was a great help while talking to each other for both sharing information and saving resources for later uses.

7.4.3. Taking lead role and sharing leadership on the team

Project leaders assigned for different parts of the project were responsible for setting the expectations for their team members, their team performance and acting as a problem solver. Taking a lead role in a team made the leader to be more motivated and taking responsibility increased the leader's commitment to the project. For increasing every team member's commitment to the project, we decided to share the leadership on the team. More information about team leaders for different parts of the project can be found in our High Level Design Document.

7.4.4. Meeting objectives

After our High Level Design Report, we made some changes in our work packages. As we learned more about deep learning and computer vision in the development stage, we designed our plan in a more detailed way. We are nearly done in our Final Report and met all of the objectives until this point. Improved work packages can be seen below.

Work Package	Work Package Title	Leader	Members
1	Project Specification Report	-	-
2	Website	Eren	Alperen
3	Analysis Report	Berke	Göktuğ, Alperen, Alkım, Berke
4.1	Jetson Nano	Alkım	Göktuğ, Eren
4.2	Python libraries (YOLOv4)	Berke	Alperen, Alkım
5.1	High Level Report	Göktuğ	Eren, Berke
5.2	High Level Report	Eren	Alperen, Alkım
6	Human detection on still images	Göktuğ	Alkım, Eren
7	Select a camera	Berke	Alperen, Göktuğ
8.1	Real time human detection	Göktuğ	Alkım, Berke
8.2	Real time human detection	Eren	Alkım, Alperen
9.1	First Demo	Alperen	Alkım, Berke
9.2	First Demo	Göktuğ	Alkım, Eren
10.1	Low level Report	Berke	Alperen, Göktuğ
10.2	Low level Report	Alkım	Eren, Berke
11.1	GUI	Eren	Alperen, Göktuğ
11.2	GUI	Alperen	Alkım, Berke
12.1	Check distance/Interpolation	Alkım	Göktuğ, Eren
12.2	Check distance/Interpolation	Berke	Alperen, Alkım
13.1	Human relations	Göktuğ	Eren, Berke
14.1	Mask Detection	Eren	Alperen, Göktuğ
14.2	Detection Analytics	Alperen	Alkım, Berke
15.1	Optimizing The Project	Alkım	Göktuğ, Eren
15.2	Optimizing The Project	Berke	Alperen, Alkım
16.1	Final Report	Eren	Alperen, Berke
16.2	Final Report	Alkım	Berke, Göktuğ
17.1	Final Demo and Presentation	Berke	Alkım, Göktuğ
17.2	Final Demo and Presentation	Göktuğ	Eren, Alperen
17.3	CS Fair	Eren	Berke, Alkım

Table 4: Updated Work Packages

7.5. New Knowledge Acquired and Applied

At the beginning of the project, we discussed with our team and figured we had a lack of knowledge for building our project. We brainstormed for specifying which advanced techniques, external tools and technologies should we use for our project. We determined what methods would serve best for our learning curve. Afterwards, we acquired some knowledge, developed new skills and planned appropriate learning strategies to do so. Some topics that we centered upon are:

- Machine Learning
- Computer Vision
- Software Project Management

Human detection algorithms use computer vision and deep learning techniques. First, we studied the basics of computer vision. As we read more papers, we gained in-depth knowledge about state-of-art technologies in object detection. Next, we dove into deep learning as some parts like mask detection needed some training and validation requirements. During the implementation stage, we tested several different methods and selected the approaches and tools that are closer to our purpose.

For us, managing the project was very important. Therefore, in the beginning stages of the project, we researched and learned about software project management. It gave us perspective on how a software can be developed efficiently. Also introduced the concepts of project leader and team member to us.

Below are some methods we used to get more information about the topics above:

- Online Learning
- Literature Review

Reading papers was beneficial for us to get more familiar with computer vision concepts, methods, and technologies. It was also effective for understanding deep learning principles and architectures. We mostly read papers from computer vision and deep learning conferences like CVPR, ICCV, and NeurIPS. However, we also studied more casual options. Our second main strategy was online learning, since there were a lot of online tutorials and guides. Watching these tutorials significantly accelerated the development process of *Third Eye*.

8. Conclusion and Future Work

8.1. Conclusion

Our project *Third Eye* is up and working, it has human detection, distance detection, mask detection, human tracking features in it. Human detection feature can be used for both picture and the video. After detecting humans, *Third Eye* measures the distance between people by using distance detection. Mask detection feature detects people with/without masks. With *Third Eye*'s human tracking feature, we give every human a track id and we use this feature to detect people's relationship.

At the end of the project, we improved our research, coding, design, team-work skills. We are now familiar with CUDA, YOLOv4/Darknet, OpenCV, PyQt5 but we still need to improve ourselves. We will be happy to get more feedback, advice about our project.

8.2. Future Plans

In our Analysis Report, we said that *Third Eye* can be used for analyzing the crowds where the people's interest peaks for commercial purposes. *Third Eye* will collect the data and give the statistics of it. The statistics about commercial causes will be about human count which is only about the numerical crowd statistics. Therefore, it can be said that the application respects personal privacy while detecting humans and collecting statistics. Thus, with this feature, our project *Third Eye* can be used after the pandemic and it will be very beneficial for the commercial purposes.

According to the feedback, *Third Eye* will be changed. The bugs which are reported from the users will be fixed in the future. As we mentioned in the maintenance plan, *Third Eye* is using YOLOv4 and it can be updated to the YOLOv5 in the future and also there will be more updates according to bugs and feedback.

9. User Manual

Main Page

The page below is the welcome page of *Third Eye*. The options that users can choose are RealTime Social Distancing, Mask Detecting, Credits, and Quit.

RealTime social distancing is the option for tracking the social distance among people.

Mask Detecting is the option for detecting people whether they are wearing face masks.

Credits show the name of team members, supervisor and jury members.

Quit button is for exiting the application.

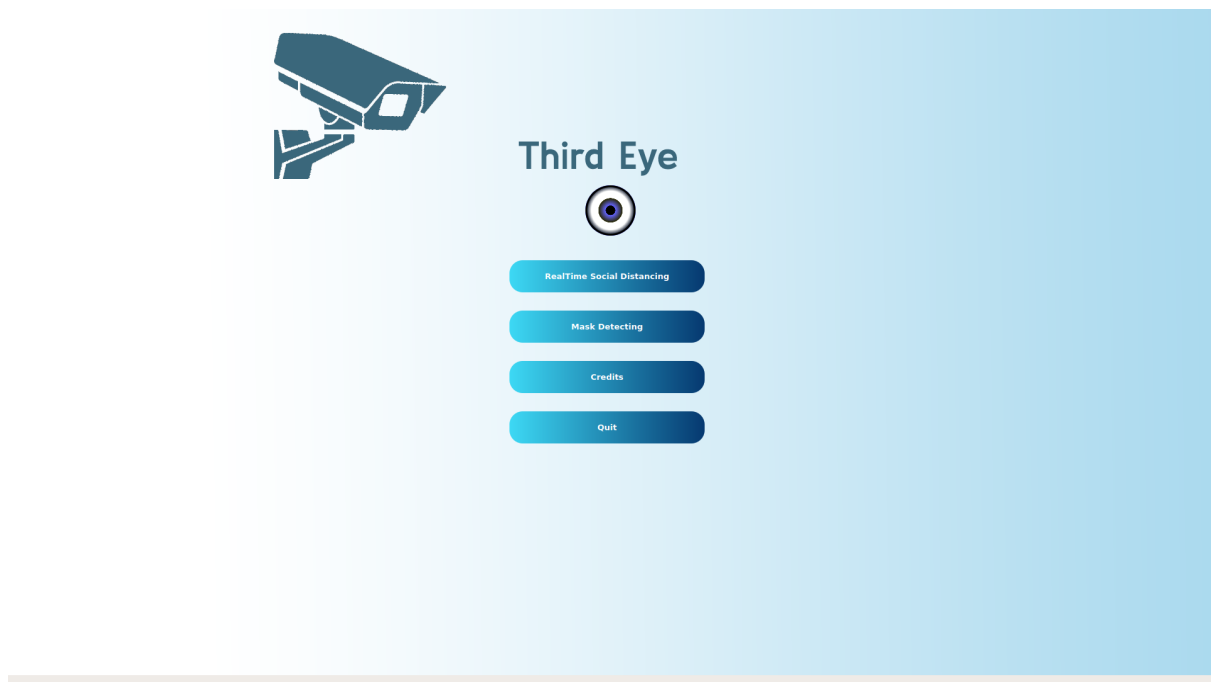


Figure 3: Main Page

Social Distancing Detection Menu Page

RealTime Social Distancing page provides Pre-recorded Videos and Live Video option to users. Back button turns the user back to the welcome page of *Third Eye*.

Pre-Recorded videos option directs users to another page which user can choose any recorded video to make the social distancing detections.

Live Video option does the same detection process with the live camera feed.

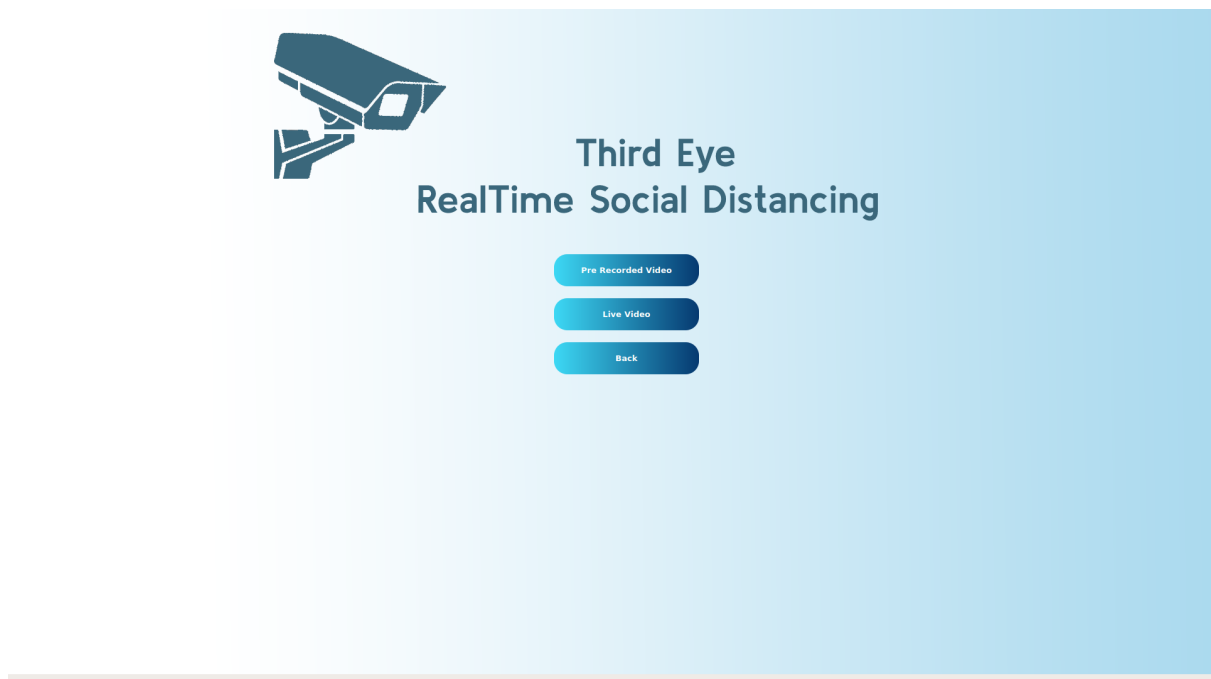


Figure 4: Social Distance Detection Menu Page

Pre-recorded Social Distancing Page

This page is for processing social distance detection on pre-recorded video footages. Users can select any recorded video from their local computers by pressing the Select button.

The settings portion is for adjusting camera settings. Camera height (cm), camera angle (degree) and view angle (degree) should be entered here for more precise results. Also, the checkbox below can be selected if real time processing is a priority. Otherwise, the results may be shown in lower FPS than the original footage if the resolution is too high.

Users can press the Run button to start the detection process. Back button sends the user to the Main menu.

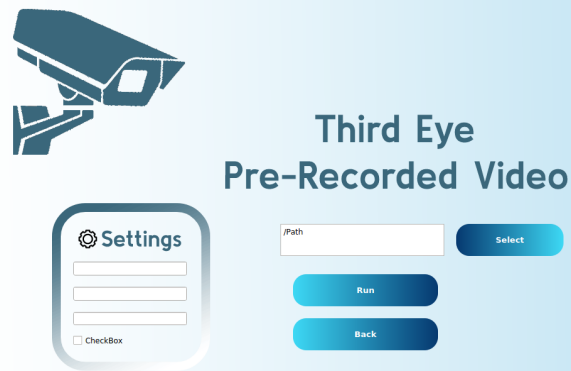


Figure 5: Pre-Recorded Social Distancing Page

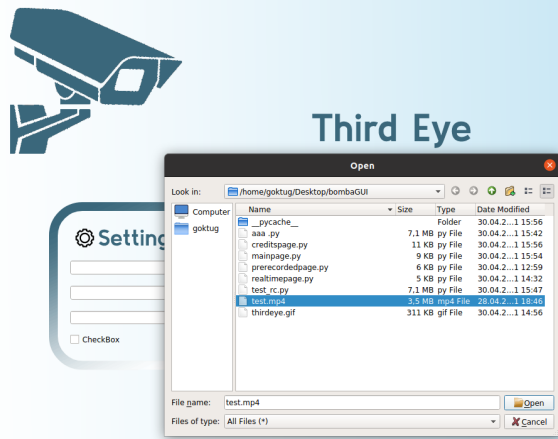


Figure 6: Selecting Video Footage From File Manager for Social Distance Detection

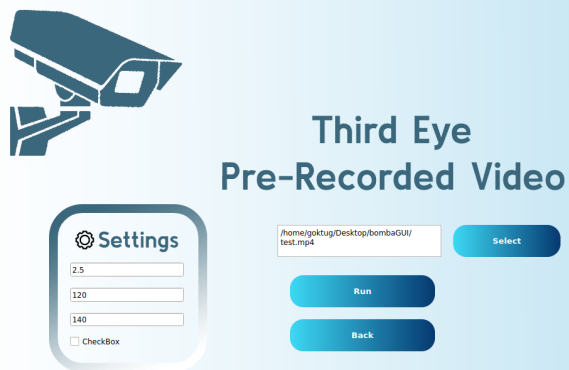


Figure 7: Adjusting Camera Settings

Social Distancing Detection on Video Footage

Social distance detection is the main functionality of *Third Eye*. The images below show some samples from video footage.

- The green boxes correspond to safe social distancing.
- Yellow boxes also correspond to safe social distancing because the people violating the distancing are recognized as possibly relatives/friends.
- Red boxes correspond to unsafe social distancing. The red line between people shows the people who violate the social distance.

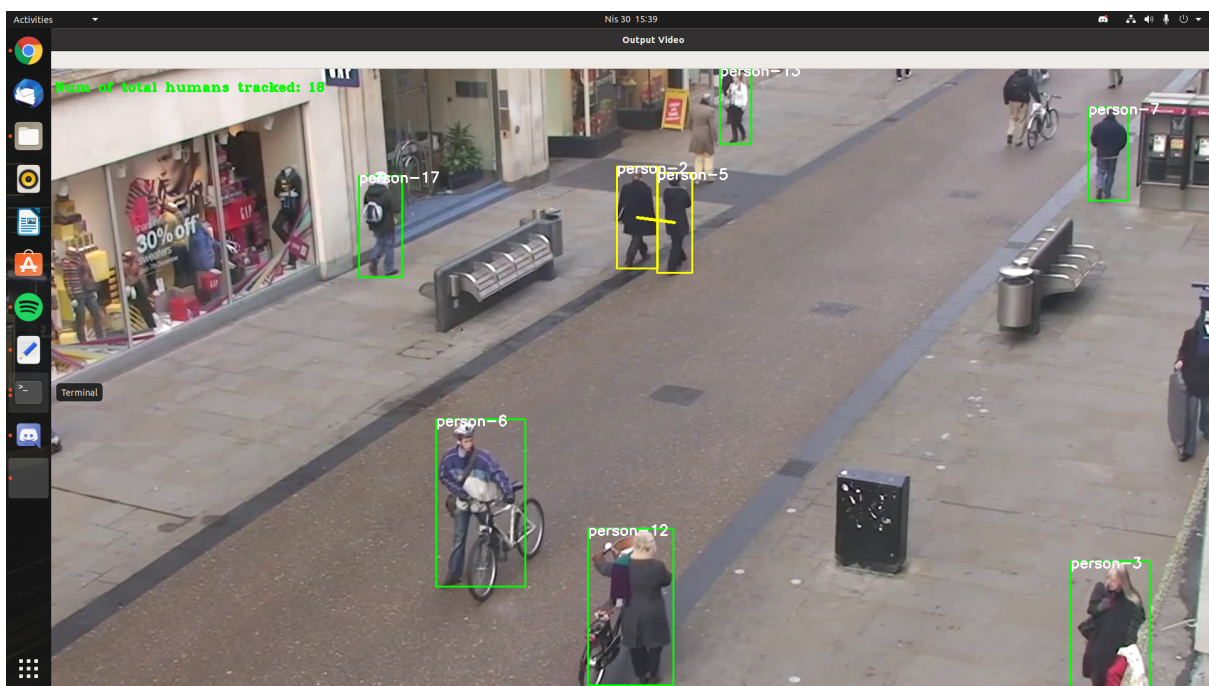


Figure 8: Social Distance Detection on Video Footage

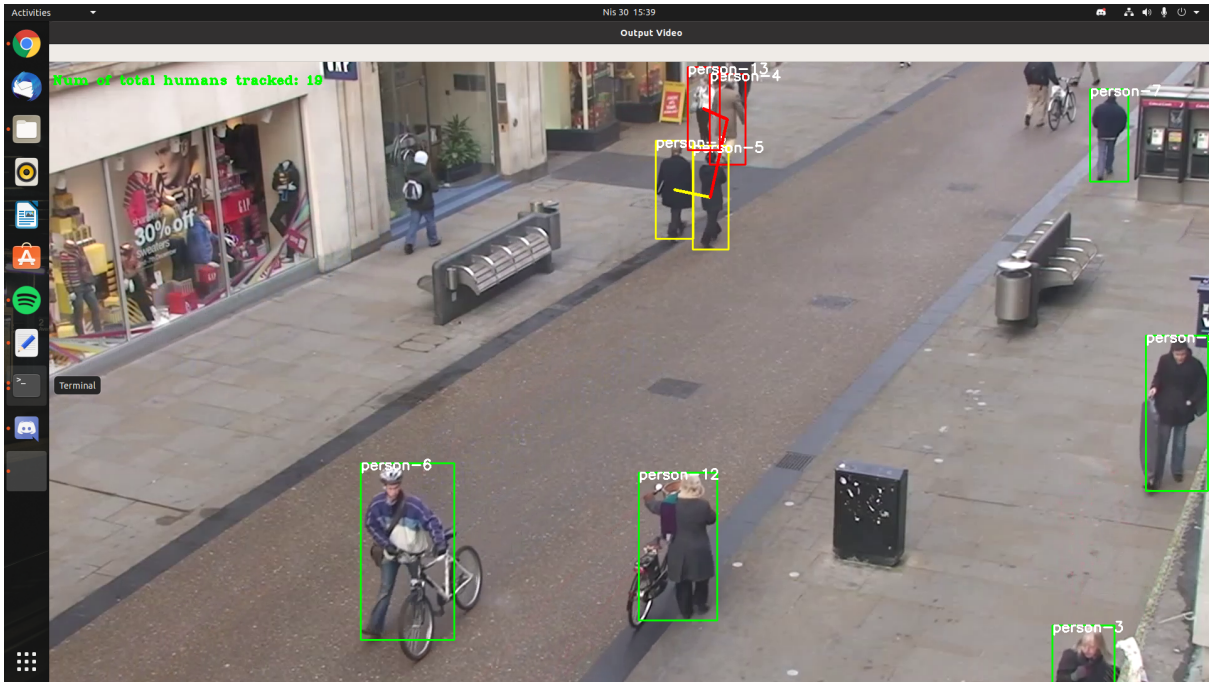


Figure 9: Social Distance Detection on Video Footage

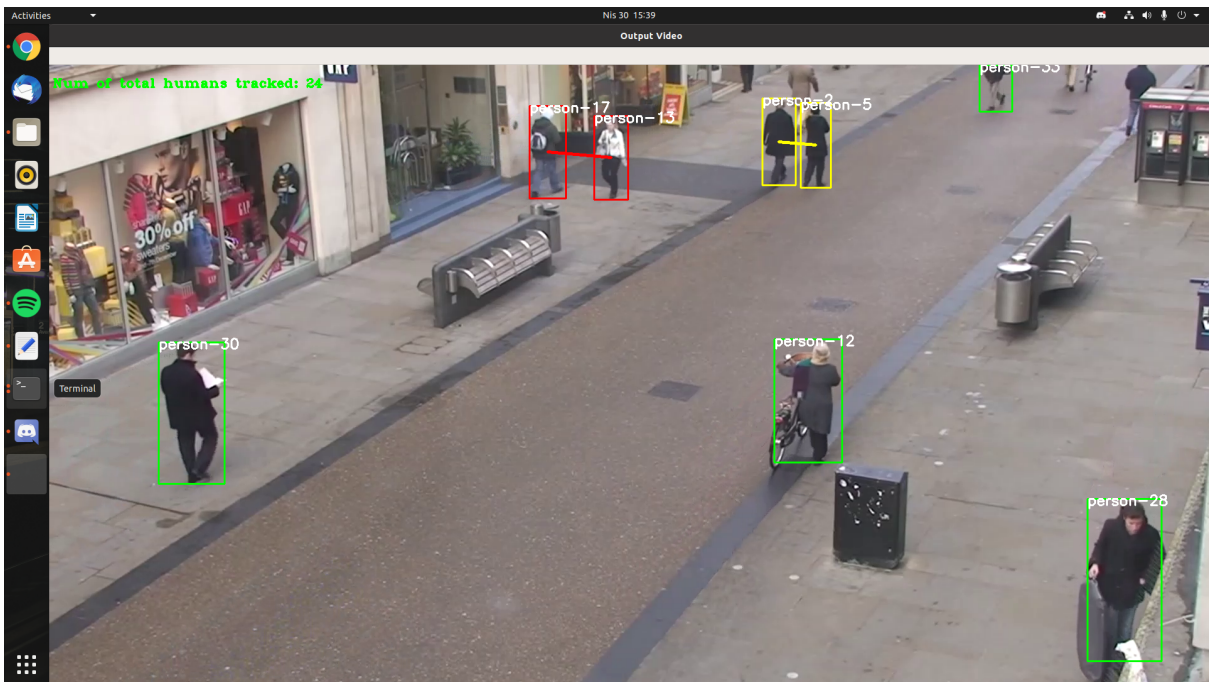


Figure 10: Social Distance Detection on Video Footage

Live Social Distance Detection

The image below is an example of social distancing with a real-time camera feed. Same box color rules also apply for the real-time detections. In the picture there is only one person which is detected as safe.

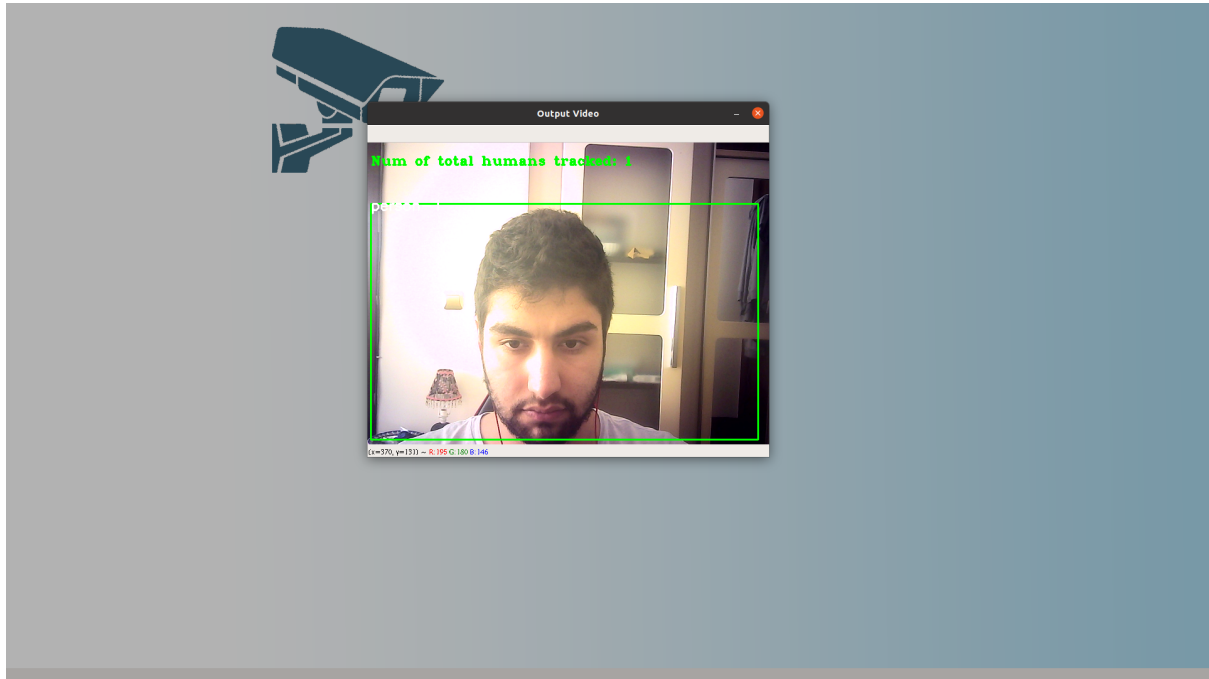


Figure 11: Live Social Distance Detection

Mask Detection Menu Page

Mask detection page also provides pre-recorded and live options. As it is in the social distancing, pre-recorded video footage can be selected from the local camera and live camera option works with real-time camera feed.



Figure 12: Mask Detection Menu Page

Mask Detection Using Webcam

The image below is an example of a live camera record which detects the face mask.

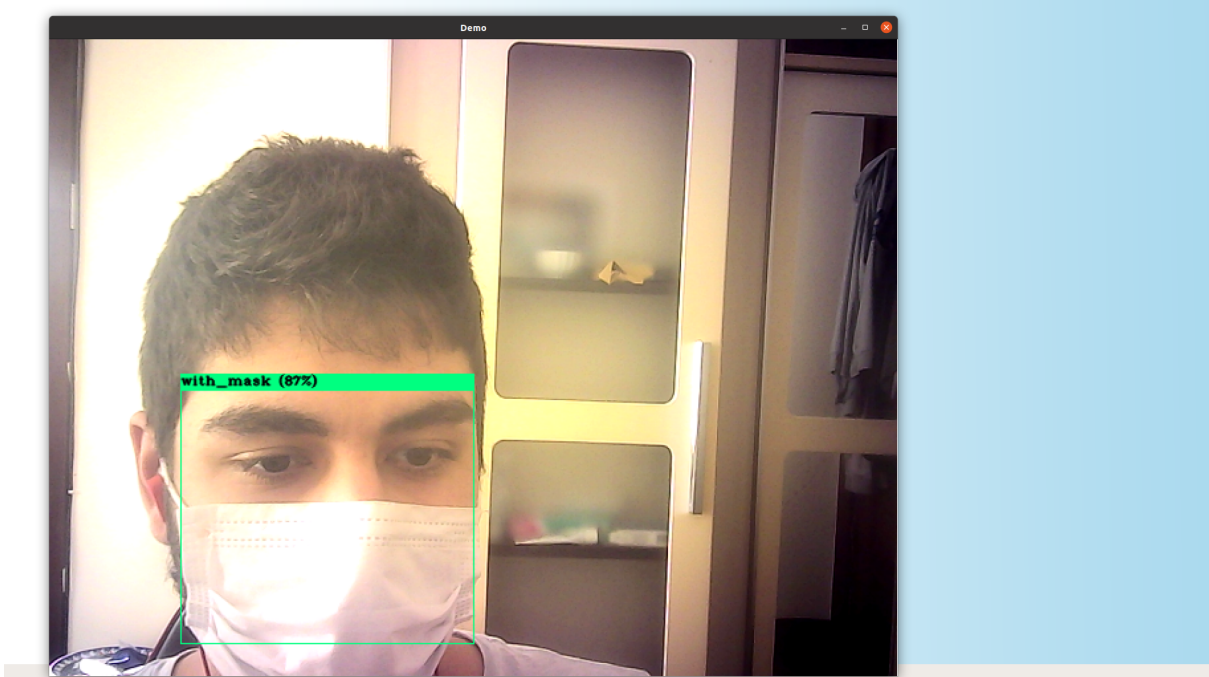


Figure 13: Mask Detection Using Webcam

Pre-recorded Mask Detection

Pre-recorded mask detection system works the same as the pre-recorded social distance detecting. Users choose a video from their local computer, and mask detection works after uploading the file. Third eye generates the record and shows the result on the screen.

Some sample pre-recorded images with the results are given below.

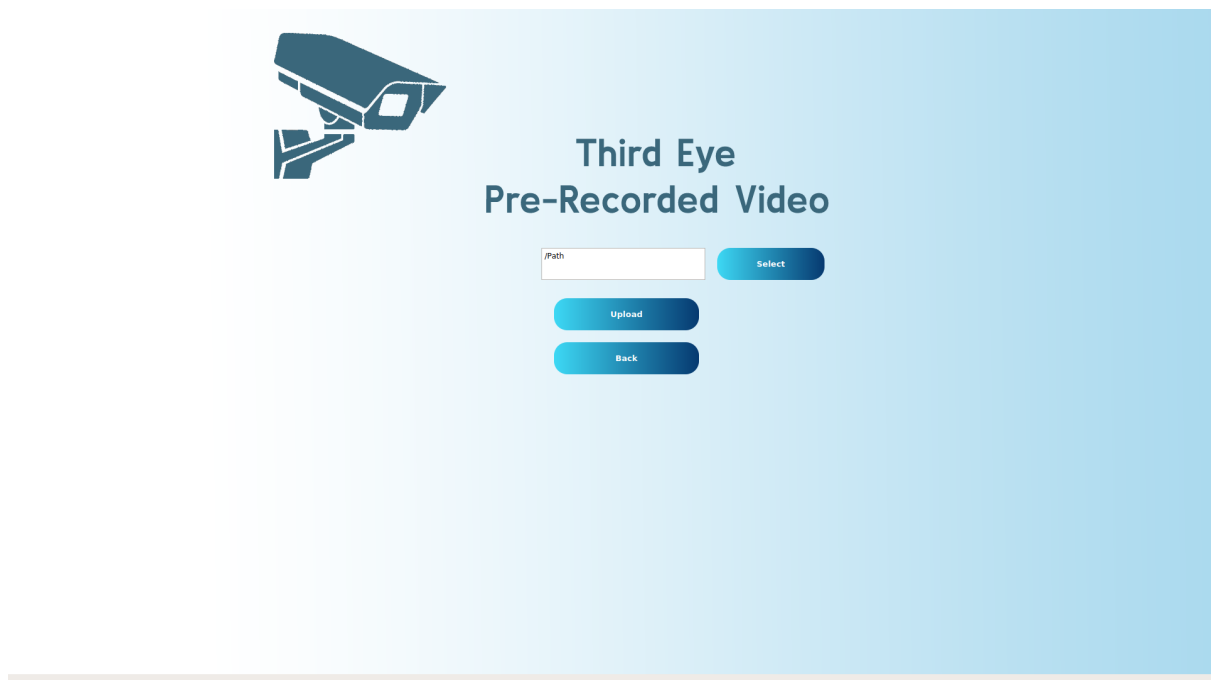


Figure 14: Pre-recorded Mask Detection



Third Eye

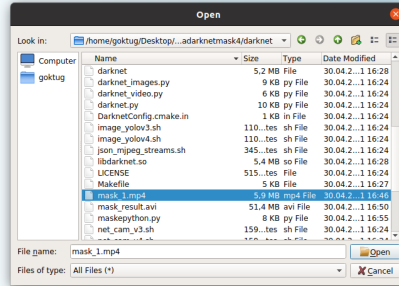


Figure 15: Selecting Video Footage for Mask Detection

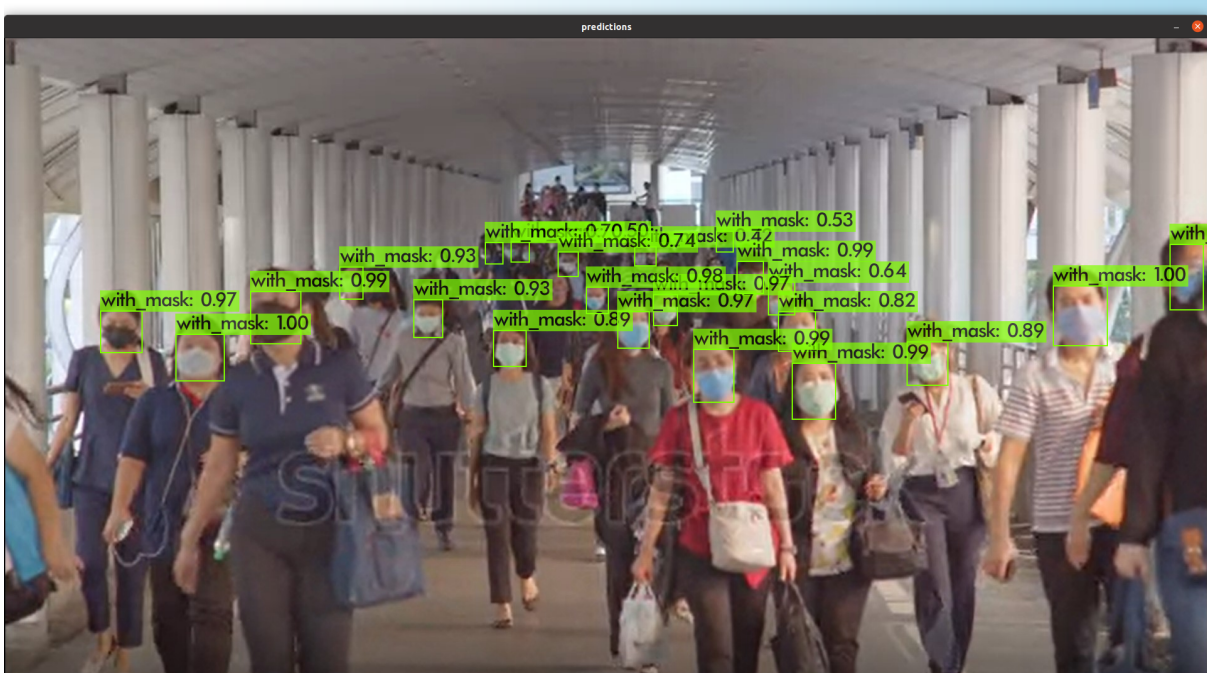


Figure 16: Mask Detection with Video Footage



Figure 17: Mask Detection with Video Footage

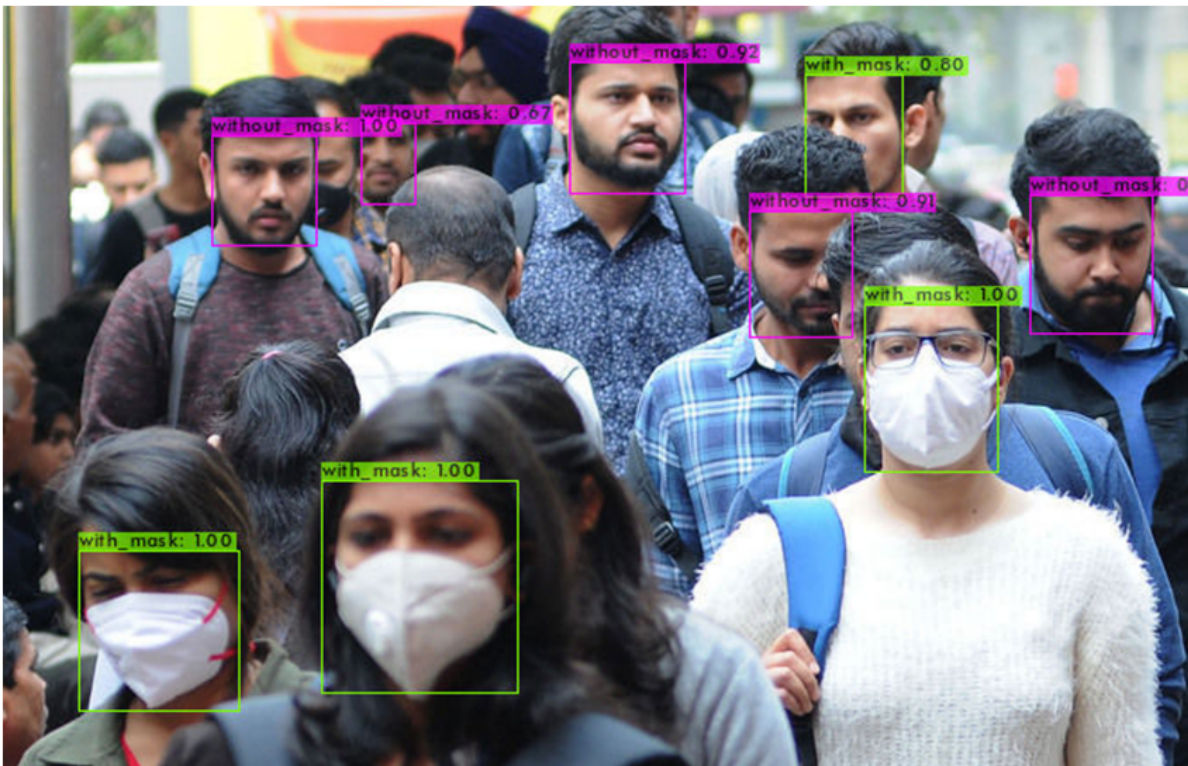


Figure 18: Mask Detection with Image



Figure 19: Mask Detection with Image

10. Glossary

GitHub: An online platform that provides hosting for software development version control using Git.

Jira: A software that helps manage agile and software development projects.

OpenCV: A library of programming functions mainly aimed at real-time computer vision developed by Intel[6].

Jetson Nano: A development kit for AI applications that reduces complexity and effort for developers. Jetson Nano is a small, powerful computer that lets you run multiple neural networks in parallel for applications like image classification, object detection, segmentation, and speech processing. All in an easy-to-use platform that runs in as little as 5 watts [7].

Computer Vision: Computer vision is a field of artificial intelligence that trains computers to interpret and understand the visual world.

Tkinter: It is a standard Python interface to the Tk GUI toolkit.

YOLOv4: It is a real-time object recognition system which can recognize multiple objects in a frame [8].

GPU: An abbreviation for Graphics Processing Unit. An electronic circuit designed to manipulate and alter memory to accelerate the creation of images in a frame buffer intended for output to a display device [15].

References

- [1] "Naming the coronavirus disease (COVID-19) and the virus that causes it," World Health Organization. [Online]. Available: [https://www.who.int/emergencies/diseases/novel-coronavirus-2019/technical-guidance/naming-the-coronavirus-disease-\(covid-2019\)-and-the-virus-that-causes-it](https://www.who.int/emergencies/diseases/novel-coronavirus-2019/technical-guidance/naming-the-coronavirus-disease-(covid-2019)-and-the-virus-that-causes-it). [Accessed: 11-Oct-2020].
- [2] I. Chakraborty and P. Maity, "COVID-19 outbreak: Migration, effects on society, global environment and prevention," *Science of The Total Environment*, vol. 728, p. 138882, 2020.
- [3] M. Greenstone and V. Nigam, "Does Social Distancing Matter?," *SSRN Electronic Journal*, 2020.
- [4] K. Kelland and M. Revell, "Pandemic behavior: Why some people don't play by the rules," *The Jakarta Post*, London, 13-Aug-2020.
- [5] "Code of Ethics," *Code of Ethics | National Society of Professional Engineers*. [Online]. Available: <https://www.nspe.org/resources/ethics/code-ethics>. [Accessed: 10-Oct-2020].
- [6] "OpenCV", *About OpenCV* [Online]. Available: <https://opencv.org/about/> [Accessed: 21-Nov-2020].
- [7] "NVIDIA Jetson Nano", *Jetson Nano* [Online]. Available: <https://developer.nvidia.com/embedded/jetson-nano-developer-kit#:~:text=NVIDIA%C2%AE%20Jetson%20Nano%E2%84%A2,as%20little%20as%205%20watts>. [Accessed: 21-Nov-2020].
- [8] J. Radmon and A. Farhadi, "YOLOv3: An Incremental Improvement." [Online]. Available: <https://pjreddie.com/media/files/papers/YOLOv3.pdf>. [Accessed: 30-Apr-2021].
- [9] "Darknet: Open Source Neural Networks in C". <https://pjreddie.com/darknet/>. [Accessed 28-Apr-2021].
- [10] C. Boldt. "Does social distancing help prevent COVID-19". July 30, 2020. <https://www.mdanderson.org/cancerwise/does-social-distancing-help-prevent-coronavirus-covid-19-spread.h00-159383523.html>. [Accessed 23-Apr-2021].
- [11] "General data privacy regulation." <https://eugdpr.org/>. [Accessed: 17-Nov-2020].

[12] “ACM code of ethics and professional conduct,” *Communications of the ACM*, vol. 35, no. 5, pp. 94–99, 1992.

[13] A. Harvey, *Exposing.ai*. [Online]. Available: <https://exposing.ai/>. [Accessed: 30-Apr-2021].

[14] B. Benfold and R. Benfold, “Stable multi-target tracking in real-time surveillance video,” *IEEE Xplore*. [Online]. Available: <https://ieeexplore.ieee.org/document/5995667/>. [Accessed: 30-Apr-2021].

[15] D. Atkin, “Computer Shopper: The Right GPU for You”. [Online]. Available: https://web.archive.org/web/20070506033224/http://computershopper.com/feature/200704_the_right_gpu_for_you. [Accessed: 30-Apr-2021].