Application to Monitor Customer Flow in Shops during COVID-19 Pandemic

Shane Daly1, Pawel Zamorski1, Lorcan Cooke1, Enda Fallan1, Mamoona Naveed Asghar2  
Faculty of Engineering and Informatics1

Software Research Institute2  
Athlone Institute of Technology  
Athlone, Ireland  
 [A00208135@student.ait.ie](mailto:A00277933@student.ait.ie),  [A00279693@student.ait.ie](mailto:A00283105@student.ait.ie),  [A00279694@student.ait.ie](mailto:A00280155@student.ait.ie), [efallan@ait.ie](mailto:efallan@ait.ie), masghar@ait.ie

**Abstract**

# ***While vaccines are currently being rolled out worldwide for COVID-19, the use of preventative measures such as mask wearing and social distancing of at least two meters in public areas are still essential for the safety of populations. An application is designed in this paper which will allow users to monitor the flow of customers into a shop by detecting people as objects, counting the number of people, tracking the safe distance between them to maintain the two-meter distance required for social distancing. The proposed solution is set up to generate an alarm when the customers reach the allowed limit as per shop dimensions or if there is a crowd. For implementing the solution, YOLOv4 and YOLOv3-Tiny were used and tested for the task of object detection and transfer learning is used to set up weights. The models were evaluated using MSCOCO API with 100 image instances per class. The results of the YOLO v4 model are also compared with YOLO v3-Tiny in terms of mean in terms of average precision (AP), frames per second (FPS) and identification of groups. The application utilized several video clips from a shopping centre CCTV. Experimental results show that the YOLOv3-Tiny maintains real-time performance, including the time to fetch images from the video and display the detections, even on a such weak machine. Unfortunately, YOLOv4 requires better GPU to keep real-time detection. However, the overall detection of objects and cluster identification is much more accurate and clearer using the YOLOv4 model as shown in the evaluation section.***

# INTRODUCTION

The Coronaviruses are a variety of viruses that cause sickness differing from the common cold to more severe diseases. A novel coronavirus is defined as a “..new strain that has not been previously identified in humans” [1]. The new virus known as “COVID-19”, as of the 4th week in 2021, has caused 103 448 210 cases of COVID-19 worldwide with 2,236,453 deaths [2]. Specifically, in Ireland so far there has been 199,430 cases and 3,512 deaths [3]. From these numbers it can be shown that this novel coronavirus has been found in nearly every country in the world and while there are a now a number of vaccines being rolled out worldwide, the overall availability of each is still relatively low and the use of preventative measures such as mask wearing and social distancing of at least two metres in public areas especially where crowds can gather, such as shops, it remains uncertain when things will return to normal, as such panic buying, and crowded shops continue to be an issue and right now and people need to be as cautious as ever. There is a need to control the customers flow in the shopping areas as per the dimensions of the shops.

This paper proposed an application which will allow users to monitor the flow of customers in the shop as well as entering and exiting the shop. The proposed application will detect people as objects. It will count the customers in the shop. An alarm will generate when the customers reach to the allowed limit as per shop dimensions. The proposed app will also track the distance between them and find the clusters of people to maintain the two metre distance required for social distancing. It will generate an alarm for notifying about the crowds.

The proposed application was implemented using Python. YOLOv4 and YOLOv3-Tiny were used for the detection task. Video processing and manipulation was done using OpenCV library. The program was executed on Nvidia GTX 1650 Max-Q GPU accelerated by CUDA.

YOLOv4 and YOLOv3-Tiny were tested, particularly performance of ‘person’ class detection, using precision and recall metrics. Both models were evaluated on MSCOCO test dataset.

The computation performance is evaluated by FPS metric. Two algorithms for identifying clusters of people were tested. Proposed application used CAVIAR dataset [4] that contains many videos from shopping centre.

The main contribution of this research paper are highlighted below:

1. Real-time object/human detection with deep learning models.
2. Two state-of-art object detection models are compared for finding their effectiveness.
3. Tracking the objects for maintaining 2 meters distance.
4. Identification of people clusters/crowd and generate an alarm for information.

This paper provides a fundamental prototype for monitoring the people flow in the covered areas i.e. shopping malls, shops, banks etc.

# LITERATURE REVIEW

This section discussed the recent related work on object detection models in the subsequent sub-section.

## Object Detection

Punn et al. describe how the current COVID 19 situation worldwide could benefit from the use of object detection to reduce the spread of the virus and how this can be achieved with Deepsort techniques. This paper examines and compares the performance of popular object detection and tracking techniques in relation to social distancing and proposes the best software to use going forward. The authors propose a deep learning framework for utilizing artificial intelligence to automate monitoring of social distancing using closed-circuit television (CCTV) and surveillance videos. Ultimately it is shown that in order to detect an object in motion, the two stages of object detection and object classification must be combined. From the analysis taken place by the were they examined a few models including Faster RCNN, SSD and YOLOv3, and determined that YOLOv3 gave the “best results with a balanced mAP and FPS score in order to monitor social distancing in real time”[5].

## YOLO

Adarsh et al. show how deep learning algorithms are developing exponentially with an enhanced object detection performance. In their paper the authors present an general revoew of object detection methods and include two classes of object detectors. The authors describe two stage detector covered algorithms including RCNN, Fast RCNN, and Faster RCNN, and also give examples of one stage detectors such as YOLO v1, v2, v3, and SSD. One stage detectors concentrate on speed while two stage detectors focus on accuracy [6].

Radmond et al introduced YOLO, which acronym stands for You Only Look Once. The two main ideas were to divide an image into grid cells that contain bounding-boxes, and to use single-stage regression to bounding-boxes instead of sliding window or region proposal techniques, that are used i.e., in family of RCNN object detectors. A single convolutional network predicts multiple bounding boxes (localization) while at the same time the class probabilities for those boxes (classification) in a single step. Regression is carried on two variables: spatially separated bounding boxes and associated class probabilities. This approach significantly increased a speed of object detector, but with trade of accuracy [7].

Next versions of YOLO used improved convolutional networks. YOLOv2 was based on Darknet-19 that contains 19 convolutional layers and 5 maxpooling layers [8].

YOLOv3 uses Darknet-53 as its backbone. Darknet-53 has 53 layers and uses residual blocks. It anticipates bounding boxes at 3 different scales for improvement of small object detection. The multilevel detection layers are added after convolutional layers. It uses upsampling technique. The unsampled layers are concatenated with the previous layers from convolution part. The 13 x 13 layer is accountable for detecting large objects, whereas the 26 x 26 layer detects [9]. Building a deeper CNN with improved accuracy with the drawback of a reduction in speed.

Adding residual blocks is crucial for models containing many convolutional layers. The authors of residual blocks, came to the conclusion that by simply adding more convolutional layers to a model it leads to a higher number of errors. The main reason is vanishing the gradient and degradation problem.

The vanishing gradient problem is fixed by utilizing normalized initialization and intermediate normalization layers (batch normalization). This ensures that networks with tens of layers start converging for stochastic gradient descent (SGD) with backpropagation.

Degradation problem occurs when the network depth to grow in size, and starts the saturation of accuracy which then degrades rapidly. This problem is solved by a residual block, which is a shortcut connection that skips one or more layers. It performs identity mapping, and the outputs are collated to the outputs of the stacked layers. Identity mapping just takes the outputs from earlier layers and passes it down[10].

## YOLO version 4

The paper announcing YOLOv4 was released in 2020. During the time between developing the new version and the previous one, many new detection models were developed, and many new techniques were used to improve detectors.

The goal of developing YOLOv4 was to create a real-time object detector with high accuracy that operates and may be trained on conventional GPU, like 1080 Ti or 2080 Ti.

YOLOv4 uses architecture of his predecessor. The authors of new version concentrated on developing the model by optimizing components within YOLO. They tested many variants. They divided different methods into two categories:

* Bag-of-Freebies – methods used during training of a model for improving the overall accuracy of an object detection model.
* Bag-of-Specials – plugins modules and post-processing methods that can significantly improve the accuracy of object detection but increase the inference cost only by a small amount.

YOLOv4 uses among others:

* Bag of Freebies for backbone or detector: CutMix and Mosaic data augmentation, Self-Adversarial Training (SAT), Random training shapes, DropBlock regularization, Complete IoU loss function (CIoU-loss), Cross mini-Batch Normalization (CmBN).
* Bag of Specials for backbone or detector: Mish activation, Cross Stage Partial Connections (CSP), Multiinput Weighted Residual Connections (WRC).

They also tested various backbone options. The tests proved that CSPDarknet53 is less accurate on classification task than CSPResNeXt-50, however it is better on detection tasks. Therefore, the CSPDarknet53 was chosen [11].

## YOLO version 3 Tiny

J. Redmon proposed a simplified version of YOLOv3, called YOLOv3-Tiny [12]. It is a stripped verion of Darknet-53. It uses 9 convolutional layers and 6 max pooling layers. It predicts bounding boxes on 2 different scales [13]. The aim of building this model was to improve the speed of object detection and to build model that can run on weak GPU.

## Video Streaming

Closed Circuit Television (CCTV) technology has made a massive impact in how the world of retail works, that its almost baffling to imagine a world without it. However, most CCTV systems are still flawed, only providing footage, they only capture activity without understanding what is actually happening. With regards to this app, CCTV technologies can be expanded further in what they can monitor in a relatively simple way.

"The main challenge lies in maximizing the use of an existing CCTV camera with a fixed viewpoint, which is tailored for security purposes instead of video analytics, by using its footage in the IVA… The final system gathered data on foot traffic, customer gender classification, and customer group size. Neural networks such as YOLO, Deep SORT, and InceptionV3 were employed in the implementation. The results show that while it is possible to gather data on these three metrics through the system, the speed and accuracy can still be improved through downsizing the frames, down sampling the videos, and using other algorithms."

-Excerpt from opening paragraph of ‘Visual-based People Counting and Profiling System for Use in Retail Data Analytics’. [14] This quote stood out as being equally relevant and appropriate for our proposed idea.

* 1. Video Streaming Applications

Hamami and peers [15] gave themselves four missions in their aim to monitor traffic using CCTV. We can aim for something similar though we only really need three of them. The names speak for themselves.

1. Real-Time Counting
2. Multiple-Object Sensing
3. Object Classification
4. Anomaly Detection

‘Deep Learning’ is the latest technology for understanding images. It mimics how the human brain functions and can recognize the object and understand the situation. Hamami et all used Deep Learning, and CNN combined with YOLO to achieve their goals.

Hamami et all, also claim that traditional databases (like SQL) are not able to store traffic data from CCTV as, among other issues, the volume of data is too large, so data is stored using MongoDB [15].

# MODEL EVALUATION

YOLOv4 and YOLOv3-Tiny were evaluated using MSCOCO (Microsoft Common Objects in Context) API with 100 image instances per class. This dataset contains 80 object categories.

## Metrics

Metrics for object detection include evaluation of classification and detection tasks. The classification task only evaluates the probability of the class object appearing in the image. The object detection task localizes the object with a bounding box associated with its corresponding confidence score. The confidence score represents probability that the bounding box detected an object. The most popular method for evaluation confidence score is Intersection over Union (IoU). It is the ratio of the overlapping area of ground truth and predicted bounding box to the total area. A prediction is positive if IoU is above given threshold. The authors of YOLO used 0.5 as a threshold to compare metrics to other object detectors, as they consider it as a good prediction [9].

The predictions are classified into True Positives (TP), False Positives (FP), False Negatives (FN), and True Negative (TN). The predicted object is True Positive if the predicted class is correct and the overlap with the ground truth is greater than a threshold. The predicted object is marked False Positive if it has overlap less than that threshold or same object instance is detected again, or it is misclassified, or a background is predicted as an object instance. False Negative is when the object exists, but a model does not predict it. True Negative occurs when empty boxes are correctly detected as non-object. Detection model predicts thousands of True Negative, making it not useful for object detection metrics. True Positives, False Positives and False Negatives are used to construct two metrics: precision and recall.

Precision, also referred to as the positive predictive value, is the probability of the predicted bounding boxes being correctly predicted. Precision scores range from 0 to 1.The higher the precision score means that more of the detected objects match the ground truth objects.

Recall is the true positive rate, also referred to as sensitivity. It measures the probability of detecting all relevant objects. Recall ranges from 0 to 1 where a higher recall score means that more of the ground truth objects were detected.

## Model Evaluation Results

Transfer learning was used to set up weights of both models, YOLOv4 and YOLOv3-Tiny. These weights were taken from the GitHub repository of official release of YOLOv4 and YOLOv3-Tiny.

The scores of YOLOv4 are as follows:

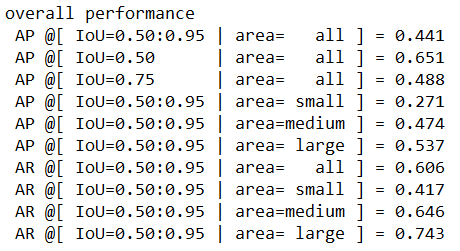


Figure 1 Overall performance of YOLOv4 tested on MSCOCO dataset

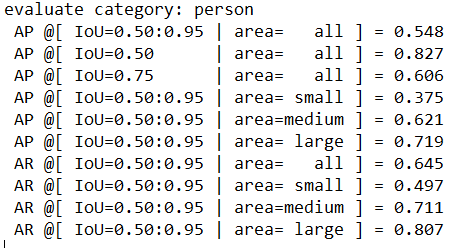


Figure 2 Performance of YOLOv4 for class ‘person’ evaluated on MSCOCO dataset

The Average Precision for IoU 0.5 (AP50) and class ‘person’ is 0.827. The AP50 for class ‘person’ above 0.8 is impressive. It is better than overall performance of YOLOv4, which is 0.651 in this test category. A high precision means that most detected objects match ground truth objects and are classified correctly. In the case of class ‘person’, 0.827 of all detected objects were True Positive and 0.173 were False Positive.

The Average Recall (AR) of class ‘person’ is slightly better than overall performance in all average recall test categories. A high recall score means that most ground truth objects were detected. For class ‘person’ it is 0.645 for all object sizes, meaning that 0.645 of all existing ‘person’ objects were detected whereas 0.355 were not detected (False Negative).

The model performs very well on large and medium objects and drops on small objects. It applies both to overall performance and to class ‘person’.

The results of evaluating YOLOv3-Tiny are as follows:

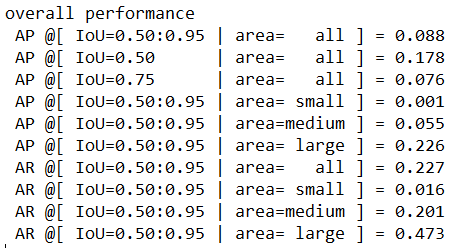


Figure 3: Overall performance of YOLOv3-Tiny tested on MSCOCO dataset

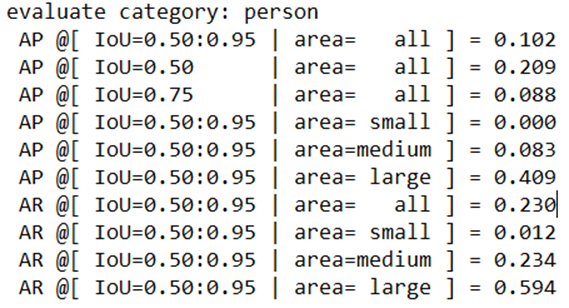
****

Figure 4: Performance of YOLO3-Tiny for class ‘person’ evaluated on MSCOCO dataset

YOLOv3-Tiny performs acceptable only on large objects. The AP for large objects of class ‘person’ is 0.409, while the AR is 0.594. It means that almost 0.6 existing large ‘person’ objects were detected, but only 0.4 of all objects detected as ‘person’ were classified correctly. The score is worst for overall category for large objects.

The medium and especially small objects were almost not detected, or the detection was incorrect.

This test shows great limitation of YOLOv3-Tiny. It may be used only to detect large objects on images.

# APPLICATION ARCHITECTURE

The proposed application counts the customers to track the number of people in the shop to maintain the two meters distance required for social distancing. As shown in figure below, the model is set up to generate an alarm when the customers reach to the allowed limit as per shop dimensions. The designed application also finds clusters of people in the shop and generate an alarm for notifying about the crowds.

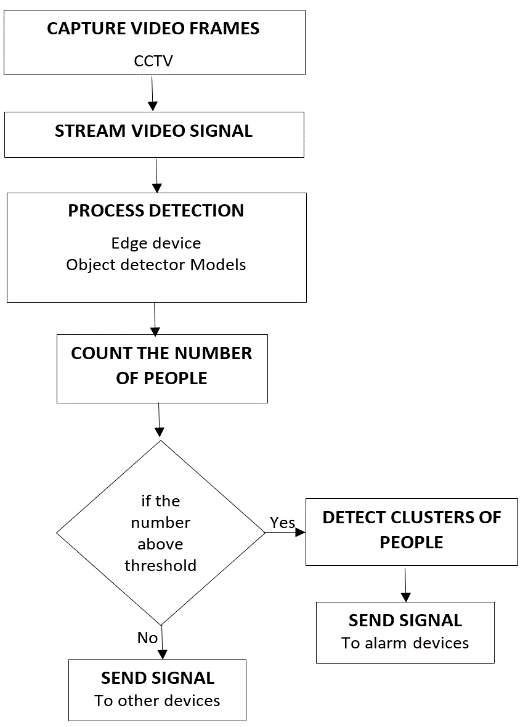


Figure 5: Process Flow diagram of proposed application.

The application was implemented using Python. Both YOLOv4 and YOLOv3-Tiny were used and tested for the detection task with application. Video processing, image extraction, and image manipulation was done using OpenCV library.

The algorithm for detecting clusters of people was proposed and tested. The algorithm uses bounding boxes predicted by object detector to track the 2 meters horizontal and vertical distance from objects.

# IMPLEMENTATION RESULTS AND ANALYSES

## Running Scenario

The program was run on Caviar dataset to detect people in retail store [4]. This dataset contains several video clips from shopping center CCTV, both inside and outside. These include among others the following scenarios: people walking alone and in group, meeting with others, entering and exiting shops.

## Algorithm for tracking the safe distance

The algorithm uses bounding boxes and 2D image property taken by a pinhole camera to track the 2 meters horizontal and vertical distance from objects.

CCTV cameras are usually located at a certain angle above the area. This ensures a distant area to be covered. However, there is a perspective distortion on images. Objects located near camera are bigger than objects located further from camera. Similarly, the corridor looks bigger on the image when it is near the camera. It is why a straight corridor appears to converge to a point despite remaining the same width. It causes a problem in calculating a distance between objects from a 2D image.

The special information of objects in real-world is required to determine the distance between them. Mapping the 3D real-world into a 2D image causes a loss of information. There are expensive 3D cameras that preserve special information. However, they are not popular to use because of their cost. Another possible solution is to restore special information by transforming a 2D image into a 3D model. It is the low-cost option. However, it requires knowledge of extrinsic and intrinsic parameters or usage of a series (2 or more) of 2D images captured from different angles [16]. Another approach could include transformation of an image into a bird's-eye view. At first step a single point representing an object should be found on a surface, i.e., on a floor. Next this surface should be transformed into bird's-eye view. After such transformation, the distance between objects could be calculated. The pitfall of such approach is that information outside the transformed area is lost. For this dataset information would be lost about objects located between pillars.

## We use a different approach. It was observed that bounding boxes detected by YOLOv4 tightly surround objects found in the image. These objects are standing on the floor – on the same surface. The flat surface of the full corridor is visible. Such scenario, a visible flat surface, is very common in case of videos taken by CCTV.

## Centre points at the bottom of bounding boxes are used to calculate if objects are in safe or close distance from each other. Those points were called ‘object position’. The objects are then in close distance if two conditions are true:

## absolute difference of x-coordinates of two object positions are below a x-threshold

## absolut difference of y-coordinates of two object positions are below a y-threshold.



Figure 6 Visualization of ‘object position’, y-threshold and x-threshold

The distance between objects is based on a rectangle, where the centre point of the rectangle is ‘object position’, and x-threshold and y-threshold are half of its sides.

The ‘object position’ is used to track the safe distance from other objects. The correct values for x and y thresholds are calculated based on ‘object position’. The values of those thresholds differ depending on the location of object within an image. They decrease for objects located further from the camera, which implies that they are dependent on y-coordinate of an ‘object position’.

In real-world the distance between objects is calculated based on Euclidean distance. However, it was observed that x-threshold and y-threshold differ from each other, whereas they represent the same distance in real-world. For that reason, the Euclidean distance could not be used.

## Calculating x-threshold and y-threshold

It is assumed that the video was taken from CCTV camera that preserves straight lines. For such camera, i.e., a pinhole camera or a camera with rectilinear lens, straight lines from real world are mapped to straight lines on image. Otherwise, such images need to be calibrated at the first step. Another pinhole camera property is that parallel lines from real world may not stay parallel on image but intersect in vanishing point that is in or outside an image. It depends on the angle view of camera to those lines. The size of the object is inversely proportional to the distance. The thresholds were calculated based on y-coordinate of an object in an image. As x-threshold and y-threshold we used an equivalent of 2.5 meters in pixel (2 meters obligatory distance + 0.25 \* 2. 0.25 is the average distance from the centre of a person the right or left side). The x and y coordinates were taken from image.

The real-world distance between points were taken from Caviar datasets that include information about location of points in real-world, measured in cm. See the figure and table below.

|  |  |  |
| --- | --- | --- |
| Figure 7: Image from Caviar dataset representing points for which the real-world locations are known [4] | | |
| **Point** | **(Col, Row) (pixels)** | **(X, Y) (cm)** |
| 1 | (91,163) | (000, 975) |
| 2 | (241,163) | (290, 975) |
| 3 | (98,266) | (000, -110) |
| 4 | (322,265) | (290, -110) |

Table Points coordinates on image in pixel and in real-world in centimeter [4]

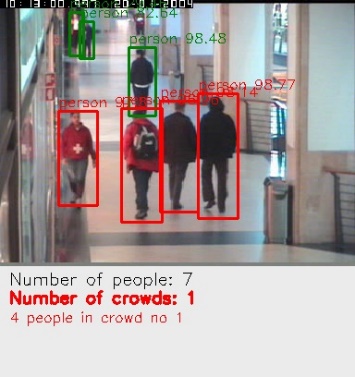
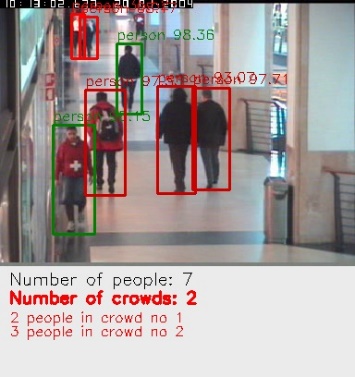
The data from the above table were used to calculate the real distance between points. The real distance between point 1 and 2 is 290 cm. The real distance between point 1 and 3 is 1085 cm. More characteristic points were also extracted from the image.

|  |
| --- |
| Figure 8: Coordinates taken from characteristic points from image used to calculate x and y threshold. |

Based on those points and the real distance in meters between those points, the linear functions for y-threshold and x-threshold were calculated. The below figures contain results, where x is y-coordinate in pixel, and y is y-threshold in pixel or x-threshold in pixel. The y-coordinate was taken using OpenCV, where y axis is inverted, meaning that top left point at image is (0,0), and bottom right point at image is (xmax, ymax).

|  |  |
| --- | --- |
| Figure 9: Y-Threshold on image in pixel graph | Figure 10: X-Threshold on image in pixel graph |

The above method enables crowd detection. Below are sample results.

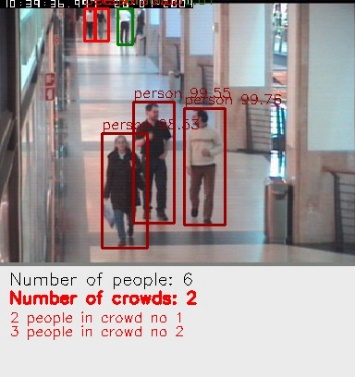
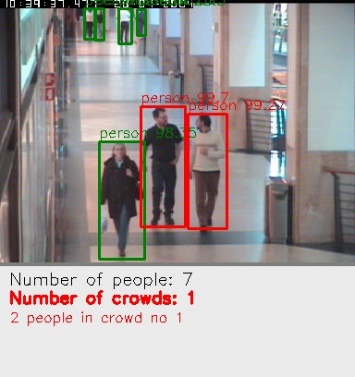
 

Figure 11 Correctly highlighted a group of people, both in vertical and horizontal line (application used YOLOv4 for detection task).

The application correctly highlighted clusters of people consisting of large and medium size objects. It drops on identifying crowds in far distance from camera.

## Filtering Information about Crowds

We also improved filtering of information about detected crowds. We used the following steps:

*Step 1.* Filter objects class ‘people’.

*Step 2.* Generate all possible combinations of 2 elements from the set of ‘people’ elements with size n.

*Step 3*. Filter combinations that their ‘people’ elements are in close distance.

*Step 4.* Merge sets from step 3 into sets that represents a group of ‘people’. The object belongs to the group of ‘people’, if it is in close distance with at least one element from that group.

As a result, we extracted information about number of crowds and number of ‘people’ in each crowd.

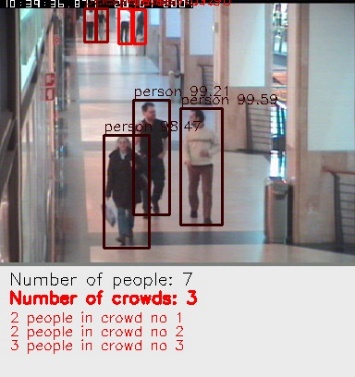
 

Figure 12 “Number of Crowds” and number of people in each crowd (application used YOLOv4 for detection task).

As shown above, the application is able to identify accurately different clusters of people.

This illustrates that the application could work and that YOLO object detector is a software which can be used to achieve its goal.

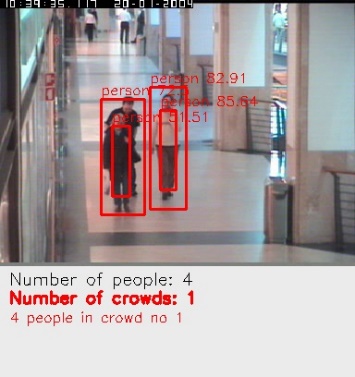
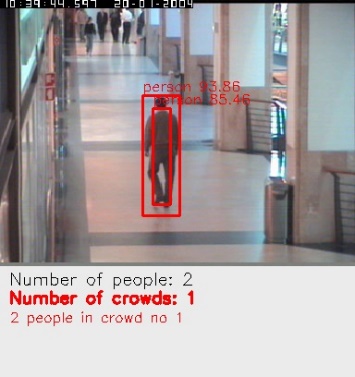
## Comparative evaluation of models

Both object detectors were tested with proposed application. For the evaluation, the program was run over a laptop with Nvidia GTX 1650 Max-Q GPU. GPU was accelerated by CUDA.

The experiments were performed on CAVIAR datasets with videos. The videos were taken form a wide-angle lens along the hallway in a shopping center. The resolution is 384 x 288 pixels, which is half-resolution PAL standard. Refresh rate is 25 frames per second. Both YOLOv4 and YOLOv3-Tine were set up to use 416x416 image as an input.

YOLOv3-Tiny was able to perform a real-time detection, including the time to fetch images from the video and display the detections. Unfortunately, YOLOv4 requires better GPU to keep real-time detection. It was able to process 4 frames per second on the above machine. YOLOv4 is able to run in real-time video on Pascal GPU [11], i.e., GTX 1080. Such video card is much stronger than GTX 1650 Max-Q.

However, in comparison of the output, YOLOv3-Tiny was found to be inefficient for object detection task as shown in below figures. It does not detect objects in far distance from camera, which appear small on image. It predicts incorrect bounding boxes for medium size objects. In case of large objects, it often predicts more than one bounding boxes for the same object.

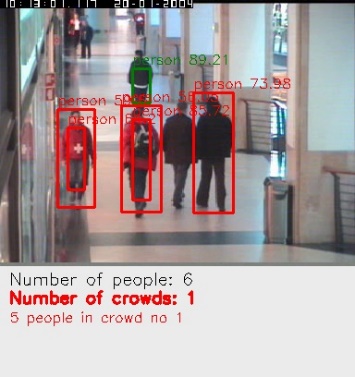
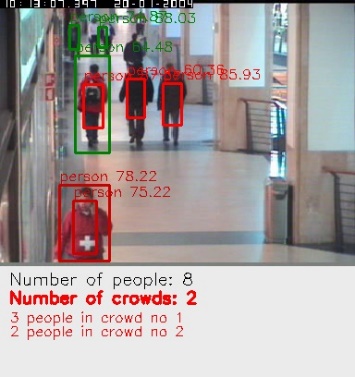
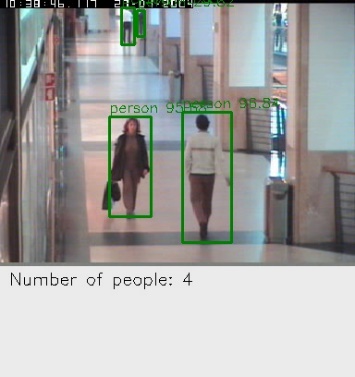
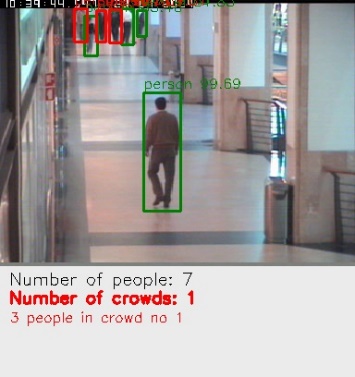
 

Figure 13 Incorrect detection results after running application using YOLOv3-Tiny.

Using YOLOv3-Tiny with proposed program confirmed results from evaluation this model on MSCOCO dataset. It does not detect small objects and struggles to detect medium objects. The test also revealed other issue of YOLOv3-Tiny. It does not predict bounding boxes correctly.

As could be seen from results after using YOLOv3-Tiny, incorrectly predicted, and not tightly surrounded bounding boxes lead to the misleading results. For example, it detects two bounding boxes around single person, which causes application to highlight a crowd.

Below are the same frames but results are after using YOLOv4 for object detection task.

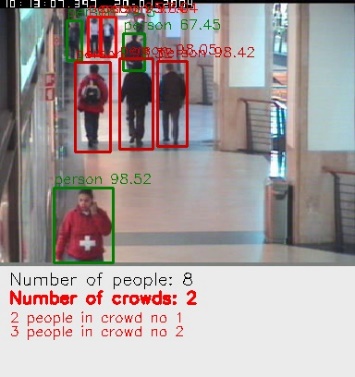
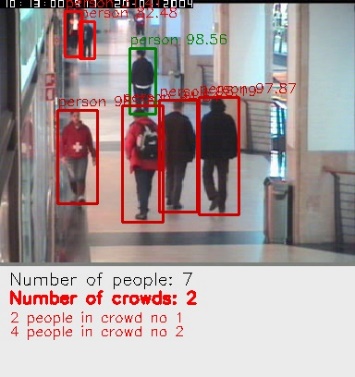


Figure 14 Improved result after using YOLOv4 object detector. YOLOv4 detects large, medium, and small objects of class ‘person’. It detects also a partially visible object.

Counting people is performed very well. YOLOv4 detects not only large and medium objects, but also small objects. Small objects are far from camera and, in our opinion, are hard to detect by human. YOLOv4 also detects correctly if a small part of a person body is visible. The bounding boxes are surrounded tightly around detected object. This aspect is crucial for the proposed application, as it enables to correctly track the distance between objects and highlight crowds.

# Conclusion

This paper describes the most recent studies related to the problem of flow of customers in shops during COVID-19 pandemic. A real-time deep learning framework required in order to automate the process of detecting each person as an object and a group of people. The proposed application provides a solution to identify the clusters of people which are within the defined social distance of 2 metres and provide an alarm or sound device to instruct the people to move further aware from each other for safety reasons. The authors are using YOLOv4 due to it being the most recent version and its speed and accuracy for object detection. YOLOv4 was found to have very good results on the detection of people that allows to track the distance between them, as shown in the results section. This is compared to the YOLOv3-Tiny which although faster in terms of processing time but proved to be inefficient in terms of object detection. It may be used on the Internet of Things (IoT) devices but should be used very carefully and only for task that require detection of large objects.

In short, this paper provides an essential solution for monitoring the people flow in the covered areas for public.

In the future, more tests are planned by using these models to determine the most effective use of YOLOv4 to provide the optimized solution for controlling customers flow during pandemic.

# References

|  |  |
| --- | --- |
| [1] | World Health Organisation, "COVID 19 - Questions and Answers," World Health Organisation, [Online]. Available: http://www.emro.who.int/health-topics/corona-virus/questions-and-answers.html. [Accessed 4th Februray 2021]. |
| [2] | European Centre for Disease Prevention and Control, "COVID-19 situation update worldwide, as of week 4, updated 4 February 2021," [Online]. Available: https://www.ecdc.europa.eu/en/geographical-distribution-2019-ncov-cases. [Accessed 4th February 2021]. |
| [3] | RTE.ie, "RTE.ie. 2021. Coronavirus in Ireland - a timeline," [Online]. Available: https://www.rte.ie/news/2020/0320/1124382-covid-19-ireland-timeline. [Accessed 4th February 2021]. |
| [4] | "CAVIAR: Context Aware Vision using Image-based Active Recognition," [Online]. Available: https://homepages.inf.ed.ac.uk/rbf/CAVIAR/. [Accessed 15 02 2021]. |
| [5] | N. S. Punn, S. K. Sonbhadra and S. Agarwal, "Monitoring covid-19 social distancing with person detection and tracking via fine-tuned yolo v3 and deepsort techniques," arXiv preprint, 2020. |
| [6] | P. Adarsh, P. Rathi and M. Kumar, "YOLO v3-Tiny: Object Detection and Recognition using," in *6th International Conference on Advanced Computing & Communication Systems* , 2020. |
| [7] | J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection.," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. pp. 779-788, 2016. |
| [8] | J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. pp. 6517-6525, 2017. |
| [9] | J. Redmon, A. Farhadi, "YOLOv3: An Incremental Improvement," 2018. |
| [10] | K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition.," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. pp. 770-778, 2016. |
| [11] | A. Bochkovskiy, C. Wang, H. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection.," arXiv:2004.10934, 2020. |
| [12] | J. Redmon, "YOLO: Real-Time Object Detection," [Online]. Available: https://pjreddie.com/darknet/yolo/. [Accessed 10 02 2021]. |
| [13] | X. Bi, Z. Chen and J. Yue, "A Novel One-step Method Based on YOLOv3-tiny for Fatigue Driving Detection," *IEEE 6th International Conference on Computer and Communications (ICCC),* no. Chengdu, China, pp. 1241-1245, 2020. |
| [14] | M. D. Cruz, J. J. U. Keh, R. G. Deticio, C. V. T. Tan, J. A. C. Jose, E. Sybingco, E. P. Dadios, "Visual-based People Counting and Profiling System for Use in Retail Data Analytics," *IEEE International Conference on Industrial Engineering and Engineering Management (IEEM),* pp. pp. 200-204, 2020. |
| [15] | F. Hamami, S. W. Prakosa, I. A. Dahlan, K. F. Somantri, "Big Data Analytics for Processing Real-time Unstructured Data from CCTV in Traffic Management," *2020 International Conference on Data Science and Its Applications (ICoDSA),* pp. pp. 1-5, 2020. |
| [16] | A. Yuniarti, N. Suciati and A. Z. Arifin, "Image Classification Performance Evaluation for 3D Model Reconstruction," *International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET),* no. Tangerang, Indonesia, pp. 229-233, 2020. |
| [17] | N. S. Punn and S. Agarwal, "Crowd Analysis for Congestion Control Early Warning System on Foot Over Bridge," “Indian Institute of Information Technology Allahabad, 2019. |
| [18] | K. Masita, A. N. Hasan and T. Shongwe, "Deep Learning in Object Detection: a Review," in *2020 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication System*, 2020. |
| [19] | N. Hutchins, Z. Kirkendoll and D. L. Hook, "Social impacts of Ethical Artifical Intelligence and Autonomous System Design," The Univeristy of Tulsa. |
| [20] | S. Chaudhary, M. A. Khan and C. Bhatnagar, "Multiple Anomalous Activity Detection in Videos," in *6th International Conference on Smart Computing and Communications, ICSCC 2017*, Kurukshetra, India, 2017. |
| [21] | S. Angra and S. Ahuja, "Machine Learning and its Applications: A Review," Chitkara University India. |
| [22] | S. A. Albawi and T. A. Mohammed, "Understanding of a Convolutional Neural Network," in *The International Conference on Engineering and Technology* , Antalya, Turkey, 2017. |
| [23] | R. Fisher, "CAVIAR Test Case Scenarios," 2007. [Online]. Available: https://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/. [Accessed Feburary 2021]. |