Social Distancing project using Computer Vision and Deep Learning

RENNAISANCE SQUAD

Sri Sairam College of Engineering Bengaluru

AJ Daya Sagar
Pooja Singh
Mahesh Kumar Chaudhary

ABSTRACT

Social distancing is one of the most effective ways to prevent the spread of COVID-19. By staying at least two meters apart from others, we can help to minimize the risk of infection. However, it can be difficult to maintain social distancing in crowded places. This is where computer vision technology can help. Computer vision systems can be used to detect and track people in a variety of ways. One common approach is to use convolutional neural networks (CNNs). CNNs are a type of machine learning algorithm that is specifically designed for image recognition. Another approach to computer vision for social distancing is to use depth cameras. Depth cameras can measure the distance between objects in a scene. This information can be used to calculate the distance between people and to alert them if they are too close together. Computer vision systems for social distancing can be deployed in a variety of settings. They can be used in public places, such as train stations and airports. They can also be used in workplaces and schools.

INTRODUCTION

Social distancing is a crucial measure in preventing the spread of contagious diseases, particularly in situations like pandemics. By maintaining a safe distance between individuals, the risk of transmitting infections is significantly reduced. However, enforcing social distancing guidelines in crowded public spaces can be challenging. To address this challenge, the social distancing project leverages OpenVINO by Intel, a computer vision-based solution. The project utilizes deep learning models to detect and track individuals in real-time, enabling the monitoring and enforcement of social distancing guidelines. By integrating OpenVINO's optimized model conversion and inference capabilities, the project provides efficient and accurate detection of social distancing violations. The project's target audience includes organizations, businesses, and public spaces where social distancing compliance is essential, such as hospitals, retail stores, airports, and public transportation systems. The project's impact is significant, as it improves public awareness and adherence to social distancing guidelines, reduces the risk of disease transmission in crowded spaces, enhances overall safety and well-being, and provides valuable data insights for authorities to manage and optimize social distancing measures. the necessary software requirements, including the Intel Distribution of OpenVINO Toolkit, Python, OpenCV, and other supporting libraries.

02

LITERATURE SURVEY

Covid-19 Social Distancing Detector Approach: Made By Basile Roth

The AI project developed by Basile Roth aims to detect and monitor social distancing compliance in public spaces. Using a Tensorflow object detection model trained on the faster_rcnn_inception_v2_coco dataset, the project identifies individuals in real-time. The system generates bounding boxes around each person and calculates the distance between their centroids to measure social distancing. If individuals maintain a safe distance, they are indicated with a green box; otherwise, the box turns red, highlighting potential risks. This project benefits everyone, including individuals going out for essential activities and social workers like policemen, by enabling them to monitor public spaces and ensure adherence to social distancing guidelines.

OpenCV Social Distancing Detector : Made by Adrian Rosebrock.

This code implements a social distancing detector using OpenCV, deep learning, and computer vision. It uses the YOLO object detector to detect people in video streams or webcam feeds. The detector computes pairwise distances between individuals and checks if they are maintaining a safe distance based on a predefined minimum distance. The implementation includes a configuration file to store variables, a utility function for person detection, and a driver script to process frames and detect social distancing violations. The code also allows for GPU acceleration and provides options to input and output video files. The detector can be extended and customized for different applications.

Social Distancing Detection with Deep Learning Model: Made by Yew Cheong Hou; Mohd Zafri Baharuddin

The paper presents a methodology for social distancing detection using deep learning to evaluate the distance between people to mitigate the impact of this coronavirus pandemic. The detection tool was developed to alert people to maintain a safe distance with each other by evaluating a video feed. The video frame from the camera was used as input, and the open-source object detection pre-trained model based on the YOLOv3 algorithm was employed for pedestrian detection. Later, the video frame was transformed into top-down view for distance measurement from the 2D plane. The distance between people can be estimated and any noncompliant pair of people in the display will be indicated with a red frame and red line. The proposed method was validated on a pre-recorded video of pedestrians walking on the street. The result shows that the proposed method is able to determine the social distancing measures between multiple people in the video. The developed technique can be further developed as a detection tool in realtime application.

Expected impact

The project to automate social distancing monitoring and enforcement is a valuable initiative that has the potential to make a significant impact on public health and safety. By leveraging OpenVINO and computer vision techniques, the project can provide real-time alerts and notifications to individuals and authorities when social distancing violations occur. This can help to improve public awareness and adherence to social distancing guidelines, which can in turn reduce the risk of disease transmission in crowded spaces. Additionally, the project can provide valuable data insights for authorities to manage and optimize social distancing measures.
Here are some of the specific benefits of the project:

- Improved public awareness and adherence to social distancing guidelines. By providing real-time alerts and notifications, the project can help to remind people to maintain a safe distance from others. This can help to improve public awareness of the importance of social distancing and encourage people to comply with the guidelines.
- Reduced risk of disease transmission in crowded spaces. When people are able to maintain a safe distance from each other, the risk of disease transmission is reduced. This is especially important in crowded spaces, where people are more likely to come into close contact with each other.
- Enhanced overall safety and well-being of individuals within public environments. By helping to reduce the risk of disease transmission, the project can help to enhance the overall safety and well-being of individuals within public environments. This is especially important for vulnerable populations, such as the elderly and the immunocompromised.
- Providing valuable data insights for authorities to manage and optimize social distancing measures. The project can also provide valuable data insights for authorities to manage and optimize social distancing measures. This data can be used to identify areas where social distancing is not being effectively implemented and to make adjustments to the measures as needed.

Overall, the project to automate social distancing monitoring and enforcement is a promising initiative that has the potential to make a significant impact on public health and safety. The project is still under development, but it has the potential to be a valuable tool for promoting and ensuring social distancing compliance.

-

Hardware Requirements:

• Intel CPU with Intel AVX2 or Intel DL Boost (Intel AI accelerator) support. • Integrated Intel GPU or Intel Neural Compute Stick 2 (NCS2) for accelerated inference. • Sufficient RAM and storage capacity to accommodate the dataset, models, and related files.

Software Requirements:

•Operating System: Linux* (Ubuntu* 18.04.5 LTS or CentOS* 7.6) or Windows* 10. • Intel Distribution of OpenVINO Toolkit: Version 2021.3 or higher. (Can be downloaded from the website) (https://software.intel.com/content/www/us/en/develop/tools/openvino-toolkit.html)

• Python: Version 3.6 or higher

.

• OpenVINE: Required for image and video processing tasks. Version 4.2.0 or higher is recommended.

•NumPy: Required for efficient numerical computations.

• Matplotlib: Optional, for visualizing the results and evaluation metrics. • Flask or Django: If developing a web-based user interface or application for deployment. Other Necessary Tools: • Annotation Tool: Any annotation tool of your choice to label the dataset with bounding boxes or other annotations for training the social distancing detection model.

• Model Optimizer: Part of the OpenVINO Toolkit, it converts trained models from popular deep learning frameworks into OpenVINO's Intermediate Representation (IR) format. • Model Zoo: The OpenVINO Model Zoo provides pre-trained models for various computer vision tasks, including object detection, segmentation, and tracking. It can be utilized for baseline models or reference purposes. It's worth noting that the hardware requirements can vary depending on the specific deployment scenario and the scale of the project. Additionally, always refer to the official Intel OpenVINO documentation for the most up-to-date information on hardware and software requirements.

Future advancements :

There are several potential enhancements and future directions for the social distancing project:
1. Multi-camera Support: Extend the project to support multiple camera feeds simultaneously, enabling comprehensive monitoring of social distancing compliance in larger areas or complex environments.

2. Real-Time Alerts: Implement real-time notifications or alerts to inform authorities or individuals when social distancing violations occur, facilitating prompt intervention and corrective actions.

3. Mask Detection: Integrate mask detection capabilities into the project to ensure compliance with face mask usage in addition to social distancing.

4. Crowd Density Analysis: Expand the project to include crowd density analysis, which can provide valuable insights into overcrowded areas and aid in crowd management.

5. Edge AI Optimization: Investigate techniques for optimizing the model further, such as model compression, quantization, or knowledge distillation, to improve its efficiency on edge devices with limited resources.

6. Continuous Model Training: Explore the implementation of online or incremental learning approaches to continuously update the model based on new data and evolving social distancing guidelines.

Conclusion:

The social distancing project aims to address the critical need for maintaining safe distances in public spaces. By utilizing the OpenVINO toolkit and deep learning techniques, an optimized model for social distancing detection is developed and deployed. Throughout the project, various steps, including dataset acquisition, preprocessing, model training, conversion, and deployment, have been undertaken to achieve accurate and efficient social distancing detection.
Challenges encountered during the project may include obtaining representative and diverse datasets, fine-tuning model hyperparameters for optimal performance, and balancing accuracy with real-time inference requirements. Continuous monitoring, evaluation, and iteration are essential for improving the model's accuracy, efficiency, and real-world applicability.
Additional Resources and Acknowledgments:

1. OpenVINO Toolkit Documentation: Intel provides comprehensive documentation and resources for the OpenVINO Toolkit, including installation guides, developer guides, and model optimization techniques. Refer to the official documentation for detailed information: https://docs.openvinotoolkit.org/

2. Deep Learning Frameworks: Consider exploring popular deep learning frameworks such as TensorFlow, PyTorch, or Caffe for model training and development.

3. Research Papers: Stay updated with the latest research papers and publications in the field of computer vision and social distancing detection to incorporate advancements and innovative techniques into your project.

References

- *https://brainypi.com/social-distance-monitoring-with-brainy-pi/*
- *https://ijtech.eng.ui.ac.id/article/view/5872*
- *https://www.researchgate.net/publication/346504501_A_deep_learning-based_social_distance_monitoring_framework_for_COVID-19*
- *https://www.qblocks.cloud/creators/social-distancing-detector-using-tensorflow-object-detection-model*
- *https://pyimagesearch.com/2020/06/01/opencv-social-distancing-detector/*
- *https://ieeexplore.ieee.org/document/9243478*
- *OpenCV AI Kit Social Distance Detection Tutorial – Raspberry Pi | Python | Computer Vision 2021 - YouTube: https://www.youtube.com/watch?v=6sQVmd2hjgU*
- *Social Distancing Analyzer using OpenCV and YOLO - GitHub: https://github.com/sherwyn11/Social-Distancing-Analyze*
- *Social Distance Detector Using Computer Vision, OpenCV and YOLO Deep Learning Algorithm - IRJET: https://www.irjet.net/archives/V9/i3/IRJET-V9I3100.pdf*
- *Social Distancing Detector using OpenCV | by Sanju Mehla - Medium: https://sanju-mehla3599.medium.com/social-distancing-detector-using-opencv-and-deep-learning-ab712c1882fc*