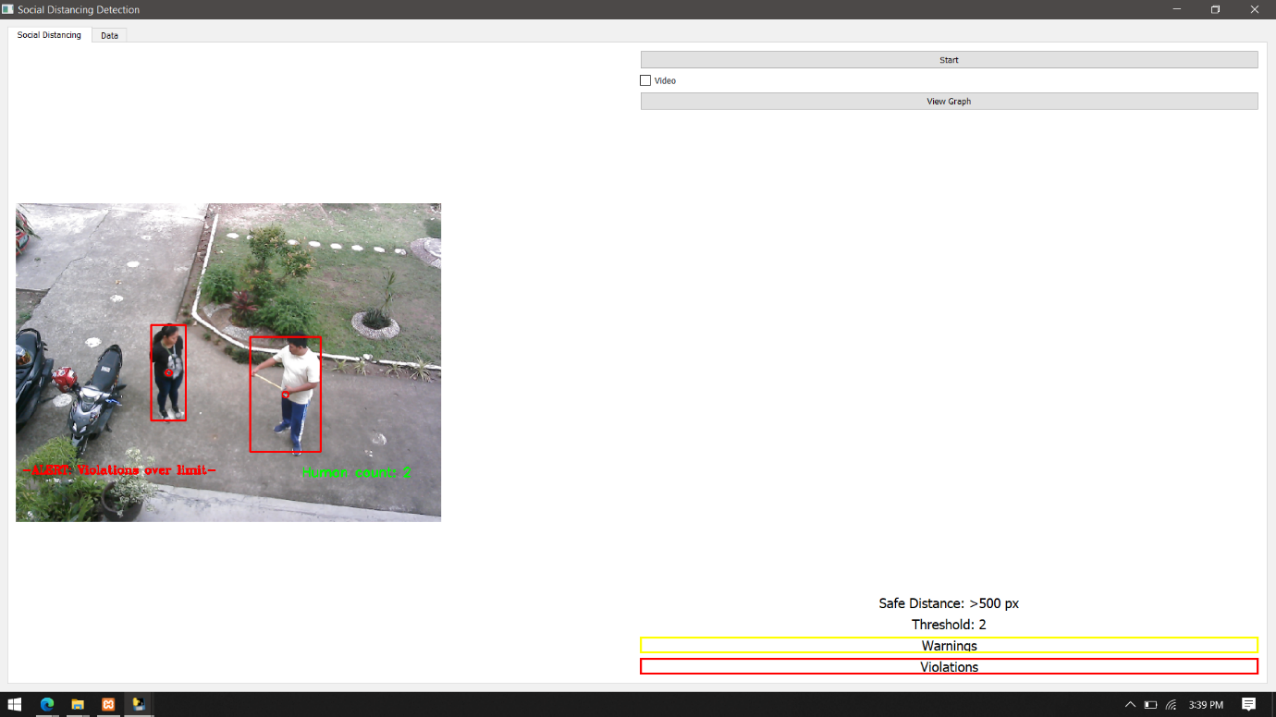
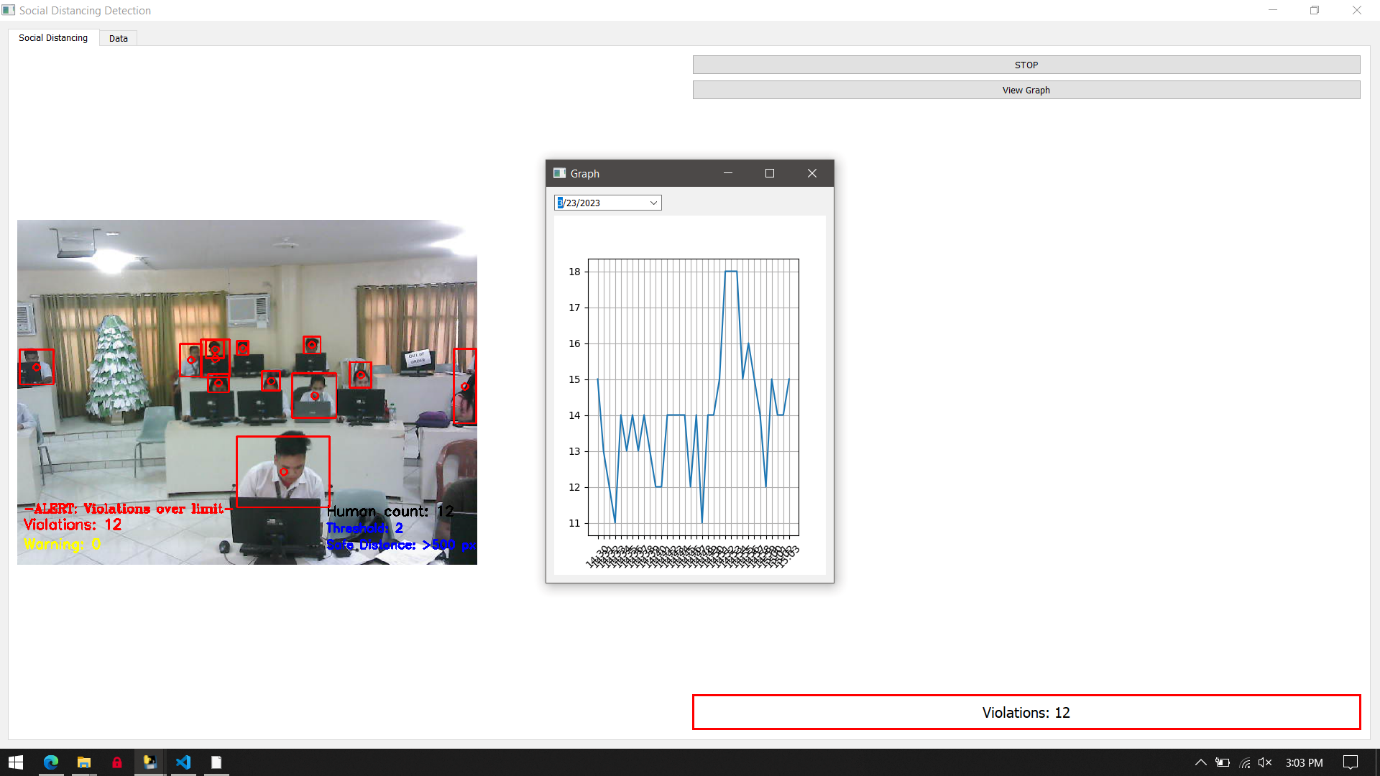
CHAPTER IV

Result And Discussion

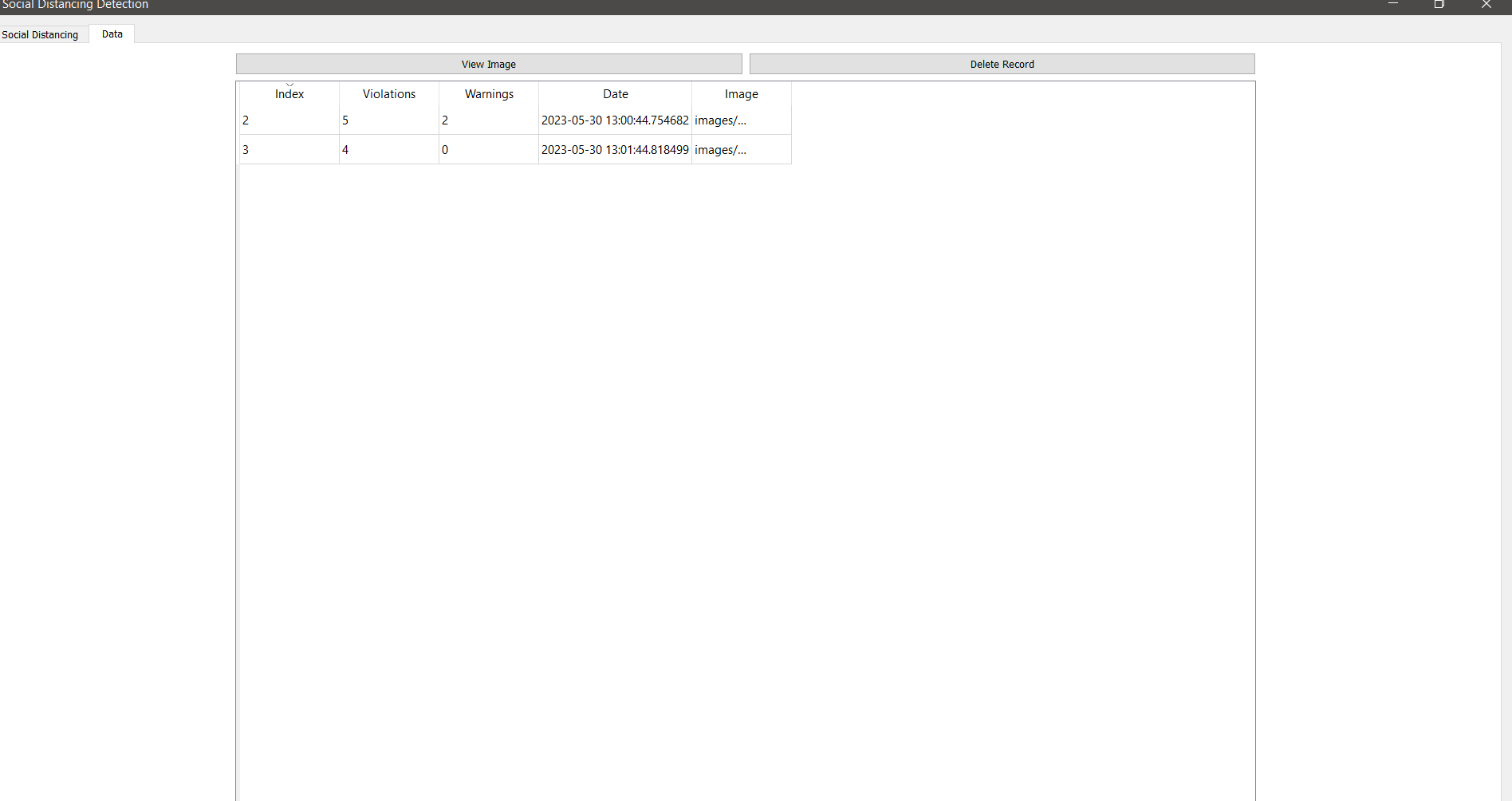
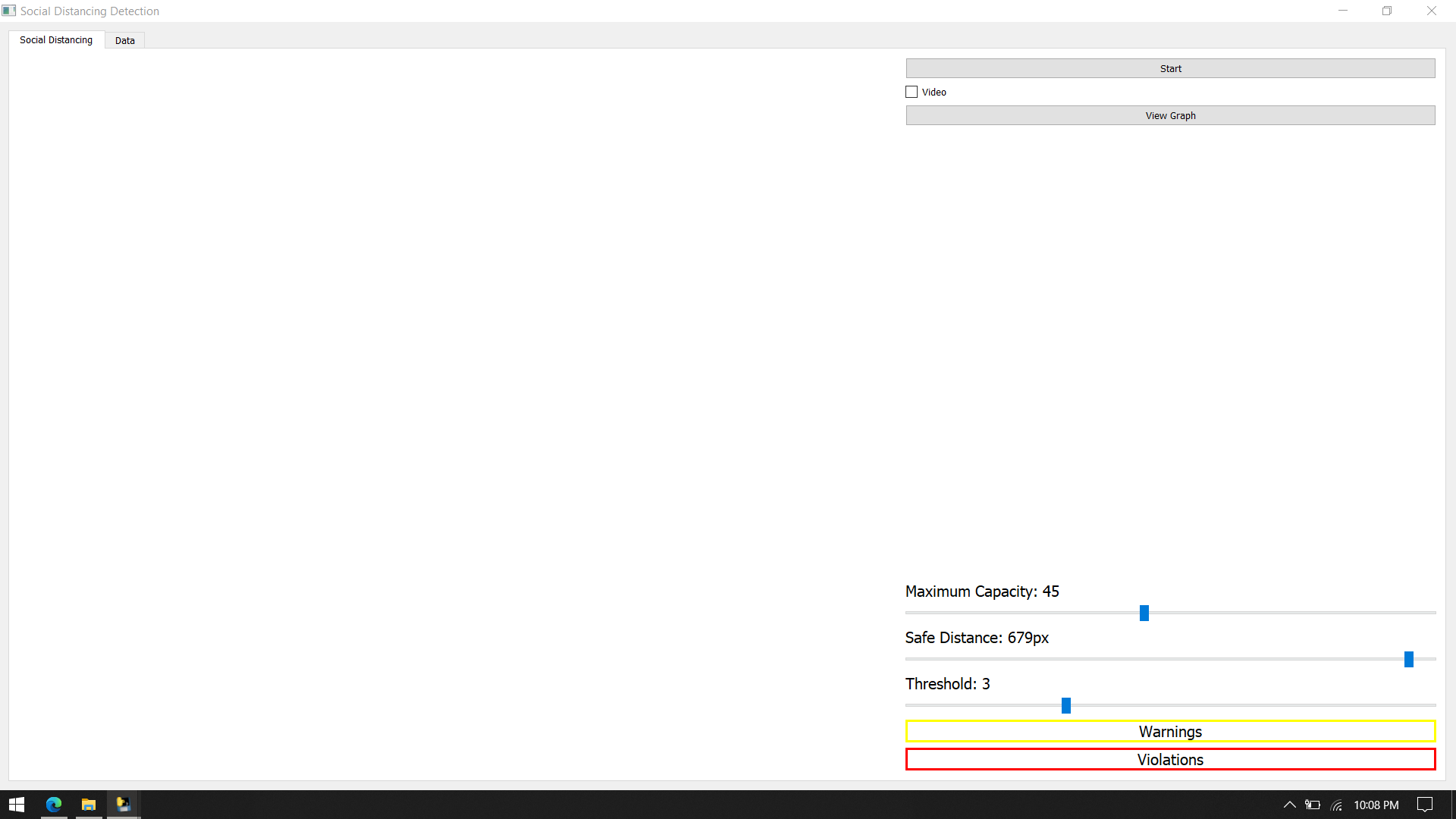
**To Design and develop a crowd counting system applying Object detection and tracking algorithm.**

Main Page

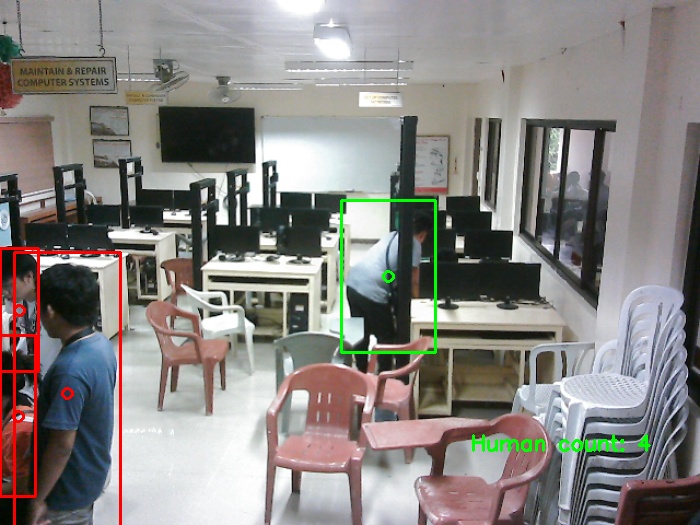
The Proponents Develop a System Using Python language with the functionalities of Video feed, Image Capturing, Object detection, Object Tracking, and Distance Detection this page have Single Camera Inputs for Capturing Data. Video capture is useful for object detection and tracking also for Distance Detection as it enables the continuous recording of video footage, which can be analyzed frame by frame to identify the distance and track objects of interest.

The page has a different Functionalities First, you can record the video when your system starts by checking the box next to the video. When it finishes, the recorded video will be saved to the system folder.

Second, when the system is running, the highest violation per minute will be recorded in the graph, and it will keep doing until the system is stopped.

Third, the human capacity, safe distance, and threshold can all be manually configured so that they appear on your recorded video when you play it. The system will automatically send a notification to the system registered user when the human capacity reaches the inputted data.

Fourth, the database contains a record of violations that are gathered when you use the system. You can examine an image of the Highest violation that took place when you were using the system. Additionally, you have the choice of erasing the recorded data.



Object detection is a computer vision technique that analyzes video frames to identify objects. Models are trained on annotated images with labeled bounding boxes indicating object positions. During detection, the model scans frames, identifying regions potentially containing objects based on learned patterns. It predicts the object's class and provides bounding box coordinates for its location within the frame.

Object tracking is a technique used to monitor and track objects across video frames. After initial detection in the first frame, object tracking assigns unique IDs to objects and associates them with subsequent frames using appearance, location, and motion cues. By analyzing these characteristics, the algorithm determines which objects in the current frame match those tracked from the previous frame. Tracked objects are updated with new bounding box coordinates in each frame, enabling continuous monitoring of their movement throughout the video.

Overall, the process involves detecting objects in each frame using deep learning models and then tracking their movements across subsequent frames. By combining both techniques, object detection and tracking enable the identification and monitoring of specific objects in live video feedback.

|  |  |  |
| --- | --- | --- |
| A | B | C |
| 1. *Green Bounding box*   *for safe distance* | 1. *Yellow bounding Box*   *for warning distance* | 1. *Red bounding box*   *for violation* |

Green Bounding Box: When you see a person on the screen, the object detection system will draw a green bounding box around them. This green color indicates that the person is following the recommended social distancing guidelines. They are maintaining a safe distance from others, which is great for preventing the spread of diseases.

Yellow Bounding Box: Now, let's say another person appears on the screen, but this time a yellow bounding box is drawn around them. The yellow color serves as a warning sign. It means that the person is standing too close to someone else and is not maintaining the recommended social distance. This situation indicates a potential violation of social distancing guidelines.

Red Bounding Box: Finally, if you see a person with a red bounding box around them, it's a critical situation. The red color indicates a high-risk violation of social distancing guidelines. This person is standing very close to others, posing a significant risk for the spread of diseases. Immediate action might be required to intervene and enforce proper social distancing measures.

By using these color-coded bounding boxes, the object detection system makes it easier for you to identify individuals who are following or violating social distancing guidelines at a glance. This helps in real-time monitoring and allows for quick interventions to maintain a safe environment.

2. Test the system functionalities using You Only Look Once (YOLO) Data set for the following process;

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Crowd Detection | | | | | | |
|  | Environment | Total Images | True Positives | True Negatives | False Positives | False Negatives |
| Test 1 | Indoor | 100 | 80 | 10 | 5 | 5 |
| Test 2 | Indoor | 100 | 70 | 15 | 10 | 5 |
| Test 3 | Outdoor | 100 | 85 | 8 | 4 | 3 |
| Test 4 | Outdoor | 100 | 90 | 5 | 2 | 3 |

In this table, each test consists of 100 images. The "Detected Crowds" column represents the number of crowds correctly detected by the YOLOv3 model.

**Remarks**

**Test 1:** The system achieved a high accuracy with a good balance between true positives and true negatives, but there were some false positives and false negatives. Further optimization may be required.

**Test 2:** The system showed a slightly lower accuracy compared to Test 1, with more false positives. Enhancement in reducing false positives is necessary for better detection.

**Test 3:** The system performed well in outdoor environments, accurately detecting crowds with a low number of false positives and negatives. Good overall accuracy.

**Test 4:** The system achieved a high accuracy rate in outdoor scenarios, with minimal false positives and negatives. Excellent performance overall.

Overall, the system demonstrates satisfactory to excellent performance in crowd detection, with the highest accuracy achieved in outdoor scenarios. Test 1 and Test 3 both show high accuracy rates and good balance between true positives and true negatives. Test 2 exhibits a slightly lower accuracy due to an increased number of false positives. Test 4 stands out with the highest accuracy, indicating the system's robustness in outdoor environments.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Detecting the distance between individuals** | | | | | | |
|  | **Environment** | **Total Images** | **True Positives** | **True Negatives** | **False Positives** | **False Negatives** |
| **Test 1** | Indoor | 100 | 70 | 10 | 5 | 5 |
| **Test 2** | Indoor | 100 | 60 | 15 | 10 | 5 |
| **Test 3** | Outdoor | 100 | 80 | 8 | 4 | 3 |
| **Test 4** | Outdoor | 100 | 85 | 5 | 2 | 3 |

In this table, each test still consists of 100 images. The "Detected Individuals" column represents the number of individuals correctly detected by the YOLOv3 model

**Remarks**

**Test 1:** The system achieved moderate accuracy with a balance between true positives and true negatives. There were some false positives and false negatives, indicating room for improvement.

**Test 2:** The system showed a slightly lower accuracy compared to Test 1, with more false positives. Further optimization is necessary to reduce false positives and improve accuracy

**Test 3:** The system performed well in outdoor environments, accurately detecting distances with a low number of false positives and negatives. Good overall accuracy.

**Test 4:** The system achieved a high accuracy rate in outdoor scenarios, with minimal false positives and negatives. Excellent performance overall.

Overall, the system's performance in detecting the distance between individuals varied across the tests. In indoor scenarios (Test 1 and Test 2), the accuracy ranged from 75% to 80%, with a higher number of false positives. This indicates that there is room for improvement to reduce false positives and enhance the accuracy of distance detection in indoor environments.

In outdoor scenarios (Test 3 and Test 4), the system performed relatively better with accuracy values of 88% and 90%, respectively. The number of false positives was lower, leading to a more accurate detection of distances between individuals.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Triggers alert when individuals are too close to each other** | | | | | | |
|  | **Environment** | **Total Images** | **True Positives** | **True Negatives** | **False Positives** | **False Negatives** |
| **Test 1** | Indoor | 100 | 70 | 10 | 5 | 5 |
| **Test 2** | Indoor | 100 | 70 | 5 | 10 | 5 |
| **Test 3** | Outdoor | 100 | 77 | 8 | 4 | 3 |
| **Test 4** | Outdoor | 100 | 85 | 5 | 2 | 3 |

In this table, each test consists of 100 images. The "Detected Triggers" column represents the number of triggers correctly detected by the YOLOv3 model.

**Remarks**

**Test 1:** The system achieved moderate accuracy in triggering alerts when individuals are too close, with a balance between true positives and true negatives. Further optimization is needed to reduce false positives and false negatives

**Test 2:** The system showed a slightly lower accuracy compared to Test 1, with more false positives. Enhancements are required to improve accuracy and reduce false positives.

**Test 3:** The system performed well in outdoor environments, accurately triggering alerts when individuals are too close with a low number of false positives and negatives. Good overall accuracy.

**Test 4:** The system achieved a high accuracy rate in outdoor scenarios, with minimal false positives and negatives. Excellent performance overall.

Overall, the system showed varying levels of accuracy in detecting close proximity between individuals. Further optimization is needed to reduce false positives and false negatives, particularly in indoor environments. The system performed better in outdoor settings, achieving higher accuracy with fewer false positives and negatives.

3.. Evaluate the system accuracy using confusion matrix on the following functionalities ;

|  |  |  |  |
| --- | --- | --- | --- |
| System Functionality | Distance | Crowd | Accuracy |
| 1. Density and Scale | 87% | 89% | 88% |
| 1. Prediction Error | 90% | 95% | 92.5% |
| 1. Processing | 89% | 90% | 89.5% |

To Evaluate the system accuracy, we use a dataset with labeled images or videos. Each sample is marked with whether social distancing is maintained or violated.

1. We apply the method to the dataset and make predictions for each sample.
2. We compare these predictions with the labeled information to count the number of correct and incorrect predictions (true positives, true negatives, false positives, and false negatives).
3. We calculate accuracy by considering the correct and incorrect predictions.

In simpler terms, we're testing two methods to see how well they can tell if people are following social distancing rules. We use a set of pictures or videos where we already know the correct answer. We compare the methods' predictions with the correct answers to see how accurate they are. The accuracy score tells us how well the methods perform in detecting social distancing. Same Process in calculating the accuracy of Crowd

**3.Processing**

|  |  |
| --- | --- |
| **Social Distance** | |
| **Step** | **Accuracy (%)** |
| Object Detection | 91 |
| Distance Calculation | 90 |
| Thresholding | 87 |
| Overall Accuracy | 89 |

|  |  |
| --- | --- |
| **Crowd** | |
| **Step** | **Accuracy (%)** |
| Object Detection | 89 |
| Counting Algorithm | 91 |
| Overall Accuracy | 90 |

In the above tables, each step involved in social distancing detection and crowd counting processing is listed along with its corresponding accuracy percentage. These accuracy values can be obtained through evaluation on a test dataset or by comparing the results with ground truth annotations.

The overall accuracy represents the cumulative accuracy of all the processing steps, providing an assessment of the system's performance as a whole.