

***An internship report submitted by***

**STUDENT NAME - Reg. No. URK20CS3026**

***in partial fulfillment for the award of the degree of***

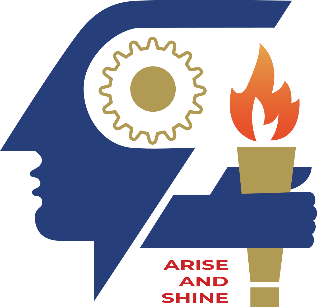
**BACHELOR OF TECHNOLOGY**

***in***

**COMPUTER SCIENCE AND ENGINEERING**

***under the supervision of***

**Dr. R. Chitra**



**DIVISION** **OF COMPUTER SCIENCE AND ENGINEERING**

**KARUNYA INSTITUTE OF TECHNOLOGY AND SCIENCES**

(Declared as Deemed to be University under Sec-3 of the UGC Act, 1956)

**Karunya Nagar, Coimbatore - 641 114. INDIA**



**DIVISION OF COMPUTER SCIENCE AND ENGINEERING**

**BONAFIDE CERTIFICATE**

This is to certify that the report entitled, “Social Distancing using Computer Vision and Deep Learning” is a bonafide record of Internship work done at Intel Unnati Industrial Training during the academic year 2022-2023 by

**STUDENT NAME: Shawn Abraham**

**(Reg. No: URK20CS3026)**

in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering of Karunya Institute of Technology and Sciences.

**Guide Signature**

Dr. R Chitra

Associate Professor

**ACKNOWLEDGEMENT**

First and foremost, I praise and thank ALMIGTHY GOD whose blessings have bestowed in me the will power and confidence to carry out my internship.

I am grateful to our beloved founders **Late. Dr. D.G.S. Dhinakaran, C.A.I.I.B, Ph. D** and **Dr. Paul Dhinakaran, M.B.A, Ph. D**, for their love and always remembering us in their prayers.

We entend Our tanks to **Dr. Prince Arulraj, M.E., Ph.D., Ph.D.,** our honorable vice chancellor, **Dr. E. J. James, Ph.D.,** and **Dr. Ridling Margaret Waller, Ph.D.,** our honorable Pro-Vice Chancellor(s) and **Dr. R. Elijah Blessing, Ph.D.,** our respected Registrar for giving me this opportunity to do the internship.

I would like to thank **Dr. Ciza Thomas, M.E., Ph.D.,** Dean, School of Engineering and Technology for his direction and invaluable support to complete the same.

I would like to place my heart-felt thanks and gratitude to **Dr. J. Immanuel John Raja, M.E., Ph.D.,** Head of the Division, Computer Science and Engineering for his encouragement and guidance.

I feel it a pleasure to be indebted to, Dr. R. Chitra, Associate Professor, Division of CSE & Mr. Mohan, Intel Guide for their invaluable support, advice and encouragement.

I also thank all the staff members of the School of CSE for extending their helping hands to make this in Internship a successful one.

I would also like to thank all my friends and my parents who have prayed and helped me during the Internship.

**Technical Report: Social Distancing using Computer Vision and Deep Learning**

**Title:** Social Distancing using Computer Vision and Deep Learning

**Team name/Authors:** One\_Hot\_Encoder / Shawn Abraham

**Date of submission:** 15 July 2023

**CONTENTS Page. No.**

[Abstract](#_TOC_250014)

1. Introduction 1
2. Implementation

2.1 Model Configuration and Training 3

2.2 Deployment with OpenVINO 4

1. [Social Distancing Monitoring System 4](#_TOC_250004)
2. Experimental Results 4
3. Motivation behind the problem 5
4. Conclusion 5
5. Result 6

**Abstract:**

Social distancing is an essential measure to prevent the spread of infectious diseases, such as COVID-19. The goal is to detect and track people in a video stream or image and determine whether they are maintaining a safe distance from each other. In this project, our system leverages the YOLOv8 object detection model and the OpenVINO toolkit for real-time monitoring and analysis of individuals' spatial proximity. By calculating the distances between the centroids of detected people, our system can identify and visualize potential violations of social distancing guidelines. The proposed approach aims to provide a real-time monitoring system to help ensure public safety and adherence to social distancing protocols. Two implementations are described in this report: one using regular YOLOv8 and another using OpenVINO runtime with a YOLOv8 model converted into Open VINO ONNX Format. The results and performance of each implementation are evaluated and compared.

**Introduction:**

The COVID-19 pandemic has highlighted the importance of social distancing in preventing the spread of infectious diseases. Traditional methods of monitoring social distancing compliance may be limited in their effectiveness and efficiency. Therefore, we propose a computer vision and deep learning-based solution to address this challenge. The objective of this project is to develop a system that can detect people in real-time to provide accurate and real-time monitoring of individuals' spatial proximity. Thus, we can identify and visualize potential violations of social distancing guidelines.

**Motivation:**

The motivation behind this work is to develop an automated system that can effectively monitor social distancing compliance in public spaces. Traditional manual methods of monitoring social distancing can be time-consuming, labor-intensive, and prone to errors. By leveraging computer vision and deep learning techniques, we aim to provide a real-time solution that can accurately detect people and identify instances of non-compliance with social distancing guidelines. This technology can assist in ensuring public safety, reducing the risk of disease transmission, and facilitating better enforcement of social distancing measures.

**Prior Work:**

Several studies have utilized computer vision and deep learning techniques for social distancing monitoring. The YOLO (You Only Look Once) object detection algorithm has been widely adopted due to its real-time performance and accuracy. Previous work has focused on using YOLO-based models to detect people and calculate the distances between them. Various distance metrics, such as Euclidean distance, have been employed to determine violations of social distancing guidelines. Additionally, post-processing techniques such as non-maximum suppression have been utilized to filter out redundant detections and improve the accuracy of the system.

**Our Approach:**

In our approach, we utilize the YOLO v8 object detection algorithm as a reference to benchmark the OpenVINO implementation to detect social distancing violations in real-time video streams. The YOLO v8n model is first loaded using PyTorch Built-in Functions and OpenCV python runtime and the video stream is processed frame by frame. The same YOLOv8n model is converted into a OpenVINO Model and implemented using OpenCV python for image processing and Built-in functions from OpenVINO runtime for loading the model and inference. The detected bounding boxes of people are extracted, and centroids are calculated for each person. The Euclidean distance between centroids is then computed to identify potential violations of social distancing guidelines. Violations are visualized by drawing lines between centroids and displaying the total number of social distancing violations on the output frame.

**Experimental Results:**

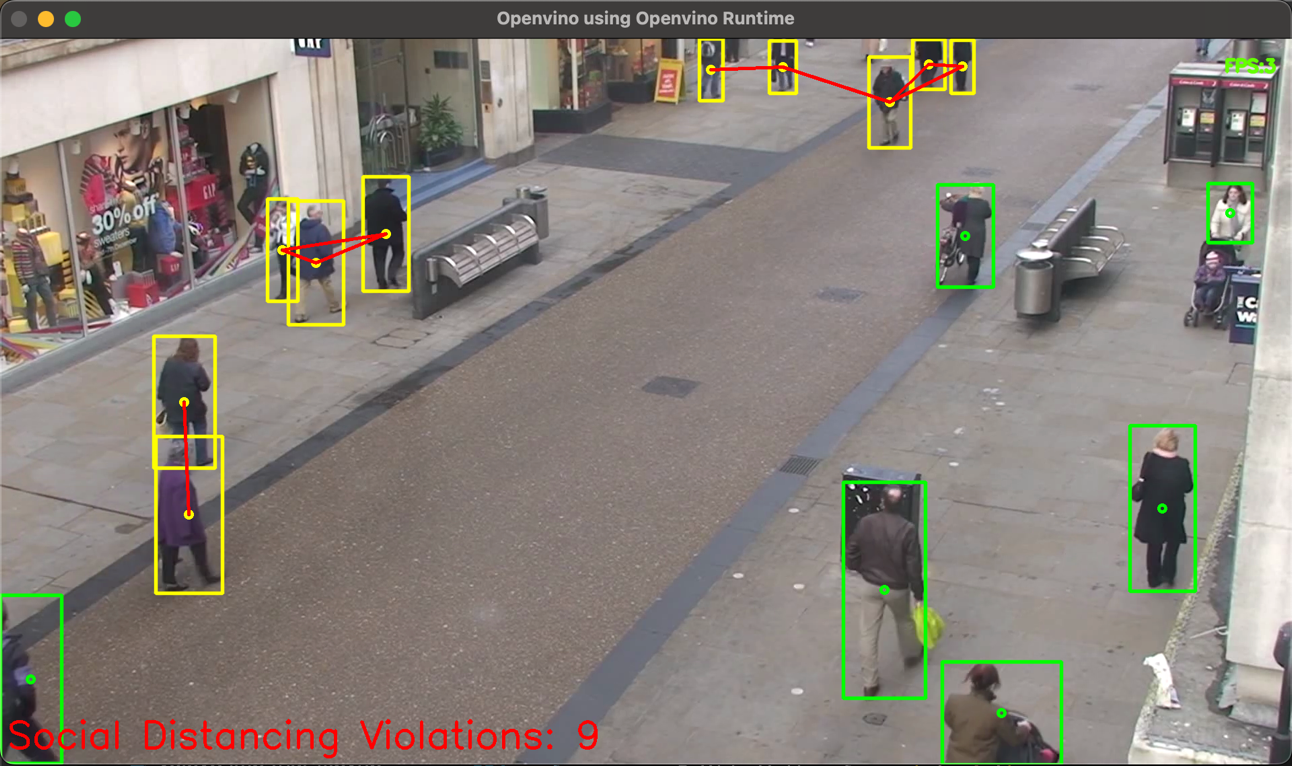
I have implemented two versions of the solution: one using regular YOLO v8 and another using a converted model optimized for OpenVINO Runtime.

I had developed and tested both models on my M1Pro MacBook, and recorded the outputs. The regular YOLO v8 implementation achieved a frame rate of 15-18 fps, while the converted model using OpenVINO runtime achieved a frame rate of 3-4 fps. While both implementations successfully detected people, and identified instances of social distancing violations in real-time video streams, it was noted that the OpenVINO implementation achieved a slightly lower accuracy than the regular Yolov8 implementation. The results were visualized by drawing bounding boxes around individuals, plotting centroids, and highlighting violations with colored lines.

The output videos with visualizations were saved for further analysis.

**Output Screenshots (My Laptop):**



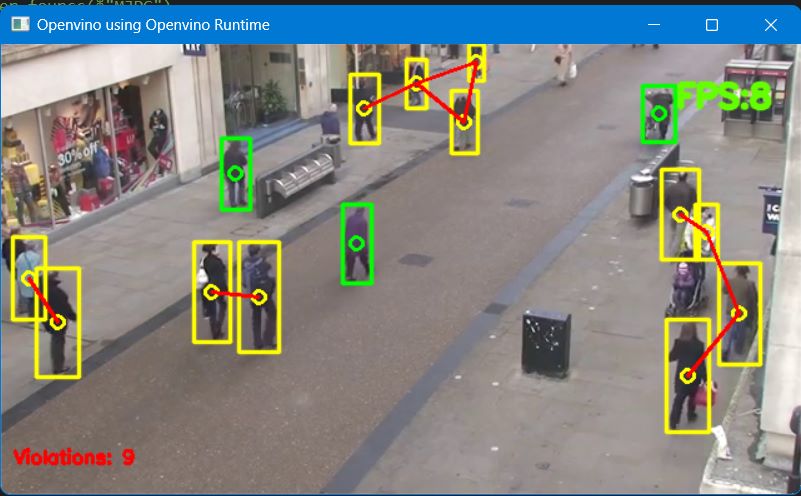


**Note:** Lately, I had asked my colleague to run the programs on an Intel Laptop and the results obtained were in stark contrast with my outputs. It showed comparatively better results with OpenVINO than the Yolov8 Model, similar to the claimed performance increase. The screenshots for the same have been attached in the next page. I am unsure of the exact cause of this discrepancy. I had queried this to my mentor during an industry mentor session, and he responded that OpenVINO was compatible with all platforms.

I hope that there will be an update on this matter and OpenVINO will truly extended support to all architectures and platforms so as to truly revolutionize the world of open-source ML-Ops deployment.

**Output Screenshots (Intel Laptop):**





**Link to Solution:** **Code, Models and Results** – GitHub link: <https://github.com/URK20CS3026SHAWN/intelunnati_One_Hot_Encoder>

**References:**

1. <https://docs.ultralytics.com/hub/>
2. <https://docs.openvino.ai/2022.3/omz_demos_social_distance_demo_cpp.html>
3. ﻿﻿﻿<https://www.intel.com/content/www/us/en/developer/tools/openvino-toolkit/download.html?ENVIRONMENT=RUNTIME&OP_SYSTEM=MACOS&VERSION=v_2022_3_1&DISTRIBUTION=PIP>
4. ﻿﻿﻿[https://github.com/openvinotoolkit/openvino\_notebooks/blob/main/notebooks/ 230-yolov8-optimization/230-yolov8-optimization.ipynb](https://github.com/openvinotoolkit/openvino_notebooks/blob/main/notebooks/%20230-yolov8-optimization/230-yolov8-optimization.ipynb)
5. ﻿﻿ <https://pyimagesearch.com/2020/06/01/opencv-social-distancing-detector/#:~:text=We%20can%20use%20OpenCV%2C%20computer,building%20an%20OpenCV%20people%20counter)>
6. <https://colab.research.google.com/github/roboflow-ai/notebooks/blob/main/notebooks/train-yolov8-object-detection-on-custom-dataset.ipynb#scrollTo=vpVNBR-Iuzwg>
7. [https://medium.com/@birla.deepak26/social-distancing-ai-using-python-deep-learning-c26b20c9aa4c](7.%09https:/medium.com/@birla.deepak26/social-distancing-ai-using-python-deep-learning-c26b20c9aa4c)