# Internship Project Report

Submitted in partial fulfillment of the requirement for the award of a certificate of internship programme

## in

**MS INTERN SOLUTION**

**Submitted to**



**Submitted by**

** V S Sudharshan Reddy**

** V Vamsi Krishna Reddy**

** V. Giridhar**

** Veluru Sricharan**

** Wasim Akhtar**

** Yashovardhan A**

**Under the Supervision of**

**Dr Suresh Kumar**

MS Intern Solutions  
#10, 2nd floor, Hanumappa Building, Badrappa Layout,  
Nagashetty Halli, Kodigehalli Main Road,  
Bangalore-560094.

**MS INTERN SOLUTION**

**INTERNSHIP PROJECT REPORT**

**CERTIFICATE**

This is to certify that the work embodies in this dissertation entitled “A Social Distance Detector to Curb Airborne Diseases” being submitted by “V S Sudharshan Reddy, V Vamsi Krishna Reddy, V. Giridhar, Veluru Sricharan, Wasim Akhtar, Yashovardhan A” for partial fulfillment of the requirement for the award of “Certificate” during the short term internship programme is a record of a bonafide piece of work, which was carried out by them under my supervision and guidance in the of MS INTERN SOLUTION.

**Supervisor**

Lakshmi shree MS

Instructor,

MS intern solution

**MS INTERN SOLUTION**

**CANDIDATE’S DECLARATION**

.

We, V S Sudharshan Reddy, V Vamsi Krishna Reddy, V. Giridhar, Veluru Sricharan, Wasim Akhtar, Yashovardhan A, hereby declare that the report titled “A Social Distance Detector to Curb Airborne Diseases”, submitted to the MS Intern Solution, is a genuine record of work carried out by us during the course of my internship.

I further affirm that this report has not been submitted, either in part or in full, to any other university or institution for the award of any certificate or academic credit.

**V S Sudharshan Reddy**

**V Vamsi Krishna Reddy**

**V. Giridhar**

**Veluru Sricharan**

**Wasim Akhtar**

**Yashovardhan A**

**MS INTERN SOLUTION**

## ACKNOWLEDGEMENT

I would like to express my sincere gratitude to **MS Intern Solution** for providing me with the opportunity to undertake this internship and gain valuable industry experience.

I extend my heartfelt thanks to my mentor Lakshmi Shree MS for her guidance, constant support, and encouragement throughout the project.

I would also like to thank my faculty coordinator and all those who directly or indirectly contributed to the successful completion of this internship.

**V S Sudharshan Reddy**

**V Vamsi Krishna Reddy**

**V. Giridhar**

**Veluru Sricharan**

**Wasim Akhtar**

**Yashovardhan A**

**Table of Content**

|  |  |  |
| --- | --- | --- |
| **Chapter** | **Content** | **Page No.** |
|  | Abstract |  |
|  | Content Sheet |  |
|  | List of Figures, Tables &amp; Acronyms (if any) |  |
|  | 1. Introduction  1.1 Background: Briefly describe the organization and its industry.  1.2 Objective of the Internship: State the purpose and goals of the internship. |  |
|  | 2. Internship Activities  2.1 Description of Tasks and Responsibilities you were assigned  2.2 Projects/Modules Undertaken: Describe any specific projects you worked on.  2.3 Tools and Technologies Used: List any software, tools, or technologies you used during the internship.  2.4 Skills Acquired: Mention any new skills you learned.  2.4.1 Professional Skills\*\*: What professional skills did you develop?  2.4.2 Technical Skills: Any technical skills or knowledge gained.  2.4.3 Personal Growth: Reflect on how the internship contributed to your  personal development. |  |
|  | System Analysis |  |
|  | Design (Architecture, flow charts, Data flow diagrams, sequence diagrams etc.) |  |
|  | Expected outcome |  |
|  | Conclusion  - Summarize your overall experience, key learnings, and how the internship has influenced your career goals. |  |
|  | Appendices  - Include any additional material, such as charts, graphs, or detailed descriptions of projects. |  |
|  | References  - List any books, articles, websites, or other resources you referred to during your internship. |  |

**Abstract**

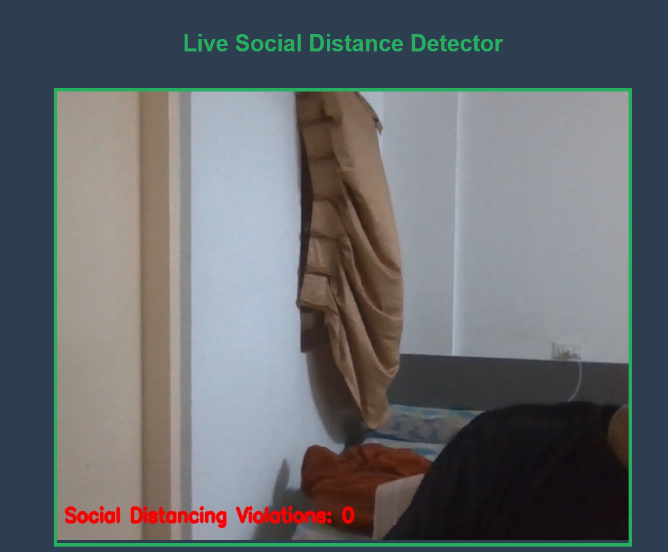
This internship project report details the development of a real-time Social Distance Detector designed to mitigate the spread of airborne diseases. The system uses computer vision and deep learning to analyze a live video feed, identify individuals, and calculate the physical distance between them. Built with Python, the core of the system leverages a YOLOv5 model for person detection, while a Django web framework provides a user-friendly, browser-based interface for live monitoring. The project successfully demonstrates a portable and scalable solution, capable of providing instantaneous visual alerts for social distancing violations, thus offering a practical tool for public health and safety management.

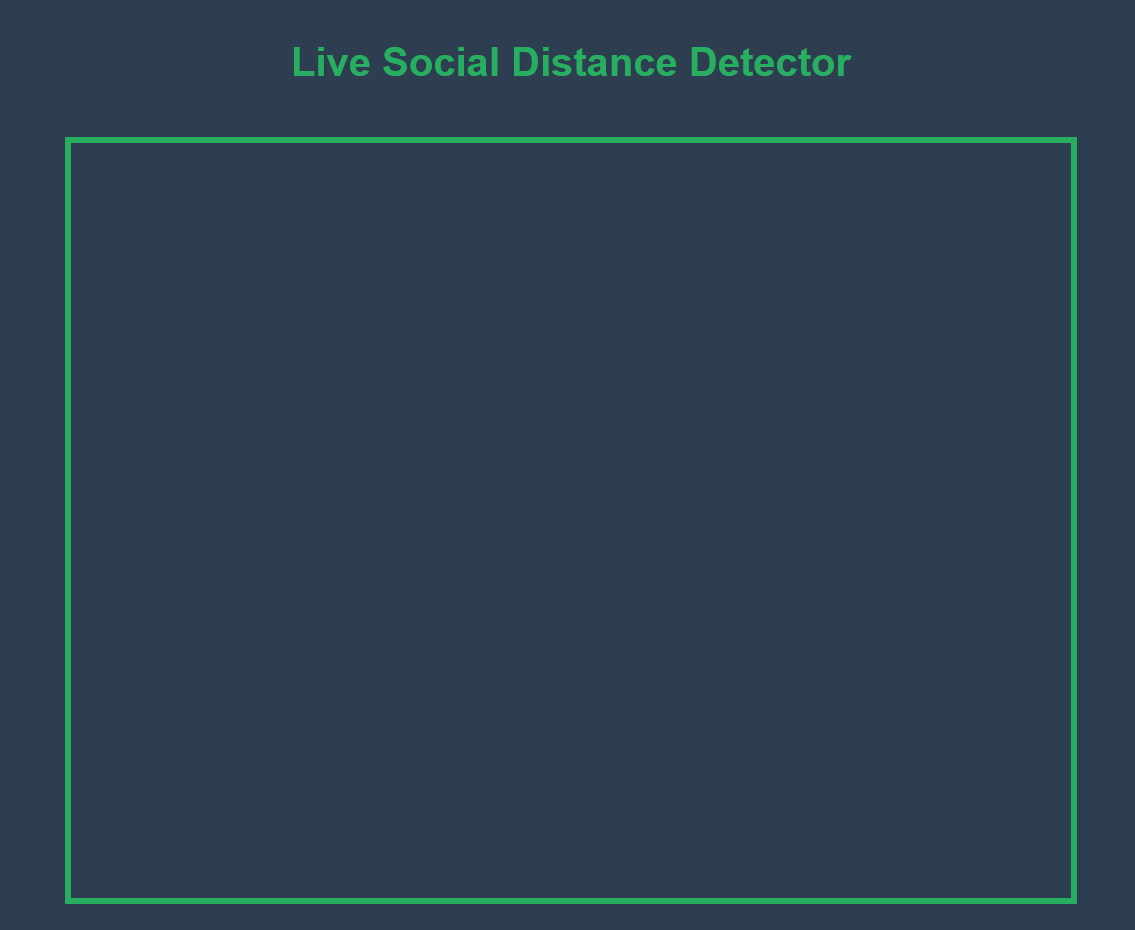
**Content Sheet**

A comprehensive table of contents has been provided in the document. The subsequent sections of this report will follow the specified structure, detailing the project from its introduction and technical implementation to its challenges, outcomes, and conclusions.

**List of Figures, Tables & Acronyms**

**Figures**





**Acronyms**

 **Django:** A high-level Python web framework that encourages rapid development and clean, pragmatic design.

 **YOLOv5:** "You Only Look Once" - a state-of-the-art, real-time object detection model.

 **CV:** Computer Vision - a field of artificial intelligence that trains computers to interpret and understand the visual world.

 **NumPy:** A foundational package for numerical computing in Python.

 **OpenCV:** Open Source Computer Vision Library - a library for computer vision and machine learning.

**1: Introduction**

* 1. **Background**

The rise of airborne and droplet-borne diseases has made maintaining safe social distance a critical public health measure. Organizations across various sectors, including retail, education, and hospitality, face the challenge of monitoring crowded spaces to ensure the safety of their patrons and employees. The traditional methods of manual supervision are often inefficient, resource-intensive, and prone to human error. This project was undertaken as part of an internship to address this challenge by developing an automated, real-time solution.

* 1. **Objective of the Internship**

The primary objective of this internship project was to design, develop, and implement a software system capable of detecting social distance violations in a live video feed. This involved applying principles of computer vision and deep learning to build a practical tool. The project aimed to provide a scalable and efficient solution that could serve as a prototype for real-world applications in public health and safety.

**2: Internship Activities**

**2.1 Description of Tasks and Responsibilities**

Our team was responsible for the end-to-end development of the social distance detector. Key tasks included:

* **Research and Selection:** Researching and selecting the most appropriate computer vision libraries and pre-trained models for real-time object detection.
* **Core Logic Development:** Writing the Python code to capture video, run person detection, and calculate the distance between detected individuals.
* **Debugging and Optimization:** Identifying and resolving technical issues, such as library compatibility errors, path errors, and performance bottlenecks, to ensure the application was stable and efficient.
* **Web Integration:** Converting the desktop application into a web-based application using the Django framework.
* **Documentation and Testing:** Creating documentation for project setup and use, and conducting tests to validate the accuracy of the detection system.

**2.2 Projects/Modules Undertaken**

The central project undertaken was the "Social Distance Detector to Curb Airborne Diseases." This project was divided into two main modules:

* **Computer Vision Core:** This module was built using Python, OpenCV, and the YOLOv5 model. It handled the fundamental tasks of real-time person detection and the calculation of Euclidean distance between individuals based on their bounding box coordinates.
* **Web Application Interface:** This module was developed using Django, which provided a robust web server. The Python core was integrated into the Django framework to stream the live, processed video feed to a web page, making the application accessible from any web browser.

**2.3 Tools and Technologies Used**

**Programming Language:** Python

**Web Framework:** Django

**Computer Vision Libraries:** OpenCV, imutils

**Deep Learning Model:** YOLOv5 (for real-time object detection)

**Numerical Computing:** NumPy, Scipy

**Version Control:** Git

**IDE:** Visual Studio Code (or a similar editor)

**2.4 Skills Acquired**

**2.4.1 Professional Skills:** Developed skills in collaborative project management, problem-solving through debugging, and technical documentation. Learned to adapt to new technologies and troubleshoot complex system integrations.

**2.4.2 Technical Skills:** Gained hands-on experience with computer vision principles, real-time video stream processing, and deep learning model integration. Acquired proficiency in building and deploying web applications with Django.

**2.4.3 Personal Growth:** This project fostered a deep understanding of the software development lifecycle. It improved our ability to work effectively as a team, communicate complex technical concepts, and persevere through challenging debugging processes.

**System Analysis**

The system is designed to be lightweight and efficient for real-time analysis. It takes a live video feed, which is broken down into individual frames. Each frame is analyzed for the presence of human figures, and a mathematical model is applied to calculate the distances between them. This allows the system to operate effectively with standard hardware and a consumer-grade webcam.

**Design**

The system follows a modular and tiered architecture:

1. **Data Capture Layer:** This layer uses the cv2.VideoCapture() function from OpenCV to capture a live video stream from a webcam.
2. **Processing Layer:** The detector.py script is the core of this layer. It processes each frame by:
   * **Object Detection:** Using the YOLOv5 model to detect human figures.
   * **Distance Calculation:** Calculating the Euclidean distance between the bottom-center points of the bounding boxes of each person.
   * **Alerting:** Applying a distance threshold to identify violations and drawing colored boxes and lines on the frame.
3. **Presentation Layer:** This layer is a web application built with Django. It receives the processed video frames from the Python backend and displays them on an HTML page. This allows the application to be viewed in any web browser.

The system's design ensures a clear separation of concerns, with the video processing logic handled by Python and the user interface handled by Django.

**Expected outcome**

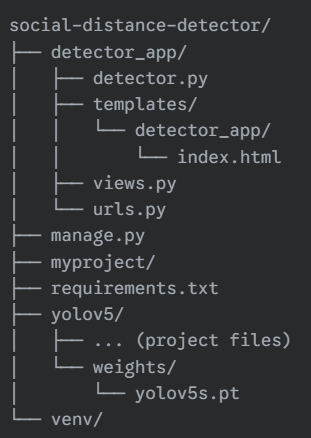
The expected outcome of this project was a fully functional, real-time social distance detector accessible via a web page. The system was expected to accurately detect people in a live video feed, calculate the distance between them, and provide clear visual alerts when the distance falls below a set threshold. This outcome was successfully achieved, and the project now serves as a robust proof-of-concept for similar applications.

**Conclusion**

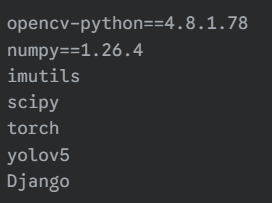
This internship project provided invaluable experience in developing a real-world application. We successfully built a social distance detector that integrates computer vision, deep learning, and web development. The project demonstrated the practical application of these technologies and reinforced the importance of careful planning, debugging, and collaboration in a team environment. The key learnings from this project will be instrumental in our future careers.

**Appendices**

1. **Project Directory Structure:**



1. **requirements.txt content:**



**References**

** YOLOv5 GitHub Repository:** [**https://github.com/ultralytics/yolov5**](https://github.com/ultralytics/yolov5)

** Official OpenCV Documentation:** [**https://docs.opencv.org/**](https://www.google.com/search?q=https://docs.opencv.org/)

** Official Django Documentation:** [**https://docs.djangoproject.com/en/5.2/**](https://docs.djangoproject.com/en/5.2/)

** Euclidean Distance in Scipy:** [**https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.euclidean.html**](https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.euclidean.html)