

Social Distance Detection System

Wireframe Documentation

Social Distancing Detection System:

Before running the program the user has to mount the drive and should have respective files in the respective folders.

- a. If a user runs code he will be asked for input video file
- b. Once video file given as input and run the program, the process will start as follows:

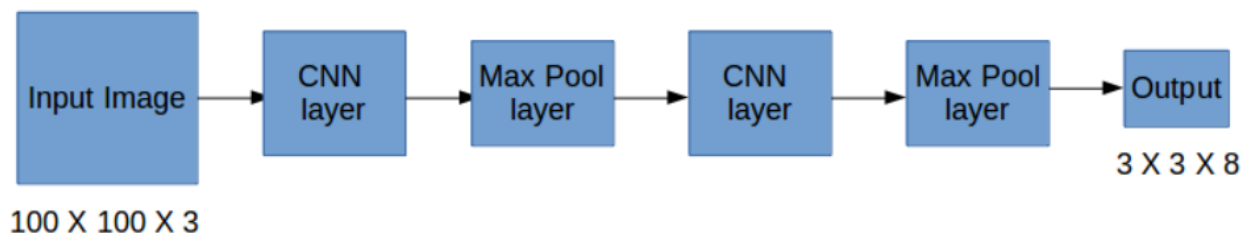


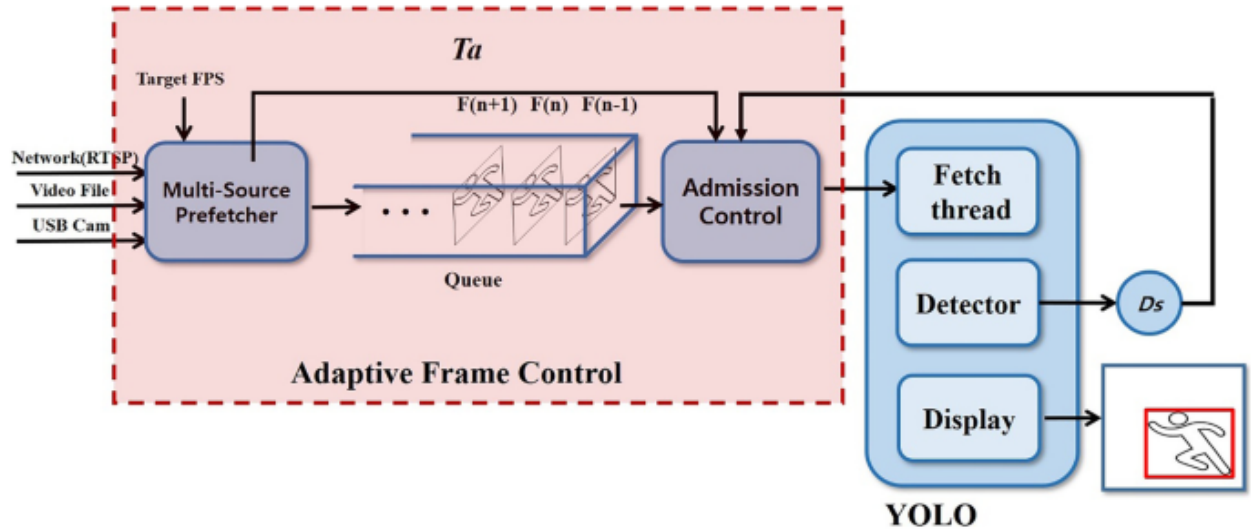
Fig. i

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.

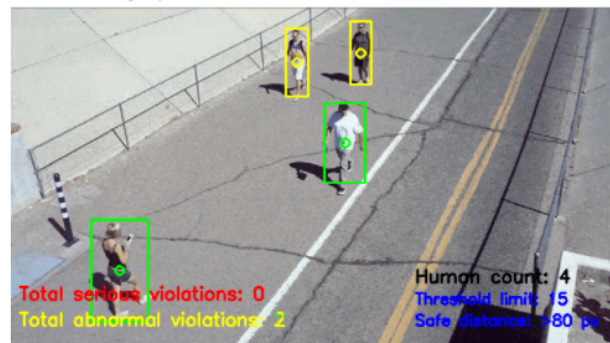
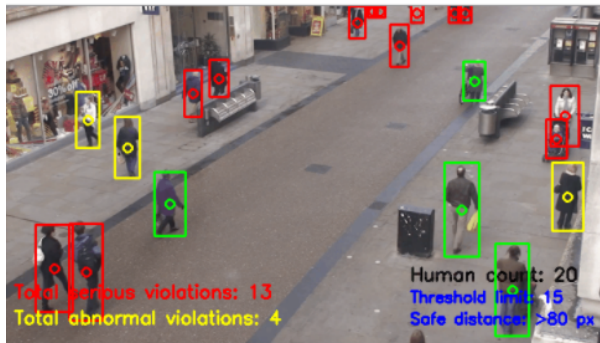
Distance calculation:

- 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.



Features:

The following are examples of the added features. Note: You can easily on/off them in the config. options (mylib/config.py):



1. Real-Time alert:

- If selected, we send an email alert in real-time. Use case: If the total number of violations (say 10 or 30) exceeded in a store/building, we simply alert the staff.
- You can set the max. violations limit in config (Threshold = 15).
- This is pretty useful considering the COVID-19 scenario.

Note: To set up the sender email, please refer to the instructions inside 'mylib/mailer.py'. Setup receiver email in the config.

2. Threading:

- Multi-Threading is implemented in 'mylib/thread.py'. If you ever see a lag/delay in your real-time stream, consider using it.

- Threading removes OpenCV's internal buffer (which basically stores the new frames yet to be processed until your system processes the old frames) and thus reduces the lag/increases fps.
- If your system is not capable of simultaneously processing and outputting the result, you might see a delay in the stream. This is where threading comes into action.
- It is most suitable for solid performance on complex real-time applications. To use threading:

set Thread = True in the config.

3. People counter:

- If enabled, we simply count the total number of people: set People_Counter = True in the config.

4. Desired violations limits:

- You can also set your desired minimum and maximum violations limits. For example, MAX_DISTANCE = 80 implies the maximum distance 2 people can be closer together is 80 pixels. If they fell under 80, we treat it as an 'abnormal' violation (yellow).
- Similarly MIN_DISTANCE = 50 implies the minimum distance between 2 people. If they fell under 50 px (which is closer than 80), we treat it as a more 'serious' violation (red).
- Anything above 80 px is considered as a safe distance and thus, 'no' violation (green).