International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 05 Issue: 06 | June - 2021 ISSN: 2582-3930

Social Distancing Detection Using Deep Learning

Prof. Dhivya S[1], Abarna R[2], Akash A[3], Dhatchana priya A[4]

[1]*Computer Science and Engineering & Sri Shakthi Institute of Engineering and Technology*
[2]*Computer Science and Engineering & Sri Shakthi Institute of Engineering and Technology*
[3]*Computer Science and Engineering & Sri Shakthi Institute of Engineering and Technology*
[4]*Computer Science and Engineering & Sri Shakthi Institute of Engineering and Technology*

---***---

Abstract - The worldwide spread of the COVID-19 pandemic has made hygiene, an unavoidable part of human life. As we are speaking of hygiene, in addition to washing hands frequently, using a mask, cleaning utensils, keeping a social distance is also a mandatory part. If the social distance is followed by every individual, the curve of coronavirus can be flattened. The governments are adopting restrictions over the minimum inter-personal distance between people. Also, the national health authorities have set the 2-meter physical distancing as a mandatory safety measure in the grocery store, schools, shopping malls, and other covered areas. This paper presents the detection of people in the area using YOLOv3 and ensuring whether the persons were following the social distancing by calculating the Euclidean distance between pairs. Based on these pairwise distances, we can check if any two people are less than/close to 'N' pixels apart. From the live-streaming camera, the YOLOv3 model is used to detect person class by bounding box coordinates and the centroid of the person. If the persons are not following the minimum inter-personal distance of 2-meter, they will be given alerts to maintain the social distancing.

Key Words: *COVID-19, Social distancing, Object detection and tracking, YOLO, Real time, Deep learning.*

1.INTRODUCTION

Since from the day humans beings evolved, socializing has become a greater part of it. They went for hunting as groups, sharing the food they brought from hunting. Humans love to socialize. They are social species; they interact with each other in everyday basis to achieve day-to-day goals, exchange ideas and so on. However, the recent COVID-19 pandemic emergency has caused a lot of challenges to these social species. Coronavirus is an infectious disease caused by the corona virus-2 extreme acute respiratory syndrome. COVID-19 belongs to the family of coronavirus caused diseases, first reported at Wuhan, China, during late December 2020. From then it is spreading worldwide in an increasing curve. When in close contact, the virus spreads mainly between individuals, including by tiny droplets formed when sneezing or coughing. Droplets falling on the ground will pass through the air through the body of a human.

The coronavirus that causes COVID-19 mainly spreads from person to person. Transmission from person to person can happen through larger droplets from sneezes and coughs but there is also growing evidence that smaller particles called aerosols can hang in the air longer and travel farther. These aerosols may also play a part in transmission. From what we know so far – transmission from surfaces is much lower risk than person to person. Still, it is possible (though not as likely) to catch the virus if you touch a surface or object that has the virus on it and then touch your mouth, nose, or eyes.

The most responsible reasons for it include the large human population gathering and human intervention which have violated the ecosystem globally. The population of the world intrudes the various untouched ecologies and put themselves before the unknown viruses and bacteria without knowing their threat or impact of exposure to humankind. Seeing the devastating spread of the disease, the World Health Organization (WHO) suggested favoring the term social distancing. After the COVID-19 pandemic, the CDC changed the concept of social distancing as keeping out of congregate environments, preventing public meetings, and preserving, when appropriate, a gap of around six feet or two meters from everyone.

Since, all the social gatherings, meetings, religious gatherings were restricted by government. Lockdown is also followed by lot of countries to flatten the increasing curve. After some days, by following strict self-isolation and lockdown. The normal economic activites were restarted by following social distancing and hygiene practices. As the people were excited to work again, they often forgot or neglect to maintain the social distancing with their colleague. So, to ensure the social distancing deep learning and computer vision came into play. There are different ways for object detection and tracking. In our model, it detects people in real-time and calculate euclidean distanc of 2 people. Based on the pairwise distance, we can check if any 2 people are less/close to 'N' pixels. The following points summarize the main components of this approach:

a. Deep learning has gained more attention in object detection (YOLO) was used for human detection purposes.

b. Develop a social distancing detection model that can detect the distance between people to keep safe.

c. Evaluation of the classification results by analyzing real-time video streams from the camera.

2. RELATED WORKS AND LITERATURE

In recent years, object detection techniques using deep models [2] are potentially more capable than shallow models in handling complex tasks and they have achieved spectacular progress in computer vision. Deep models for person detection focus on feature learning [3] contextual information learning, and occlusion handling. Deep learning object detection models [4] can now mainly be divided into two families: (i) two-stage detectors such as R-CNN[5], Fast R-CNN[6] and Faster R-CNN[7] and their variants and (ii) one-stage detectors such as YOLO[8] and SSD. In two-stage detectors detection is performed in stages, in the first stage, computed proposals and classified in the second stage into object categories. However, some methods, such as YOLO, SSD MultiBox, consider detection as a regression issue and look at the image once for detection. In proposed system we are using Single Shot Detector MultiBox(SSD) which seems to be a good choice for real-time object detection and the accuracy trade-off is also very little. SSD uses the VGG-16 model pre-trained on ImageNet as its basic model to extract useful image feature. At the top of VGG16, SSD adds several convolutional feature layers of decreasing sizes.

The Viola – Jones[9] object detection system can be trained to detect any object, but is especially common for facial detection and is more accurate and faster. The Viola and Jones process is an example of supervised learning. Zhu[10] also shared another very widespread facial detection algorithm is a neural network-based detector. It only works well with the front, upright face. Li et al. [11, 12], suggested another model for facial detection which was a MultiView Face Detector with surf capabilities. Oro et al. [13] also proposed a haar-like feature based face detection algorithm for HD video on the GTX470 and obtained an improved speed of 2.5 times. However, they only used CUDA which is a GPU programming tool for NVIDIA GPUs. Compared to OpenCL which is used in a number of computed components, it is unable to resolve the imbalanced workload issue experienced during the implementation of the viola-jones face detection algorithm in GPUs. Glass et al. (2006)[14] addressed the importance of social differencing and how the risk of pandemic growth can be slowly decreased by successfully preserving social distance without the use of vaccines or antiviral drugs. The authors have carried out an exhaustive study on this in both rural and urban communities in order to demonstrate a reduction in the growth rate. Z., Luo[15] studies the identification of people with full-face or partial occlusion. This approach categorizes into way, people with hand over their faces or occluded with objects. This approach is not suited to our scenario, which requires, in essentially, to detect faces that have their mouths covered with masks such as scarves, mufflers, handkerchiefs, etc.

Recent developments showed that the identification of individuals through video surveillance cameras can be achieved by face, and a person's manner of walking. However, the detection of a person under crowds' technique is difficult and hard to optimize.

3. OBJECTIVE

a. No physical surveillance needed:

The main objective of the project is to reduce the human surveillance. As the pandemic hits, there are lot of healthcare workers who is been working day and night for the hygiene of our human kind. The police surveillance is also needed to maintain the social distancing in the public areas. There is lot of human power and physical surveillance is needed to ensure there is no violation on social distancing. To reduce the human intervention in surveillance, this system will be of good use. It is a risk of going surveillance in the public places. Since the person will be at the risk of getting infected by the COVID-19.

b. Reduce the cost of surveillance:

Some of the methods which were proposed by researchers needed lot of hardware devices. As every individual are needed to wear some sensor devices to get alerted and more IoT devices to detect the distance between people. Here in our proposed system, we reduce the cost needed for the highest IOT devices and to maintain a portable system which can be carried anywhere to ensure the social distancing.

c.* Flatten the corona wave*:

The proposed system is very much helpful in flattening the corona wave which has been going in peak. One of the reasons for the spread of corona virus is human contact. As the corona virus can spread through air, close contact of human must be avoided to escape the risk of infection. This system can give alert once the person doesn't follow social distancing. This alert can give awareness to the people that they are been in close contact with other.

4. PROPOSED SYSTEM

Many researchers in the medical and pharmaceutical fields are aiming at the treatment of COVID-19 infectious disease; however, no definite solution has yet been found. On the other hand, controlling the spread of the virus in public places is another issue. People forget or neglect to follow the social distancing, where AI, computer vision, and technology can step in. Initially, the admin will be logged into the page and the page triggers the python script. Then the input data is fetched from the live-streaming camera. From this input video, each frame will be captured by OpenCV. The captured frames will be given to the YOLOv3 model which is trained on the COCO dataset of 80 classes. The model constructs a blob from the input frame and then performs a forward pass of the YOLO object detector, giving us our bounding boxes and associated probabilities. After the forward pass and given bounding boxes, filter detections by ensuring that the object detected as a person and that the minimum confidence is met. Then apply non-maxima suppression to suppress weak, overlapping bounding boxes. From the final detection, the Euclidean distance and pairwise distance will be found. If there is any violation, an alert will be sent to the default mail address in the model and an alarm sound will be given by the buzzer which we have connected in our Arduino UNO board.

In this module, the administrator have to login. Only admin have their username and password. Initially there will be some information on social distancing to avoid infection. Then the admin can login by the Login button below, which will take them to the admin login page. As soon as the admin logins, he/she will be directed to this page. Admin have to enter the userID and password to login. After successful login, the page is refreshed and the Python script which is responsible for the Social distancing detection is triggered. This page will be seen as a window with the input data.

Once the admin login the credentials will be verified and message box alert of successful login. Then the page directs the python script which holds the model for social distancing detection. This is the crucial point of the module, where the rela-time live streaming video input will be feed by the external camera. The detector model will be detecting the people only class. Then Euclidean distance will be calculated. After successful calculation pairwise distance will be determined, which will convey if there is any violation based on the threshold we have provided. Once there any violation occurs, the buzzer which we have connected in the Arduino UNO board will give an alert. Also a mail will be triggered to the provided email ID.

5. METHODOLOGY

As per the information and knowledge we've gathered, we have a tendency to found that these following tasks should be allotted so as to induce correct predictions. The tasks that we have a tendency to are about to do are given by

- Loading Object Detector Model
- Input data
- Detection and anchor box
- Calculation of Pairwise distance
- Alarm system on violation

Loading Object Detector Model:

The object detector model YOLOv3 is loaded. Before loading the YOLOv3 model, we have to load the COCO class labels our YOLO model was trained on. Here we use only people class for object detection. Then YOLO object detector trained on COCO dataset (80 classes) is loaded. Finally it creates the network on YOLO weights and config file.

Input data :

The input data can be given with as per the user. They can provide a stored video or they can choose to provide a live streaming video as input data. In this part the frames will be captured by OpenCV module. The captured frames will be resized to a standard size.

Detection and Anchor box:

We create three empty list i.e., class_ids, confidences and boxes. Firstly scores, will be calculated. The class_ids will contain all the class (object) id's of the objects which will be detected. If confidence>0.5 (50%), then the object will be detected. Only if the confidence is greater than the confidence threshold, then only class_id, confidence and scores will be appended in the list. The scores contains (x,y,w,h) of the rectangle. The Non-max suppression removes multiple boxes around the same object. From boxes, we extract the x,y,w,h coordinates of the object and label them with their class_ids.

Calculation of pairwise distance:

Once the anchor boxes were drawn, we initiated a variable close person which later will be used to generate alert. Then initiated a variable off to calculate no. of offenders. Run two loops in final list which contains coordinates of humans detected. Hence, we calculate distance of every human with each other and the distance is stored in d variable. If d is less than N pixel, then we will consider that they are walking too close. Then, we change the color of the box into red.

Alarm system on violation:

If the persons are too close and violate the social distancing, the system will show the details in the frames. Then the alert on the violation will be sent to the mail address we have specified in the model itself. Also the alert using buzzer which we have connected through Arduino UNO board will be given on spot. This alert can make people aware of being close and to maintain social distancing the environment.

REGION BASED CNN (R-CNN):

Region-based convolutional neural networks or regions with CNN features (R-CNNs) are a pioneering approach that applies deep models to object detection. R-CNN models first select several proposed regions from an image (for example, anchor boxes are one type of selection method) and then label their categories and bounding boxes. Then, they use a CNN to perform forward computation to extract features from each proposed area. Afterwards, we use the features of each proposed region to predict their categories and bounding boxes. R-CNNs are composed of four main parts:

- Selective search is performed on the input image to select multiple high-quality proposed regions. These proposed regions are generally selected on multiple scales and have different shapes and sizes. The category and ground-truth bounding box of each proposed region is labeled.
- A pre-trained CNN is selected and placed, in truncated form, before the output layer. It transforms each proposed region into the input dimensions required by the network and uses forward computation to output the features extracted from the proposed regions.
- The features and labeled category of each proposed region are combined as an example to train multiple support vector machines for object classification. Here, each support vector machine is used to determine whether an example belongs to a certain category.
- The features and labeled bounding box of each proposed region are combined as an example to train a linear regression model for ground-truth bounding box prediction.

Given a predicted bounding box coordinate p=(px,py,pw,ph) (center coordinate, width, height) and its corresponding ground truth box coordinates g=(gx,gy,gw,gh), the regressor is configured to learn scale-invariant transformation between two centers and log-scale transformation between widths and heights. All the transformation functions take p as input.

$$\hat{g}_x = p_w d_x(\mathbf{p}) + p_x$$
$$\hat{g}_y = p_h d_y(\mathbf{p}) + p_y$$
$$\hat{g}_w = p_w \exp(d_w(\mathbf{p}))$$
$$\hat{g}_h = p_h \exp(d_h(\mathbf{p}))$$

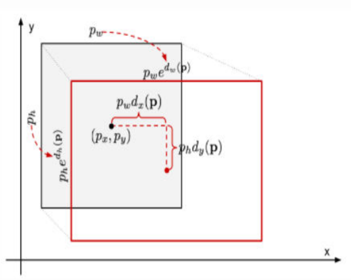


Fig. 1.The predicted bounding box coordinate

An obvious benefit of applying such transformation is that all the bounding box correction functions, di(p) where i∈{x,y,w,h}, can take any value between [-∞, +∞]. The targets for them to learn are:

$$t_x = (g_x - p_x)/p_w$$

$$t_y = (g_y - p_y)/p_h$$

$$t_w = log(g_w/p_w)$$

$$t_h = log(g_h/p_h)$$

Fig. 2.The targets

The regularization term is critical here and RCNN paper picked the best λ by cross validation. It is also noteworthy that not all the predicted bounding boxes have corresponding ground truth boxes.

FAST R-CNN:

To make R-CNN faster, Girshick improved the training procedure by unifying three independent models into one jointly trained framework and increasing shared computation results, named Fast R-CNN. Instead of extracting CNN feature vectors independently for each region proposal, this model aggregates them into one CNN forward pass over the entire image and the region proposals share this feature matrix. Then the same feature matrix is branched out to be used for learning the object classifier and the bounding-box regressor.

FASTER R-CNN:

An intuitive speedup solution is to integrate the region proposal algorithm into the CNN model. Faster R-CNN is doing exactly this: construct a single, unified model composed of RPN (region proposal network) and fast R-CNN with shared convolutional feature layers.

MASK R-CNN:

Mask R-CNN extends Faster R-CNN to pixel-level image segmentation. The key point is to decouple the classification and the pixel-level mask prediction tasks. Based on the framework of Faster R-CNN, it added a third branch for predicting an object mask in parallel with the existing branches for classification and localization. The mask branch is a small fully-connected network applied to each RoI, predicting a segmentation mask in a pixel-to-pixel manner. Because pixel-level segmentation requires much more fine-grained alignment than bounding boxes, mask R-CNN improves the RoI pooling layer so that RoI can be better and more precisely mapped to the regions of the original image.

SINGLE SHOT DETECTION (SSD):

The Single Shot Detector is one of the first attempts at using convolutional neural network's pyramidal feature hierarchy for efficient detection of objects of various sizes. SSD uses the VGG-16 model pre-trained on ImageNet as its base model for extracting useful image features. On top of VGG16, SSD adds several conv feature layers of decreasing sizes. They can be seen as a pyramid representation of images at different scales. Intuitively large fine-grained feature maps at earlier levels are good at capturing small objects and small coarse-grained feature maps can detect large objects well. In SSD, the detection happens in every pyramidal layer, targeting at objects of various sizes. Unlike YOLO, SSD does not split the image into grids of arbitrary size but predicts offset of predefined anchor boxes for every location of the feature map. Each box has a fixed size and position relative to its corresponding cell. All the anchor boxes tile the whole feature map in a convolutional manner.

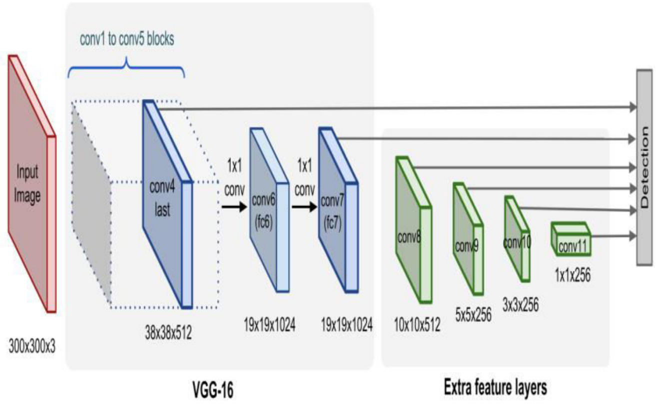


Fig. 3. The model architecture of SSD

Feature maps at different levels have different receptive field sizes. The anchor boxes on different levels are rescaled so that one feature map is only responsible for objects at one particular scale. The width, height and the center location of an anchor box are all normalized to be (0, 1). At a location (i,j) of the ℓ-th feature layer of size m×n, i=1,…,n,j=1,…,m, we have a unique linear scale proportional to the layer level and 5 different box aspect ratios, in addition to a special scale when the aspect ratio is 1. This gives us 6 anchor boxes in total per feature cell.

At every location, the model outputs 4 offsets and cc class probabilities by applying a 3×3×p conv filter for every one of k anchor boxes. Therefore, given a feature map of size m×n, we need kmn(c+4) prediction filters. Same as YOLO, the loss function is the sum of a localization loss and a classification loss.

$$\mathcal{L} = \frac{1}{N}(\mathcal{L}_{\text{cls}} + \alpha\mathcal{L}_{\text{loc}})$$

Fig. 4. Loss function of SSD

Where N is the number of matched bounding boxes and αα balances the weights between two losses, picked by cross validation. The localization loss is a smooth L1 loss between the predicted bounding box correction and the true values. The coordinate correction transformation is same as what R-CNN does in bounding box regression.

YOU ONLY LOOK ONCE (YOLO):

The YOLO model is the very first attempt at building a fast real-time object detector. Because YOLO does not undergo the region proposal step and only predicts over a limited number of bounding boxes, it is able to do inference super-fast. It is a state-of-the-art object detection algorithm that is incredibly fast and accurate. In practical it runs a lot faster than faster R-CNN due its simpler architecture. Unlike faster RCNN, it's trained to do classification and bounding box regression at the same time.

The loss consists of two parts, the localization loss for bounding box offset prediction and the classification loss for conditional class probabilities. Both parts are computed as the sum of squared errors. Two scale parameters are used to control how much we want to increase the loss from bounding box coordinate predictions (λcoord) and how much we want to decrease the loss of confidence score predictions for boxes without objects (λnoobj). Down-weighting the loss contributed

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 05 Issue: 06 | June - 2021 ISSN: 2582-3930

by background boxes is important as most of the bounding boxes involve no instance. In the paper, the model sets λcoord=5and λnoobj=0.5.

$$\mathcal{L}_{\text{loc}} = \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{obj}} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2]$$

$$\mathcal{L}_{\text{cls}} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} (1_{ij}^{\text{obj}} + \lambda_{\text{noobj}}(1 - 1_{ij}^{\text{obj}}))(C_{ij} - \hat{C}_{ij})^2 + \sum_{i=0}^{S^2} \sum_{c \in \mathcal{C}} 1_i^{\text{obj}} (p_i(c) - \hat{p}_i(c))^2$$

$$\mathcal{L} = \mathcal{L}_{\text{loc}} + \mathcal{L}_{\text{cls}}$$

Fig. 5. Loss function of YOLO

The loss function only penalizes classification error if an object is present in that grid cell, $1^{obj}_{i}=1$. It also only penalizes bounding box coordinate error if that predictor is "responsible" for the ground truth box, $1^{obj}_{ij}=1$. As a one-stage object detector, YOLO is super-fast, but it is not good at recognizing irregularly shaped objects or a group of small objects due to a limited number of bounding box candidates. YOLOv3 is created by applying a bunch of design tricks on YOLOv2.

6. TECHNOLOGY USED

Anaconda

Anaconda distribution could be a free and open source distribution of the R and Python programming languages for computing such as data science, machine learning applications, large-scale data processing, predictive analytics, etc. that aims to change package management and preparation. Anaconda has many packages further as conda packages and virtual atmosphere. It conjointly includes a user interface referred to as Anaconda Navigator.

Anaconda Navigator

Anaconda Navigator can be a graphical desktop interface included in Anaconda Navigator Distribution that allows users to launch applications and control conda packages, environment, and networks. Navigator can rummage around for packages on the Anaconda Navigator Cloud or in a very native Anaconda Navigator Repository, install them in an environment, run and upgrade the packages. It is on the Windows, macOS, and UNIX market.

Spyder

Spyder (Scientific Python Development Environment) is an integrated software environment (IDE) with open source cross-platform for scientific programming in the Python language. Spyder also incorporates NumPy, SciPy, Matplotlib, Pandas, IPython, SymPy and Cython as an alternative open source program with a range of excellent packages within the scientific Python stack. It is cross-platform on the market via Anaconda.

Python

Python is a general programming language, which is high-level. The language constructs and object-oriented methodology are aimed at helping programmers write simple, logical code that comes in for small and large scale. This embraces many programming paradigms, as well as programming that is procedural, object-oriented and sensible.

MySQL Workbench

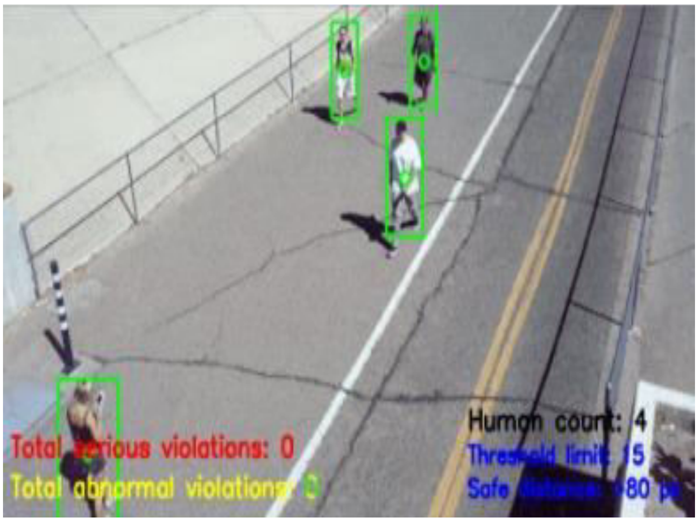
MySQL Workbench is a unified visual tool for database architects, developers, and DBAs. MySQL Workbench is available on Windows, Linux and Mac OS X. MySQL Workbench enables a DBA, developer, or data architect to visually design, model, generate, and manage databases. MySQL Workbench delivers visual tools for creating, executing, and optimizing SQL queries. MySQL Workbench provides a visual console to easily administer MySQL environments and gain better visibility into databases.

Tkinter

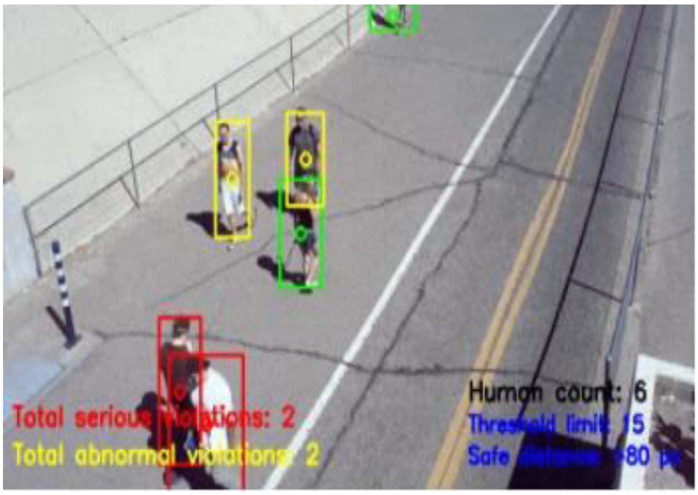
Python Tkinter is the standard Graphical User Interface (GUI) that is supported in Python. When Tkinter is used alongside Python it results in the creation of GUI very conveniently and quite fast. The main advantage of using Tkinter is that it has an object-oriented interface. It's cross-platform, so the same code works on Windows, macOS, and Linux. Visual elements are rendered using native operating system elements, so applications built with Tkinter look like they belong on the platform where they're run.

6. OUTCOME

Here is the outcome of the proposed system. As the green boxes indicate the persons who are at safe distance and the red boxes indicate those who are violating the safe distance. The yellow boxes show the count of abnormal violation. The output also shows the number of humans in the frame (Human count), Threshold limit and the safe distance in pixels.



(a) Social distancing followed by pedestrians-Green box



(b)Social distancing not followed by pedestrians-Red box

REFERENCE

1. S.Wang Chen, Horby Peter W, Hayden Frederick G, Gao George F. A novel coronavirus epidemic of global concern for health. It's the Lancet. 2020;395(10223):470-473. 10.1016 / S0140-6736(20)30185-9.
2. Lin, Tsung-Yi, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie, "Type Pyramid Networks for Object Detection," IEEE Conference Proceedings on Computer Vision and Pattern Recognition, pp. 2117-2125. 2017.
3. GOODFELLOW, I., BENGIO, Y., & COURVILLE, A. (2016). Deep learning process. Chapter 6.
4. S. S. Farfade, M. J. Saberian, and L. Li. Multi-view face recognition using deep convolutional neural networks. In ACM ICMR, pages 643–650, 2015
5. Masita, K. L., Hasan, A. N., and Satyakama, P., 2018. Pedestrian identification by mean of R-CNN object detector. IEEE Latin American Conference on Computational Intelligence (Nov. 2018). DOI=10.1109/LA-CCI.2018.8625210.
6. R. Girshick, "Fast R-CNN," in Proc. IEEE International Confrence Computer Vision, Dec. 2015, pp. 1440–1448.
7. S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with area proposal networks," in Proc. Adv. Neural Inf. Process. Syst., 2015, pp. 91–99.
8. Liu, H., Chen, Z., Li, Z., and Hu, W. 2018. An powerful method of pedestrian detection based on YOLOv2. Mathematical Engineering Issues(Dec. 2018). DOI=https://doi.org/10.1155/2018/3518959.
9. P. Viola and M. Jones, "Fast object detection using an enhanced cascade of simple features," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Kauai, HI, USA, 2001, pp. I-I.
10. X.Zhu and D.Ramanan. Face detection, pose estimation,and landmark position in the wild. In IEEE CVPR, pages 2879–2886, 2012.
11. Howard, Andrew & Zhu, Menglong & Chen, Bo & Kalenichenko, Dmitry & Wang, Weijun & Weyand, Tobias & Andreetto, Marco & Adam, Hartwig. (2017). MobileNets: Effective Convolutional Neural Networks for Mobile Vision Applications.
12. Zhu, X. 2006. Semi-supervised study of learning literature survey. Computer Science, University of Wisconsin-Madison 2(3):4.
13. David Oro ; Carles Fernández ; Javier Rodríguez Saeta ; Xavier Martorell ; Javier Hernando. Real-time GPU-based face detection in HD video streams. IEEE Int. Conf. Comp Vis 2011
14. Glass RJ, Glass LM, Beyeler WE, Min HJ. Targeted social distancing architecture for pandemic influenza. Emerging Infectious Diseases. 2006;12:1671– 1681.
15. S. Ge, J. Li, Q. Ye and Z. Luo, "Detection of Masked Faces in the Wild with LLE-CNNs," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 426-434.