Social Distancing using Computer Vision

Aanuoluwapo Otitoola[1] Saiteja Pendyala[2] Shahzaad Karim Nawaz[3] Viraj Karalay[4]

[1]Department of Computer Science, University of Liverpool, United Kingdom
[2]Department of Computer Science, University of Liverpool, United Kingdom
[3]Department of Computer Science, University of Liverpool, United Kingdom
[4]Department of Computer Science, University of Liverpool, United Kingdom

a.a.otitoola@liverpool.ac.uk, s.pendyala@liverpool.ac.uk, s.k.nawaz@liverpool.ac.uk, v.karalay@liverpool.ac.uk

Abstract -

The current global COVID-19 pandemic has caused social and economic disruption. Social distancing is by far the only remedy to flatten the curve of the rise in cases and it will be difficult if citizens do not take action to prevent the spread of the virus. This paper focuses on the review of the existing object detection techniques that could help in monitoring social distancing in a pandemic. The dataset used is the MS COCO (Microsoft - Common Objects in Context) and the algorithms used are SSD (Single-Shot Multibox Detector)and YOLO (You only look once). We have compared the performance of the algorithms based on the results obtained from testing these algorithms on the dataset as it relates to social distancing. If we can correctly map bounding boxes around people, we can estimate and detect if a violation has occurred. Additionally, we have reviewed ways to implement the same in CCTV cameras that we see installed in the city. As such, we came up with a unique approach that uses a different type of camera (coupled with a Raspberry-Pi microcontroller) that can replace existing CCTV cameras. Integrating this algorithm with the new camera can serve as a viable technique in ensuring social distancing rules are followed.

Keywords -

COVID-19, Computer Vision, Object Detection, Social Distancing, Deep Learning, Cameras

1 Introduction

The COVID-19 pandemic has impacted human lives and the global economy [1]. COVID-19 also known as Coronavirus is a family of viruses that infect animals and human beings. COVID-19 is one of the viruses that belongs to this family and has spread rapidly. Many people have lost their lives after contracting the virus. A major way we can contain the spread is to self isolate as much as possible and to maintain social distancing in public spaces. At least a 1 meter distance between each person is advised to avoid physical contact. Figure 1-2 shows the social distancing measures being taken in public places, with cross markings on the seats so that people do not sit next to each other [2]. Some countries like the UK, Brazil and the US now have a new variant of COVID-19 which is also infecting a lot of people. Since, social distancing is an important aspect to contain the spread as recommended by the World Health Organisation (WHO). In this literature review, we decided to discuss how we can use computer vision to detect people who are violating social distancing norms.

Figure 1. Social Distancing in a public place [3]

Computer vision often known as CV is a field of study used for developing techniques that help computers to understand contents inside digital images and videos. We have used Object Detection in detecting the objects (person in our case) and Object Classification to categorise the object present inside the image or a video. For our review, we need to classify objects as “Person” [4]. We have used the MS COCO dataset for this. Detecting a person is done using an accurate deep learning algorithm and then calculating the euclidean distance between coordinates of positions where human bodies are detected. If the distance between the points is less than the threshold value then this will give an indication that there is a violation [1]. After that, this can be applied on CCTV cameras to help in detecting people violating the social distancing norms [2].

2 Methodology

2.1 Dataset (MS COCO)

The COCO dataset contains 80 classes (we are interested only in one class - person class containing 66808 images).

The dataset is a list of "metadata," "licences," "images," "annotations," "categories," and "segment info" that is formatted in JSON [5]. The "info" portion of the dataset contains high-level information.You can fill in whatever information you want if you're making your own dataset. A list of image licences that apply to images in the dataset can be found in the "licences" section. The complete list of images in the dataset can be found in the "images" section. This section contains no marks, bounding boxes, or segmentations; it is simply a list of images with details about each one. The process of labelling or classifying an image using text, annotation tools, or both to display the data features you want your model to recognise on its own is described in the "annotations" section. You're adding metadata to a dataset when you annotate an image. [6]. The "categories" object includes a set of categories (for example, dog, boat), each of which is a subcategory of a supercategory (e.g. animal, vehicle).

2.2 YOLOv4

We have observed experiments using YOLOv4 on the MS COCO dataset. To achieve these results, Bochkovskiy et al.[2020] have used features such as Weighted-Residual-Connections (WRC), Cross-Stage-Partial-Connections (CSP), Cross mini-Batch Normalization (CmBN), Self-adversarial-training (SAT), Mish activation, Mosaic data augmentation, DropBlock regularization and CIoU loss. These features were combined to achieve revolutionary results and can be applied to a greater number of datasets, models and object detection tasks. Until recently, CNN-based detectors were mostly used for recommender systems. Yolo v4 not only detects objects in real-time but also with minimal cost, requiring only one conventional Graphics Processing Unit (GPU). With Yolo v4 anyone can use a 1080Ti or 2080Ti GPU to train a super fast and accurate object detector. Previous state of the art methods have been modified and made more efficient for single GPU training.

It is noteworthy to mention that YOLOv4 is based on the Darknet and has obtained an AP (Average Precision) value of 43.5 percent on the COCO dataset along with a real-time speed of 65 FPS (Frames per second) on the Tesla V100, beating the fastest and most accurate detectors in terms of both speed and accuracy. When compared with YOLO v3, the AP and FPS have increased by 10 percent and 12 percent, respectively. Figure 3 shows that EfficientDet D4-D3 achieves better AP than YOLO v4 models. On the other hand we can see that YOLOv4 can run up to (> 120 FPS) but at a lower accuracy.

2.3 General Architecture of an Object Detector

To better appreciate the design of YOLOv4, let's take a look at the components of a one-stage object detector (Figure 3).

- **Input:** Usually images, patches or an image pyramid.
- **Backbone:**
  The backbone is pre-trained on ImageNet. Depending on the platform on which the detector is running, models such as ResNet [8], MobileNet [9, 10, 11, 12], DenseNet [13], VGG [14], CSPDarknet53 [15] could be used as feature extractors.
- **Neck:**
  Object detectors now include some layers called the neck between the backbone and head. They are focused on extracting feature maps of different stages of the backbone. The neck can be composed of Feature Pyramid Network (FPN) [16], Path Aggregation Network (PAN) [17], BiFPN [18], or NAS-FPN [19].
- **Head:**
  The head is the network the handles the actual prediction of classes and bounding boxes of objects.

2.4 Yolov4 Architecture

- **Backbone:**
  CSPDarknet53 was chosen for feature extraction as it performs better compared to CSPResNeXt50 in terms of object detection on the MS COCO dataset. According to Bochkovskiy et al. [2020], an additional module, Spatial Pyramid Pooling (SPP) was added as "it significantly increases the receptive field, separates out the most significant context features and causes almost no reduction of the network operation speed." [7]
- **Neck:**
  A modified Path Aggregation Network (PaNet) was used instead of the FPN used in YOLOv3 as the method of parameter aggregation. An additional module, Spatial Pyramid Pooling (SPP)
- **Head:**
  The YOLOv3 anchor based head was used.

2.4.1 Bag of freebies and Bag of specials:

The authors of YOLO v4 paper described the methods that were used to improve the object detector's accuracy.

- Bag of Freebies (BoF) used for backbone include CutMix and Mosaic data augmentation, DropBlock regularization, Class label smoothing.

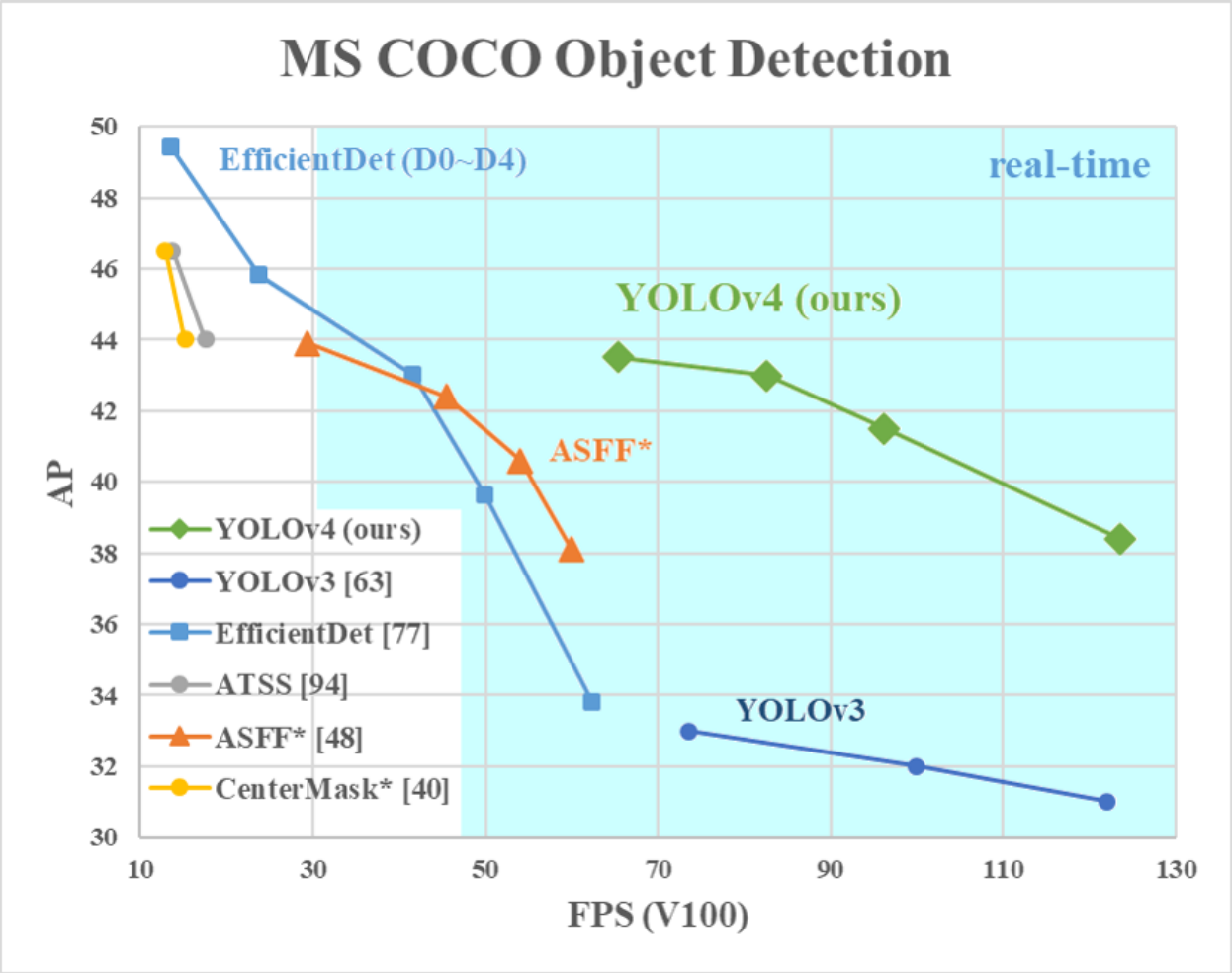

Figure 2. Comparison of the proposed YOLOv4 and other state-of-the-art object detectors. [7]

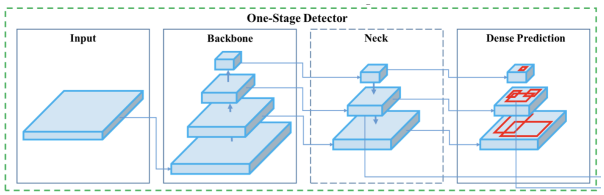

Figure 3. One-Stage Object detector [7]

- Bag of Specials (BoS) used for backbone include Mish activation, Cross-stage partial connections (CSP), Multiinput weighted residual connections (MiWRC)
- Bag of Freebies (BoF) used for detector contains CIoU-loss, CmBN, DropBlock regularization, Mosaic data augmentation, Self-Adversarial Training, Eliminate grid sensitivity, Using multiple anchors for a single ground truth, Cosine annealing scheduler, Optimal hyperparameters, Random training shapes
- Bag of Specials (BoS) for detector: Mish activation, SPP-block, SAM-block, PAN path-aggregation block, DIoU-NMS

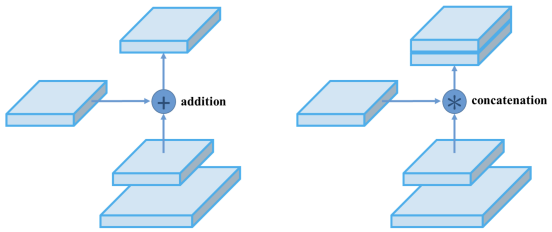

Figure 4. PAN Modification [17]

2.4.2 Advantages of YOLO v4

YOLOv4 uses a Feature Pyramid Network (FPN) for predicting each layer in its structure. You can use YOLOv4

to detect objects using a single GPU. When compared to previous real-time detectors it is very fast and accurate largely due to the fact that it uses a unified model where detection is seen as a single regression problem. YOLOv4 is less error-prone than Fast R-CNN as it can see the bigger context (i.e. a global picture) encoding some of the information into classes in real-time. YOLOv4 is known to successfully differentiate natural images from artworks.

2.4.3 Limitations

YOLOv4 struggles to identify small objects that appear in groups e.g. flock of birds. We may have spatial restriction on bounding boxes where each cell maps two boxes to a single class. Objects of unusual or new aspect ratios and configurations may be difficult to generalize. Also, loss function might treat the errors of small or large bounding boxes as same.

2.5 SSD - Single Shot Detector

With the help of Single shot detector approach, it is possible to predict the is predict the category scores and box offsets for a set of bounding boxes using small convolutional filters applied to feature maps and each feature map is used to create a default box with a different scale and ratio. It calculates the coordinates and class values through the model and then uses them to create the final bounding box [20].

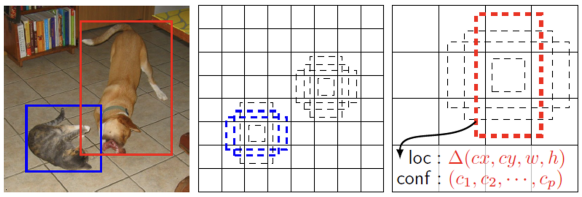

Figure 5. The SSD Framework showing input images and ground truth boxes [20]

2.5.1 Model

Single Shot Detector (SSD) uses a unified two-part network,(fig 7) the base network leveraging a pre-trained VGG16 network on ImageNet, truncated before the last classification layer to extract high level features, then converting FC6 and FC7 to convolutional layers. Next SSD extends this base network by adding auxiliary structure of convolution feature layers of progressively decreasing resolutions $Conv8_2, Conv9_2, Conv10_2 and Conv11_2$ [20].

2.6 Reason for choosing YOLO v4 over SSD

Single shot detectors have a very complex design so that the images are more precise and regions-based detector can accelerate the operation. The detector Yolo single shot, for example, associated features of other detectors, the specific difference may not be the basic concept but the details of implementation [20].

Yolo is more accurate compared to SSD interms of accuracy even though SSD is faster. SSD faces a lot of difficulty when detecting people in an outdoor surveillance. While Yolo is well suited as it more accurate and can hardly miss a person [22].

2.7 Distance

Yolov4 or SSD algorithms are used to generate bounding boxes on people after detecting them. People are detected in a particular Region of Interest (ROI). We must first estimate ROI and this is done with the help of masking.

Masking is an image enhancement method where a small image piece is defined and used to modify a larger image.Masking involves setting some of the pixel values in an image to zero and some other background value as in equation (1).

$$fmask = f - mask(f) \tag{1}$$

By masking the ROI region, it will extract the objects,For example, a video is a series of images which are repeated in a precise sequence over an amount of time. To build ROI for each frame of the input frame, the OpenCV masking method will be used in this study.Each defected ground point is used to compare the ROI range when deciding the location of a person's bounding box as well as the segment involved.Surveillance cameras are typically installed at high places, such as with the overhead camera, to track a specific location, such as a high-risk area or an organization's areas of interest. In this case, it is more suitable to compare the ground plane for the detection box instead of using the center point value.

Calculate the center point of a bounding box:

To measure the center point, C(x,y), of the bounding box for the detected person, the midpoint equation is used as in (2).

$$C(x, y) = (xmin + xmax/2, ymin + ymax/2) \tag{2}$$

The centre point of the bounding box will be calculated using the minimum and maximum values for the corresponding distance, xmin and xmax height, ymin and, ymax of the bounding box.

Calculate distance between bounding box:

To measure the distance, c1(xmin ,ymin) and c1(xmax ,ymax), between each of the detected persons in the frame,

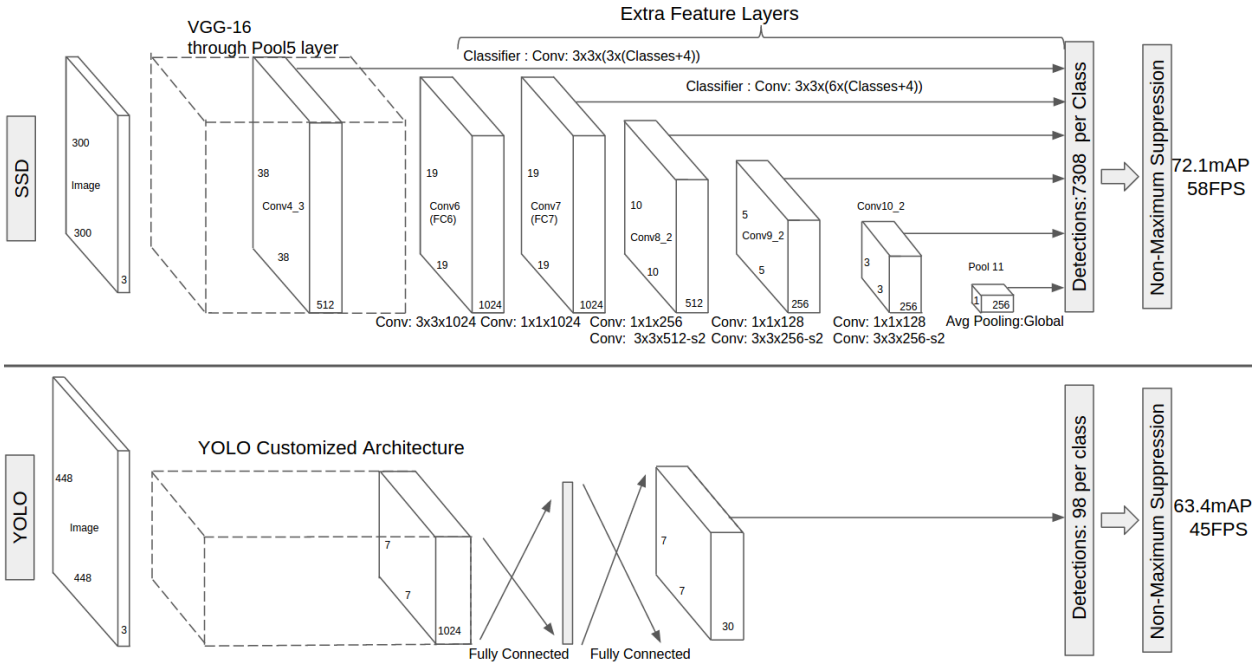

Figure 6. A comparison between two single shot detection models: SSD and YOLO [21]

the distance equation is used as in (3).

$$d(C1, C2) = \sqrt{(xmax - xmin)^2 + (ymax - ymin)^2} \quad (3)$$

The centre point of the bounding boxes is used in this analysis to distinguish between two separate bounding box positions. After obtaining the value of the centre point, the algorithm determines if the distance is less than or greater than 300 pixels.

2.7.1 Masked ROI

The masked ROI consists of two parts. The foreground of the frame is the first masked location. ROI for the region where individual detection would result in a return of the desired ground plane point value is shown in the foreground. The background mask is created by inverting the foreground mask.

Figure 7. Open CV masking step to get foreground frame [23]

2.7.2 Person Detection Localization and Social Distancing

The human detection algorithm in this study is divided into two sections. The first section is configured to compare the distance between each detected person's computed centre point and the computed centre point of the bounding boxes. The distance would be compared to the standard set of social distances. Depending on the frame data, the default pixels for social distance shift. Since each video input has a different camera perspective view, the default minimum distance for social distance varies.

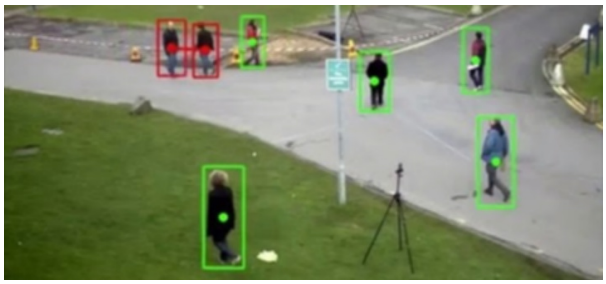

Figure 8. Bounding boxes are green when social distancing is not violated and red when violated [23]

When the safe social distance is breached, the affected persons bounding boxes turn red, meaning that the distance is less than the minimum value for a safe distance. The difference between those who breach the safe distance and those who do not can thus be seen.The accuracy of this method is measured to determine its efficiency. The values for true positive (TP), true negative (TN), false positive (FP), and false negative (FN) for social distance monitoring are counted when measuring the accuracy.The

formula used in calculating the accuracy is shown in (4).

$$Accuracy = TP + TN/TP + TN + FP + FN \quad (4)$$

In this study, the second part of the person detection algorithm entails locating detected people who have entered a prohibited or dangerous location. The ground plane of the bounding boxes for the identified person is determined using the coordinates of the person bounding boxes obtained by the object detection algorithm. To determine whether a bounding box is within the ROI, the ground plane value for the bounding boxes will be compared to the four points of the ROI.When the bounding boxes are inside the ROI, the people are in the forbidden or hazardous place, as shown in Fig. (9). The output in figure demonstrates the same method, with a mark above each bounding box indicating the section in which it is located. When an individual is detected outside the ROI, however, the device does not show and returns the red bounding boxes for the production, indicating that no infringement has occurred or that no people are detected inside the restricted or dangerous area. The output frame for the condition is shown in Figure (9).

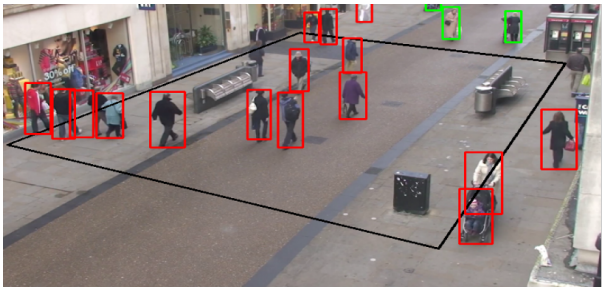

Figure 9. Red bounding boxes for people detected outside the region of interest

3 Applying it on CCTV

In order to integrate this entire project on a CCTV camera a new product must be made where a Raspberrypi microcontroller should be installed. We can use a Raspberry Pi 3 Model B (but we can use any version on Raspberry pi but to connect to this microcontroller wirelessly pi3 is better since it has an inbuilt WIFI module) [24].

To set up the Raspberry Pi we must first download the Raspberry Pi OS which has a Linux based operating system designed for pi microcontroller and install or write it onto the SD card. After that, it needs to be attached to the pi controller [25]. This OS has an inbuilt python environment where we can run the python script along with the python program needed for social distancing which we can be used for this project [26].

Connecting to Raspberry pi remotely We must also be able to connect to raspberry pi remotely after setting it up the VNC server inside the Raspberry OS the first time after manually connecting it to laptop or a pc with an Ethernet or an HDMI cable. Then we need to enable VNC server on Raspberry pi and then connect laptop or any device using VNC viewer and connect it using IP address [27]. However this will only work if the device is connected to the same network or if it is connected through a cloud network.

To connect to the camera remotely (Camera server), a general web camera or a Raspberry pi camera for doing any kind of surveillance and also to apply into this project. To be able to see the live feed remotely we need to download and install "motion" software inside the Raspberry pi OS and change some settings based on the requirement. After that connect to camera with the help of Raspberry Pi's IP address along with its port number 8081 [28] or through VLC player [29] and also ensure that this address or URL is present in the python script as well [30]. But this URL or link will work only if the device and raspberry pi are connected to the same network. If we want to be able to connect to the outside network then a fixed or a global IP or we need a DNS service [29].

4 Conclusion

In trying to reduce the spread of COVID-19, social distancing has remained one of the most important safety precautions that we can take. If social distancing guidelines are not followed, we risk a high rate of virus transmission. This paper seeks to compare the efficiency of two object detection models in identifying human activities and detecting when social distancing policies are violated.

We have reviewed the YOLO v4 object detection model and the Single Shot Detector SSD model. Both models have been reviewed for accuracy and speed using the MS COCO dataset.

Based on our findings, YOLOv4 performs better than SSD. SSD on the other hand is faster but falls short in accuracy. Having also compared YOLOv4 to the CNN based models, we found out that the Faster R-CNN is fast but because it uses CNN, the FPS is slow compared to YOLOv4. We have also expressed the possibilities of monitoring social distancing by implementing models in CCTV cameras using Raspberry Pi.

Results show that YOLOv4 performs well on the MS COCO dataset but it is not perfect. It struggles to identify small objects that appear in groups and objects of unusual aspect ratios and configurations may be difficult to generalize.

To further improve performance and processing speed, we can use more GPUs and train the model on more data.

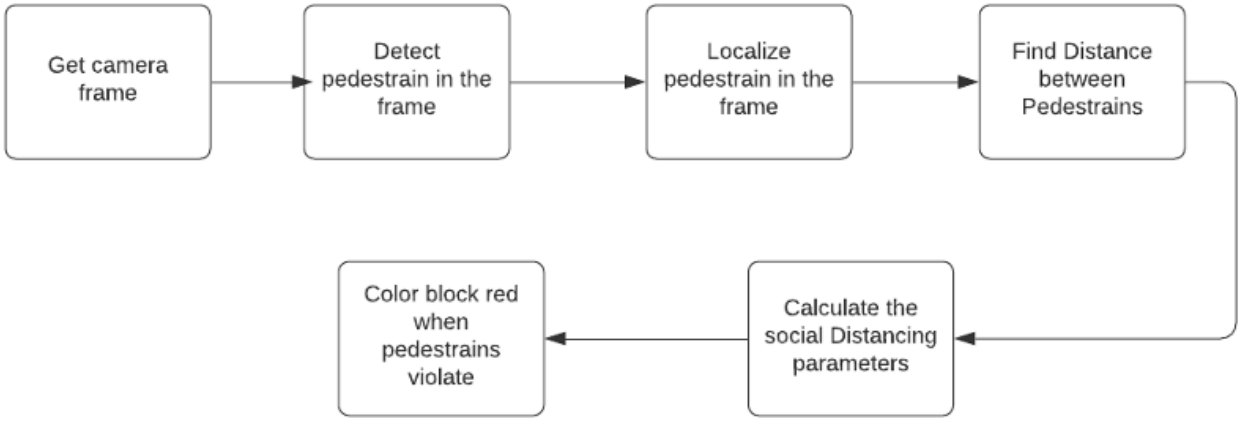

Figure 10. Proposed Flow Diagram

5 Future Scope

An extended study and review may be necessary to add facial recognition technology so that we can identify people violating the social distancing norms. Also, the MS COCO dataset has a lot of classes and since the project needs to identify objects of class "Person" only, a review with a "Pedestrian" dataset for example may be necessary.

6 Acknowledgements

We would like to extend our acknowledgement to the Department of Computer Science, University of Liverpool, UK for imparting the knowledge to be able to write this. We also thank those who have contributed directly to this literature review. Any opinions, findings, conclusions or recommendations presented in this material are those of the authors and do not necessarily reflect the views of the University of Liverpool.

References

[1] A. Ghorai, S. Gawde, and D. Kalbande. Digital solution for enforcing social distancing. In *Proceedings of the International Conference on Innovative Computing Communications (ICICC)*, pages 1–2, Mumbai, India, 2020.

[2] A. H. Ahamad, N. Zaini, and M. F. A Latip. Person detection for social distancing and safety violation alert based on segmented roi. *2020 10th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, 1:113–118, 2020. doi:10.1109/ICCSCE50387.2020.9204934.

[3] D. Kaplan. How Pilota Uses Artificial Intelligence To Assess Air Travel Risk. `https://www.kambr.com/articles/how-pilota-uses-artificial-intelligence-to-assess-air-travel-risk`, 27/07/2020.

[4] J. Brownlee. A Gentle Introduction to Computer Vision. `https://machinelearningmastery.com/what-is-computer-vision`, 01/04/2019.

[5] Immersive Limit. Create COCO annotations from scratch. `https://www.immersivelimit.com/tutorials/create-coco-annotations-from-scratch`, 10/01/2019.

[6] A. Scalabrino. Image Annotation for Computer Vision. `https://www.immersivelimit.com/tutorials/create-coco-annotations-from-scratch`, 10/01/2019.

[7] A. Bochkovskiy, C. Wang, and H. M. Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 1, 2020.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[9] A. G. Howard, M. Zhu, B. Chen, D Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 1, 2017.

[10] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen. Searching for mobilenet v3. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

[11] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen. Inverted residuals and linear bottlenecks.

In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4510–4520, 2018.

[12] M. Tan, B. Chen, R. Pang, A. Vasudevan, M. Sandler, A. Howard, and Q. V. Le. Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2820–2828, 2019.

[13] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4700–4708, 2017.

[14] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 1, 2014.

[15] C. Wang, H. M. Liao, Y. Wu, P. Chen, J. Hsieh, and I. Yeh. Cspnet: A new backbone that can enhance learning capability of cnn. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPR Workshop)*, 2020.

[16] T. Lin, P. Doll´ar, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2117–2125, 2017.

[17] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8759–8768, 2018.

[18] M. Tan, R. Pang, and Q. V. Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[19] G. Ghiasi, T. Lin, and Q. V. Le. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7036–7045, 2019.

[20] W. Liu, D. Anguelov, C. Erhan, D. Aand Szegedy, S. Reed, C. Fu, and Bergm A. C. Ssd: Single shot multibox detector. *Lecture Notes in Computer Science*, pages 21–37, 2016. doi:10.1007/978-3-319-46448-0_2. URL `http://dx.doi.org/10.1007/978-3-319-46448-0_2`.

[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. *arXiv:1506.02640*, 1, 2016.

[22] S. Sanchez, H. J. Romero, and A. D. Morales. A review: Comparison of performance metrics of pretrained models for object detection using the tensorflow framewor. In *IOP Conference Series: Materials Science and Engineering*, pages 10–11, 2020. doi:10.1088/1757-899X/844/1/012024.

[23] B. Roth. Open CV masking to get a foreground frame. On-line: `https://towardsdatascience.com/a-social-distancing-detector-using-a-tensorflow-object-detection-model-python-and-opencv-4450a431238`, Accessed: 17/06/2020.

[24] D. Nate. Build a Raspberry Pi CCTV Camera Network. `https://www.techradar.com/uk/how-to/build-a-raspberry-pi-cctv-camera-network`, 21/12/2016.

[25] Raspberry Pi Foundation. Installing Operating System Images. `https://www.raspberrypi.org/documentation/installation/installing-images/`, Accessed: 12/03/2021.

[26] Real Python. Build Physical Projects With Python on the Raspberry Pi. `https://realpython.com/python-raspberry-pi`, 20/11/2020.

[27] Raspberry Pi Foundation. VNC (Virtual Network Computing). `https://www.raspberrypi.org/documentation/remote-access/vnc`.

[28] Being Engineers. How to make raspberry pi webcam server and stream live video. `https://www.instructables.com/How-to-Make-Raspberry-Pi-Webcam-Server-and-Stream-/`, 21/09/2017.

[29] Tutorials-RaspberryPi. How to Setup a Raspberry Pi Security Camera Livestream. `https://tutorials-raspberrypi.com/raspberry-pi-security-camera-livestream-setup`.

[30] Adventurous Armadillo. How to connect ip Camera to open CV with Python. `https://www.codegrepper.com/code-examples/python/python+ip+camera`, 12/05/2020.