iNeuron

High Level Design (HLD)

Social Distance Detection System

Revision Number	1.0
Last date of revision	11/03/2022

Document Version Control

Date Issued	Version	Description	Author
11/03/2022	1	Initial HLD — V1.0	Sai Kumar

Contents

Abstract

The rampant coronavirus disease 2019 (COVID-19) has brought global crisis with its deadly spread to more than 180 countries, and about 3,519,901 confirmed cases along with 247,630 deaths globally as on May 4, 2020. The absence of any active therapeutic agents and the lack of immunity against COVID19 increases the vulnerability of the population. Since there are no proper vaccines available for complete prevention, social distancing is the only feasible approach to fight against this pandemic.

Motivated by this notion, this system proposes a deep learning based framework for automating the task of monitoring social distancing using surveillance video. The proposed framework utilizes the YOLO v3 object detection model to segregate humans from the background and Deepsort approach to track the identified people with the help of bounding boxes and assigned IDs. The results of the YOLO v3 model are further compared with other popular state-of-the-art models, e.g. faster region-based CNN (convolution neural network) and single shot detector (SSD) in terms of mean average precision (mAP), frames per second (FPS) and loss values defined by object classification and localization. To detect the distance between persons we use Euclidean Distance. Later, the pairwise vectorized L2 norm is computed based on the three-dimensional feature space obtained by using the centroid coordinates and dimensions of the bounding box. The violation index term is proposed to quantize the non adoption of social distancing protocol. From the experimental analysis, it is observed that the YOLO v3 with Deepsort tracking scheme displayed best results with balanced mAP and FPS score to monitor the social distancing in real-time.

iNeuron

1. Introduction

1.1 Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

- Present all of the design aspects and define them in detail
- Describe the user interface being implemented
- Describe the hardware and software interfaces
- Describe the performance requirements
- Include design features and the architecture of the project
- List and describe the non-functional attributes like:
    1. Security
    2. Reliability
    3. Maintainability
    4. Portability
    5. Reusability
    6. Application compatibility
    7. Resource utilization
    8. Serviceability

1.2 Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

1.3 Definitions

Term	Description
CNN	convolution neural network

mAP	mean average precision
SSD	single shot detector

2. General Description

2.1 Product Perspective

The Social Distancing Detection System is a deep learning based framework for automating the task of monitoring social distancing using surveillance video. The proposed framework utilizes the YOLO v3 object detection model to segregate humans from the background and Deepsort approach to track the identified people with the help of bounding boxes and assigned IDs.

2.2 Problem Statement

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

The main objective here is -

- First human detection.
- After that distance between them if any violation raises an ALERT.
- Maintain a database to store each and every data.

2.3 Proposed Solution

The emergence of deep learning has brought the best performing techniques for a wide variety of tasks and challenges including medical diagnosis, machine translation, speech recognition, and a lot more. Most of these tasks are centered around object classification, detection, segmentation, tracking, and recognition. In recent years, the convolution neural network (CNN) based architectures have shown significant performance improvements that are leading towards the high quality of object detection, which presents the performance of such models in terms of mAP and FPS on standard benchmark datasets, PASCAL-VOC and MS-COCO, and similar hardware resources. In the present article, a deep learning based framework is proposed that utilizes object detection and tracking models to aid in the social distancing remedy for dealing with the escalation of COVID-19 cases. In order to maintain the balance of speed and accuracy, YOLO v3 alongside the Deepsort are utilized as object detection and tracking approaches while surrounding each detected object with the bounding boxes. Later, these bounding boxes are utilized to compute the pairwise L2 norm with

computationally efficient vectorized representation for identifying the clusters of people not obeying the order of social distancing. Furthermore, to visualize the clusters in the live stream, each bounding box is color-coded based on its association with the group where people belonging to the same group are represented with the same color. Each surveillance frame is also accompanied with the streamline plot depicting the statistical count of the number of social groups and an index term (violation index) representing the ratio of the number of people to the number of groups. Furthermore, estimated violations can be computed by multiplying the violation index with the total number of social groups.

2.4. Further Improvements

Since this application is intended to be used in any working environment; accuracy and precision are highly desired to serve the purpose. Higher number of false positives may raise discomfort and panic situations among people being observed. There may also be genuinely raised concerns about privacy and individual rights which can be addressed with some additional measures such as prior consents for such working environments, hiding a person's identity in general, and maintaining transparency about its fair uses within limited stakeholders.

2.5. Theory of Social Distancing Detection System

Object detection:

- We will be using YOLOv3, trained on the COCO dataset for object detection.
- In general, single-stage detectors like YOLO tend to be less accurate than two-stage detectors (R-CNN) but are significantly faster.
- YOLO treats object detection as a regression problem, taking a given input image and simultaneously learning bounding box coordinates and corresponding class label probabilities.
- It is used to return the person's prediction probability, bounding box coordinates for the detection, and the centroid of the person.

Distance calculation:

- NMS (Non-maxima suppression) is also used to reduce overlapping bounding boxes to only a single bounding box, thus representing the true detection of the object. Having overlapping boxes is not exactly practical and ideal, especially if we need to count the number of objects in an image.
- Euclidean distance is then computed between all pairs of the returned centroids. Simply, a
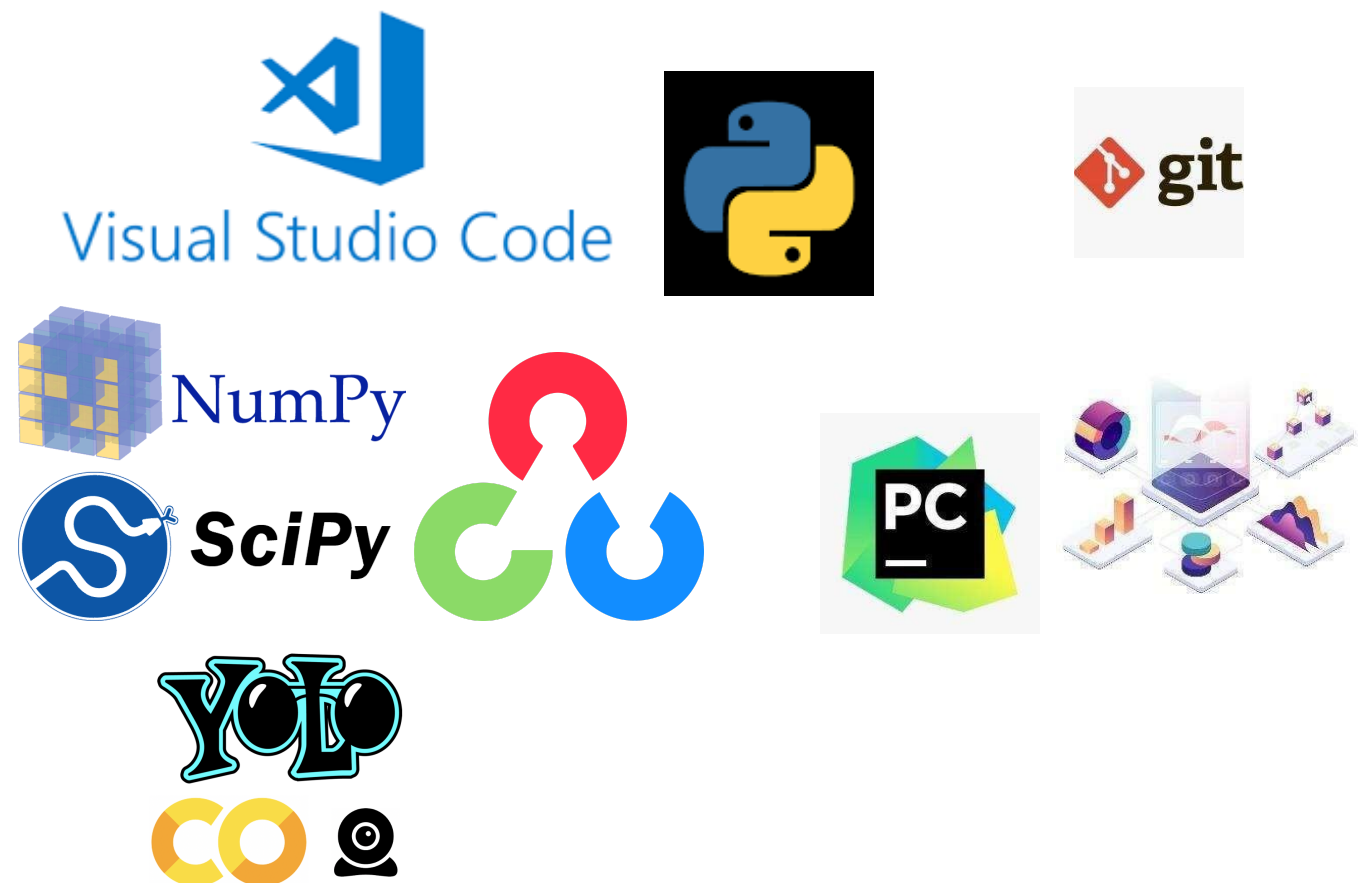
centroid is the center of a bounding box.
- Based on these pairwise distances, we check to see if any two people are less than/close to 'N' pixels apart.

2.6. Tools Used

Python programming language and its framework such as Django, Db sqlite-3, Html, Javascript, Jinja2, Css, Bootstrap and agora are used to build the whole model.

- VsCode / PyCharm can be used as IDE.
- For Human detection yolo.
- For Gpu runtime Google Collab is used.
- Numpy used to manage array structure.
- SciPy is an open-source Python library which is used to solve scientific and mathematical problems.
- OpenCV is a great tool for image processing and performing computer vision tasks.
- GitHub is used as a version control system.

2.7.   Constraints

The Social Distancing Detection system must be user friendly, as automated as possible and users should not be required to know any of the workings.

2.8.   Assumptions

The main objective of the project is to implement the use cases as previously mentioned (2.2 Problem Statement) to enable the process of automating the task of monitoring social distancing using surveillance video. Since there are no proper vaccines available for covid-19 complete prevention, social distancing is the only feasible approach to fight against this pandemic. Yolo and open cv are used for holding the above-mentioned use cases via google collab for gpu runtime. It is also assumed that all aspects of this project have the ability to work together in the way the designer is expecting.

# 3. Design Details

3.1. Process Flow

For identifying the different types of anomalies, we will use a deep learning base model. Below is the process flow diagram as shown below.

Proposed Methodology
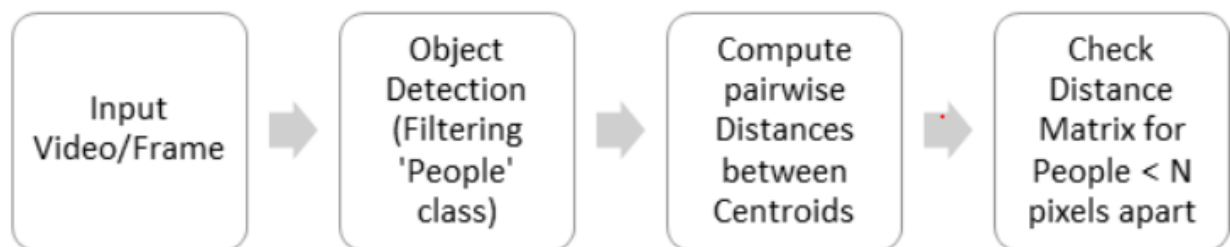
1.   Apply object detection to detect all people (and only people) in a video stream.

2.   Compute the pairwise distances between all detected people.

3.   Based on these distances, check to see if any two people are less than N pixels apart

This social distancing detector implementation will rely on pixel distances, which won't necessarily be as accurate.
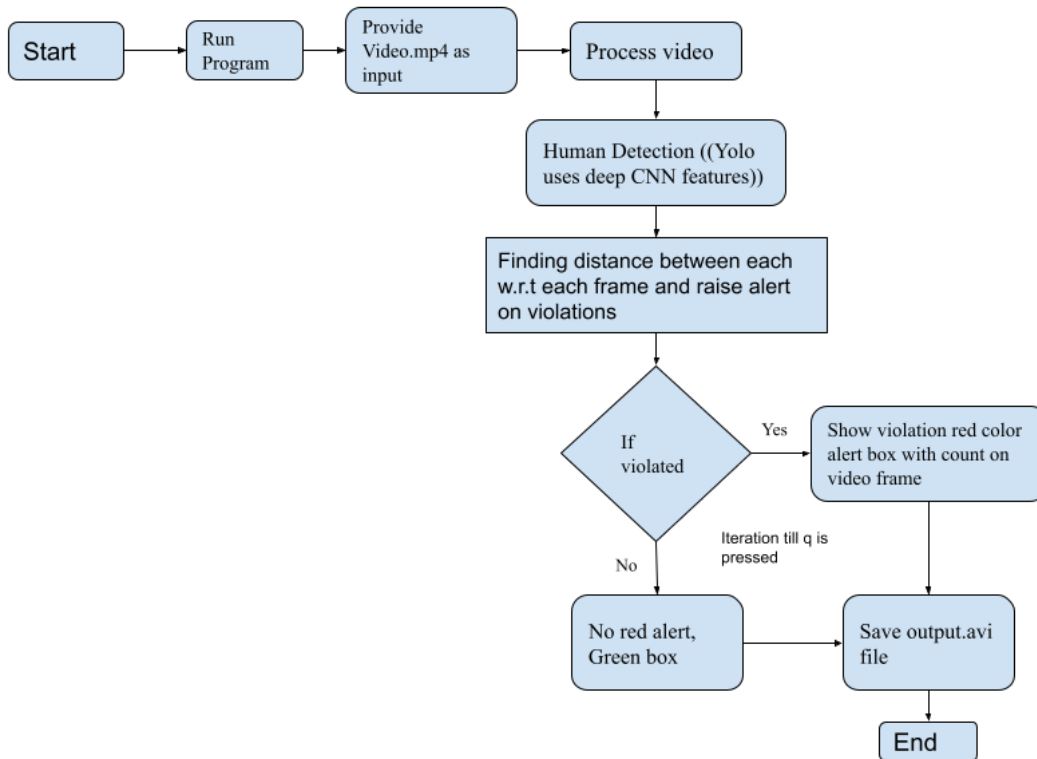
3.1.1. Deployment Process

3.2. Event log

The system should log every event so that the user will know what process is running internally.

Initial Step-By-Step Description:

1. The System identifies at what step logging required
2. The System should be able to log each and every system flow.
3. Developers can choose logging methods. You can choose database logging/ File logging as well.
4. System should not hang even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

3.2.1. Error Handling

Should errors be encountered, an explanation will be displayed as to what went wrong? An error will be defined as anything that falls outside the normal and intended usage.

4. Performance

Proper internet, best hardware with scalable resources is needed in order to hold good performance.

4.1. Reusability

The code written and the components used should have the ability to be reused with no problems.

4.2. Application Compatibility

The different components for this project will be using Python as an interface between them. Each component will have its own task to perform, and it is the job of the Python to ensure proper transfer of information.

4.3. Resource Utilization

When any task is performed, it will likely use all the processing power available until that function is finished.

5. Conclusion

The Social Distancing Detection system proposes an efficient real-time deep learning based framework to automate the process of monitoring the social distancing via object detection and tracking approaches, where each individual is identified in the real-time with the help of bounding boxes. The generated bounding boxes aid in identifying the clusters or groups of people satisfying the closeness property computed with the help of pairwise vectorized approach. The number of violations are confirmed by computing the number of groups formed and violation index term computed as the ratio of the number of people to the number of groups.

The extensive trials were conducted with popular state-of-the-art object detection models: Faster RCNN, SSD, and YOLO v3, where YOLO v3 illustrated the efficient performance with balanced FPS and mAP score. Since this approach is highly sensitive to the spatial location of the camera, the same approach can be fine tuned to better adjust with the corresponding field of view.

6. References

1. https://en.wikipedia.org/wiki/Social_distancing
2. Google.com for images of tools used.
3. Research paper used : https://arxiv.org/pdf/2005.01385.pdf