Karunya INSTITUTE OF TECHNOLOGY AND SCIENCES
(Declared as Deemed to be University under Sec.3 of the UGC Act, 1956)
MoE, UGC & AICTE Approved
NAAC A++ Accredited

An internship report submitted by

ELVINA SHARON J - URK20CS1006

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING

under the supervision of

Dr. P. Getzi Leela Pushpam
Mr. Mohan Nikam

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

Karunya INSTITUTE OF TECHNOLOGY AND SCIENCES

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

MoE, UGC & AICTE Approved

NAAC A++ Accredited

DIVISION OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

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

ELVINA SHARON J URK20CS1006

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.P.Getzi Jeba Leelipushpam

Associate Professor

Department of Computer Science and Engineering

Team Name: Spacebar

Title: Social Distancing project using Computer Vision and Deep Learning

Team Member(s)

The project was carried out by the following team members:

• Elvina Sharon J – elvinasharon@karunya.edu.in

Mentor

We would like to express our gratitude to our mentors for providing guidance and support throughout the internship:

• Dr. P. Getzi Leela Pushpam(Academic Mentor) - getzi@karunya.edu

• Mr. Mohan Nikam (Industry Mentor) – mohannikam@theprograms.i

ABSTRACT

The "Social Distancing Project using Computer Vision and Deep Learning" aims to address the critical need for effective social distancing measures in public spaces, especially during contagious disease outbreaks like COVID-19. Leveraging advancements in computer vision and deep learning techniques, this project proposes an innovative solution to monitor and enforce social distancing guidelines.

The project utilizes a network of surveillance cameras equipped with computer vision algorithms to detect and track individuals in real-time. By analyzing the video feed, the system can accurately estimate the distance between people and identify instances of close proximity that violate social distancing norms. Additionally, the project incorporates deep learning models to distinguish between different activities, such as standing, walking, or running, to further enhance the accuracy of distance measurements.

To ensure user privacy and data security, the project employs advanced anonymization techniques that process video data without identifying individuals personally. The system generates real-time alerts or notifications whenever social distancing violations occur, enabling immediate intervention to mitigate the risk of disease transmission. These notifications can be sent to designated personnel or displayed on digital signage to promote public awareness and compliance.

The effectiveness of the proposed solution will be evaluated through extensive testing in various public environments, including retail stores, transportation hubs, and public parks. The project aims to provide valuable insights into social distancing compliance rates and patterns, facilitating evidence-based decision-making for public health authorities and policymakers.

In conclusion, the "Social Distancing Project using Computer Vision and Deep Learning" offers a promising approach to monitor and enforce social distancing guidelines in real-time. By harnessing the power of computer vision and deep learning, this project has the potential to significantly contribute to public health efforts during contagious disease outbreaks, ultimately helping to safeguard communities and reduce the spread of infections.

INTRODUCTION

Motivation

Traditional methods of monitoring social distancing rely heavily on manual intervention and are prone to human error. By harnessing the power of computer vision and deep learning, this project seeks to revolutionize the way social distancing is monitored and enforced. It aims to provide an automated, accurate, and real-time solution that can ensure adherence to social distancing guidelines in diverse public environments.

The project's motivation also lies in the potential to contribute to public health efforts by providing valuable insights into social distancing compliance rates and patterns. The data generated through extensive testing in various public spaces will enable evidence-based decision-making for public health authorities and policymakers. This information can guide the implementation of targeted interventions, facilitate resource allocation, and enhance the effectiveness of public health measures in reducing the transmission of contagious diseases.

Prior work/ Background

Previous studies in the field of social distancing monitoring have employed various detection methods. Some of these approaches have utilized AI and Deep Learning algorithms, while others have incorporated IoT motion sensors to detect human presence and activities. These methods have been implemented on real-time monitoring systems or utilized existing video and image data for analysis.

However, deploying these algorithms on local monitoring systems has presented several challenges in achieving accurate detection and analysis. To improve the precision of detection and enhance analysis capabilities, researchers have been exploring the emergence of more effective algorithms and procedures. Notably, technologies such as OpenVino and OpenCV, which encompass advanced Computer Vision algorithms, have been utilized to achieve better analysis results and refine the detection process.

PROJECT APPROACH

Source Compliers and Libraries

1. Python 3.10:

Python 3.10 is a popular programming language that can be used for artificial intelligence (AI) and deep learning tasks. While Python itself is a versatile language for various applications, it has gained significant popularity in the AI and deep learning communities due to its simplicity, extensive libraries, and vibrant ecosystem.

2. OpenCV:

OpenCV (Open Source Computer Vision) is an open-source library of computer vision and image processing functions. It provides a wide range of tools and algorithms that enable developers to perform various tasks related to image and video analysis, object detection and recognition, and more.

3. OpenVino:

OpenVINO (Open Visual Inference and Neural Network Optimization) is an opensource toolkit developed by Intel. It is designed to optimize and accelerate the deployment of deep learning models across a wide range of Intel hardware platforms, including CPUs, GPUs, FPGAs, and Neural Compute Stick devices.

4. Numpy:
NumPy is a popular numerical computing library for Python. It provides a powerful and efficient way to handle large arrays and matrices of numeric data, along with a wide range of mathematical functions for array manipulation and computation.

Project Details

The primary objective of this project is to develop a robust system that leverages the OpenVINO (Open Visual Inference and Neural Network Optimization) toolkit to perform social distance inference on video data. The purpose is to create an application capable of analyzing video footage and accurately detecting violations of social distancing guidelines, thereby contributing to the mitigation of contagious disease transmission.

To achieve this, the project will utilize the pre-trained deep learning model known as "yolov8" The OpenVINO toolkit's inference engine will be leveraged to optimize the performance of this model specifically for social distance estimation.
By employing OpenVINO, optimizing the yolov8model, and integrating it with video-related functionalities, this project aims to provide an effective solution for monitoring social distancing compliance. The ultimate goal is to enable real-time analysis of video footage, accurate identification of social distancing violations, and contribute to the broader efforts of mitigating the spread of contagious diseases.

Algorithm

1. Install necessary libraries in the Python compiler.
2. Perform model pre-processing and store the pre-processed model files in the designated directory.
3. Provide the path for the XML and BIN files of the pre-processed model.
4. Initialize the IECore (Inference Engine Core) and compile the model core with CPU as the processor for local system implementation.
5. Define the Euclidean distance function to calculate the distance between people.
6. Set the pixel-to-distance ratio, initializing it to 200 pixels representing 1 meter.
7. Define variables for frame height, width, and the destination path for storing results.
8. Initialize empty lists to store box coordinates, confidences, and class IDs.
9. Calculate the frames per second (FPS) using the appropriate formula and set the threshold value for detection confidence to 0.5.
10. Implement code to display bounding boxes around each detected person, along with the detection rate and the number of social distancing violations based on the calculated distance.

Result Analysis

OpenVINO significantly outperforms OpenCV in terms of accuracy when comparing them using similar parameters. The output provided by OpenVINO demonstrates a notably higher level of accuracy compared to OpenCV. OpenVINO exhibits higher frame rates, resulting in smoother output compared to OpenCV. Moreover, the inference time and computational resources required by OpenVINO are considerably lower than those demanded by OpenCV. The OpenVINO CPU inference engine, with its high optimization methods, competes impressively with OpenCV GPU, thereby leveling the field between the two frameworks.
By developing a social distance inference system using OpenVINO, this project aims to make a valuable contribution to public health efforts. The system serves as a tool for monitoring and enforcing social distancing guidelines across various settings, including public spaces,

workplaces, and events. Upon implementation and execution of the application website, comprehensive analysis of the results has been conducted, exploring various possibilities and potential outcomes. The screenshots of them are given below.

Model Results:

Optimized Model Results:

CONCLUSION

In conclusion, the Social Distancing Project using Computer Vision and Deep Learning has successfully developed a system that leverages the power of OpenVINO to perform social distance inference on video data. By optimizing a pre-trained deep learning model and integrating it with video processing functionalities, the project has achieved accurate and real-time detection of social distancing violations.

Comparative analysis with OpenCV has demonstrated the superior performance of OpenVINO in terms of accuracy, frame rates, inference time, and computational efficiency. OpenVINO's CPU inference engine, coupled with its high optimization methods, has proven to be a remarkable alternative to OpenCV's GPU implementation.

Furthermore, the project's utilization of computer vision and deep learning techniques showcases their potential in addressing real-world challenges. By combining these cutting-edge technologies, the project has paved the way for future advancements in social distancing monitoring and related applications.

In summary, the use of OpenVINO, alongside optimized deep learning models, has demonstrated exceptional accuracy, efficiency, and applicability in monitoring and enforcing social distancing guidelines. This project holds tremendous potential in supporting public health initiatives and safeguarding communities during contagious disease outbreaks.

REFERENCES

Github Link: **https://github.com/elvinasharon/Social-Distancing-using-computer-vision-and-deep-learning.git**

Model Link: **https://github.com/elvinasharon/Social-Distancing-using-computer-vision-and-deep-learning/blob/main/yolov8n.onnx**

Video link: **https://youtu.be/ZxIXXDFqQ00**

References:

- **https://github.com/openvinotoolkit/open_model_zoo**
- **https://docs.openvino.ai/2021.4/workbench_docs_Workbench_DG_Tutorial_Import_YOLO.html#import-model**
- **https://docs.openvino.ai/2022.3/openvino_docs_MO_DG_prepare_model_convert_model_tf_specific_Convert_YOLO_From_Tensorflow.html**