A Smart Surveillance Prototype Ensures the Respect of Social Distance During COVID19

Ikram Ben abdel ouahab, Lotfi Elaachak, Fatiha Elouaai, and Mohammed Bouhorma

Laboratory of Computer Science, Systems and Telecommunications (LIST), Faculty of Sciences and Techniques, University Abdelmalek Essaadi, Tangier, Morocco

ikram.benabd@gmail.com, lotfi1002@gmail.com, elouaaif@gmail.com, mbouhorma@gmail.com

Abstract. At the moment, the best and only way to reduce the spread of coronavirus disease 2019 is by limiting close contact with others. Respecting social distance, infection is less probable. Since this is a new lifestyle for everyone, it’s hard to be distance all the times. People forget to keep distance or they are not taking seriously the actual situation. That’s why in this paper, we propose a smart surveillance solution. Our test prototype ensures the respect of social distancing by detecting persons, calculating distances between them and generating loud vocal alerts. The smart surveillance prototype is based on Raspberry Pi and Camera Pi. Then, we make a comparison study of object detection pretrained models. SSD-MobileNet gives the most satisfying result using Raspberry pi with limited computing resources. Despite that implementing CNN based model on the Raspberry Pi is such a challenging work, we reach a value of 1.1 FPS on real-time object detection and distance analysis system.

Keywords: Computer vision · Object detection · Deep learning · COVID19 · Raspberry pi

1 Introduction

The Corona Virus Disease of 2019 or COVID-19 is a disease caused by a new strain of coronavirus. This new virus is linked to the same family of viruses as Severe Acute Respiratory Syndrome (SARS) and some common types of cold. The coronavirus spreads through direct contact with infected persons by coughing or sneezing. Infections can also happen when touching surfaces contaminated with the virus and touching eyes, nose or mouth.

Every day, researchers learn and discover new things about COVID-19. Currently, there is no vaccine available for COVID-19. However, there are some clinical trials that are being conducted to evaluate potential therapeutics for COVID-19. Furthermore, public health measures are critical to slow the spread of the virus [5, 13, 17]. People should respect preventive actions like:
– Social distance, keeping enough space between each other
– if sick, stay at home
– wear mask, cover mouth and nose, don’t touch the face
– hygiene and sanitary practices like washing hands frequently and cleaning surfaces and objects

The concept of social distancing is very important to prevent infection. By standing further away from others, avoiding crowds and not touching. Regrettably, people around the globe are not conscious enough of the seriousness of COVID-19. They are ignoring social distancing protocol, thus helping to further spread of the virus.

In this paper, we propose a solution to urge people to respect the social distancing protocol. It’s a smart surveillance system that generates vocal alerts in the case of disrespecting social distance. We make a test prototype based on the Raspberry pi. And we perform object detection model based on deep learning algorithms to detect persons and calculate distances between them. The proposed prototype can also detect groups of persons not respecting social distance.

Our proposed prototype can be implemented in many places that require the respect of social distance. For instance, a classroom where students should keep distance, administrations, university corridors and common spaces, and others.

The rest of the document is organized as follow. In Sect. 2, we present some related works regarding highly to object detection algorithms in literature. In Sect. 3, we cover the methodology used including hardware specifications and object detection algorithms. Then, Sect. 4 gives the detection process and describes the proposed prototype. After that, in Sect. 5, we give the obtained results. Finally, we make conclusions and we introduce some of the future works.

2 Related Works

Recently many intelligent surveillance systems have been developed; each has its own particularity. Depending on the need, surveillance system could focus more on a point than another. These are some of the similar implementations we get inspired from.

In [1], authors proposed an efficient CNN model able to detect relevant objects in a real-time video surveillance applications. So they exploit a pre-trained model on a large dataset. After that, they fine-tune the network and apply a transfer learning in several others datasets. Testing the resulting model on the Penn-Fudan dataset with a GPU they get a speed of 53 FPS and accuracy of 95%.

In [3], we found the use of a new methodology called multi-object detection and tracking (MODT) using an optimal Kalman filtering technique to track moving objects. Tested on video clips, the accuracy of MODT framework is equal to 76.23% for detection and 86.78 for tracking. However it requires powerful graphics processing units.

In addition, another application of video surveillance concerns traffic flow estimation. Researchers in [4] employed the state-of-the-art Faster R-CNN two-stage detector together with SORT tracker to solve their issue. As a result, their solution was able to count vehicles and driving directions with less than 10% mean average percentage error.

Object detection task is high computation-intensive and energy-consuming, designed to work on advanced GPU architectures. Adapting a deep neural network used for object
detection purpose for embedded devices is a challenging task. However, we found many interesting works proving that is possible to perform average quality object detection on embedded devices, such as [6, 8–11, 15].

Moreover, wide ranges of application use object detection algorithms, in many domains. For example, autonomous driving [14] where detecting objects in the road is very important. Other examples include surveillance, robotics and smart cities. Finally in our case, we use object detection pretrained models for the surveillance of social distancing in the era of COVID-19.

Particularly, our proposed solution implement a CNN based model in an embedded device; Raspberry Pi. Our main goal is to detect persons that ignore social distance by generating vocal alerts in real-time. And the use of Raspberry pi gives us the ability to put the prototype anywhere we want. It shouldn’t necessary be fixed or connected to a computer.

3 Methodology

In this section we cover hardware materials used in the development of the prototype, including Raspberry Pi and Camera Pi. Also, we present deep learning model used for object detection task: YOLO and SSD models. Then, we give techniques used to calculate the social distance.

3.1 Raspberry Pi

The Raspberry Pi is a series of tiny and cheap single board computers developed in the United Kingdom by the Raspberry Pi Foundation [18]. It’s a credit card sized that can use a computer monitor or TV, a standard keyboard and mouse. Particularly, the Raspberry Pi series provides a set of GPIO pins that allows the control of various electronic devices such as sensors. It now is widely used even in research projects, such as weather monitoring, robotics, Smart Home Automation projects and others. Since 2012, 4 generations of the Raspberry Pi have been developed meeting all needs. We give a brief timeline of all version of the Raspberry Pi in Fig. 1.

In our proposed prototype, we use the 3rd generation Raspberry Pi 3 Model B because of its availability during quarantine. Technical specifications of the Raspberry Pi 3 Model B are presented in the following points:

- Quad Core 1.2 GHz Broadcom BCM2837 64bit CPU
- 1 GB RAM
- Bluetooth 4.1 and Wi-Fi
- 40 Pin extended GPIO
- 4 × USB 2.0 Ports
- 10/100 LAN Port
- 3.5 mm 4-pole Composite Video and Audio Output Jack
- CSI Camera Port
- Full size HDMI Output
- Micro USB Power Input 2.5A
Fig. 1. The raspberry pi generations timeline

- DSI Display Port
- MicroSD Card Slot

The official operating system for all the Raspberry Pi versions is provided by the same Foundation and called Raspberry Pi OS (previously Raspbian). We use the latest version available, it is the Raspberry Pi OS based on Debian buster released on May 2020 with a kernel version 4.19. Then to perform the object detection model, we used Python3.7, TensorFlow v2, OpenCV, Numpy and others useful libraries.

3.2 Raspberry Pi Camera

The Raspberry Pi Camera module is an official Raspberry Pi Foundation product. In the proposed prototype we use the Raspberry Pi Camera Rev1.3 (Fig. 2) which is able to deliver a crystal clear 5MP resolution image or 1080p HD video recording at 30FPS. It’s a coin sized camera dedicated to the Raspberry Pi usage via the CSI Camera Port. It can be easily implemented using the “picamera” Module [16] using Python.

Many others versions of the Raspberry Pi Camera are available including the latest high quality camera module. This newly high quality camera have 12.3 megapixel Sony IMX477 sensor, 7.9 mm diagonal image size, and back-illuminated sensor architecture, with adjustable back focus and support for C- and CS-mount lenses. We are looking forward to use it in a future work.
3.3 Object Detection Pretrained Models

Object detection is a hottest topic in the computer vision field. An object detection model is able to detect multiple objects within an image, with bounding boxes. Particularly, in our case we are interested only on the person class. To do so, we test two pretrained object detection models: MobileNet-SSD and YOLOv3.

**MobileNet-SSD.** Single Shot object detection (SSD) was developed by Google researcher teams. As indicated by its name, SSD takes one single shot to recognize multiple objects within the image. SSD is composed of two parts: 1/extracting the feature maps, 2/then applying convolutional layers to detect objects. Two models of SSD are available:

1. **SSD300** where the input image size is fixed to $300 \times 300$. This model is useful for low resolution images and it offers a faster processing speed.
2. **SSD512** where the input image size is fixed to $500 \times 500$. This model is dedicated to higher resolution images and it has a higher accuracy than the first one.

In our application, we used a Caffe implementation of Google MobileNet SSD model [2], with pretrained weights on the PASCAL Visual Object Classes database (VOC0712). The MobileNet SSD model was initially trained on the COCO dataset then it was fine-tunes on VOC0712. The mean Average Precision (mAP) of the model is about 72.7%. Because we’ll use the model on an embedded device, we use the SSD300 version that requires less computing resources.

**YOLOv3.** You Only Look One is an object detection approach proposed in [12]. It is based on a deep Convolutional Neural Network architecture. The main advantage of YOLO is the high performance and its high speed. Also, it is an open source real-time detection system [19] dedicated to work on GPU. For instance, using a NVIDIA TITAN X Pascal GPU YOLO detection model processes images at 30 FPS (Frame/Second) and has a mAP of 57.9% on COCO test-dev, which is better than real-time. Many versions
of the YOLO had been developed regarding highly to the ability of detection a wide variety of objects correctly and rapidly. Latest improved version is YOLOv3. Using miniaturized embedded devices, the conventional YOLOv3 algorithm runs slowly.

In our application, as we are using the Raspberry Pi, we test the YOLOv3-tiny network. The tiny version of YOLOv3 is dedicated to work on limited hardware resource and can basically satisfied real-time requirements.

### 3.4 Social Distance

After using the object detection model, we have the detected persons’ coordinates. These coordinates are used to display bounding boxes exactly where the person is. In our case, we use these coordinates to compute firstly the midpoint of the box then the distance between each two persons. To do so, we adopt the Euclidean distance give in the formula below (Eq. 1).

\[
\text{distance}(A, B) = \sqrt{\sum_{i=1}^{n} (A_i - B_i)^2}
\] (1)

For a given box with coordinates: \(P1(x_{\text{center}}, y_{\text{center}})\), where \(P1\) is the detected person 1 and \((x_{\text{center}}, y_{\text{center}})\) are the coordinates of the center of the box. When detecting for example two persons we have: \(P1(x_{\text{center}}, y_{\text{center}})\) and \(P2(x_{\text{center}}, y_{\text{center}})\). Now we apply the Euclidean distance formula to get distance between them, as you can see in Fig. 3.
4 Proposed Prototype

In this paper, we propose a smart surveillance box that detect persons, calculate distancing between them and generate vocal alert if the social distance is not respected. The proposed prototype can be placed anywhere, for example in a classroom, university corridors, companies, administrations, supermarkets or any other place.

The main element in the prototype is the Raspberry Pi and Camera Pi. The process of detection is given in Fig. 4. The camera Pi is connected directly to the raspberry pi using dedicated flexible flat cable. Capturing frames on a live stream, the camera Pi send captured frames to the SSD model to detect object then calculate distances under the Raspberry Pi. If the social distance is not respected between detected persons, the Raspberry Pi generates vocal alerts using a mini speaker. This speaker could be connected to the Raspberry either using Bluetooth or a classic AUX audio cable. The prototype box components and a real picture of our prototype are presented in Fig. 5.

![Detection process](image)

**Fig. 4.** Detection process

![Prototype components](image)

**Fig. 5.** Prototype components

Our smart surveillance box is able to make two different types of alerts depending on the type of detection. First case is when just two persons are not respecting the social distance. Here a small alert is generated and the speaker says: “Stay away please”. Second case is when more than three persons are detected not respecting the social distance. Here, we say that a cluster of persons is detected then a big alert is generated saying: “Cluster detected, stay away!”. As described in Fig. 6.
5 Results and Discussion

In order to have the most efficient system, we test SSD and YOLO models in Raspberry Pi. In Table 1, we present the given results with some variation and we compare out results to others found in literature. And to evaluate the model we use 2 metrics: FPS and mAP. Frame per Second or FPS is the frequency rate at which consecutive images appear on the screen. Mean Average Precision or mAP is a well-known metric used to measure accuracy of object detection models.

Starting with YOLO, we implement the YOLOv3-tiny pretrained model using the Raspberry Pi and also using a Laptop with CPU. Using same deep learning model, where input size is $416 \times 416$ in both cases. The results are so different. FPS using i7 processor is 10 times better than Raspberry Pi. Moreover, in [12] using a GPU the performance is much higher as intended to be, reaching a value of 35 FPS. So as a conclusion, the actual YOLOv3-tiny model is very slow in the Raspberry Pi. And we can’t use it for a real-time detection application.
Table 1. Comparing variations of YOLO and SSD object detection models performances

| Technical specifications | Metrics | Refs |
|--------------------------|---------|------|
| **YOLO** | | |
| Hardware: Raspberry Pi 3 Model B. Processor: Quad Core 1.2 GHz Broadcom BCM2837 64bit CPU, RAM: 1 GB | Model: YOLOv3-tiny | Input size: 416 × 416 | FPS ≈ 0.4 | ours |
| Hardware: Processor: Intel(R) Core(TM) i7-7500U CPU @ 2.70 GHz 2.90 GHz RAM: 8 GB | Model: YOLOv3-tiny | Input size: 416 × 416 | FPS ≈ 4 | ours |
| Hardware: NVIDIA TITAN X Pascal GPU | Model: YOLOv3 | Input size: 416 × 416 | FPS ≈ 35 mAP = 55.3% | [12] |
| **SSD** | | |
| Hardware: Raspberry Pi 3 Model B. Processor: Quad Core 1.2 GHz Broadcom BCM2837 64bit CPU, RAM: 1 GB | Model: SSD MobileNet (on COCO and VOC2012) | Input size: 300 × 300 | FPS ≈ 1, 1 | ours |
| Hardware: Processor: Intel Xeon E5-2667v3@3.20 GHz, with Titan X and cuDNN v4 | Model: SSD MobileNet (on VOC2007, batch_size = 8) | Input size: 300 × 300 | FPS ≈ 59 mAP = 74.3% | [7] |

Moving to the SSD-MobileNet pretrained model, it uses an input size of 300 × 300 using Caffe framework. This implementation gives an accurate result where the FPS is about 1.1 during real-time object detection and social distance calculations in the Raspberry Pi. We emphasize that we are using the Raspberry Pi 3 model B which is very limited in term of processing and memory RAM (1 GB). However, in [7] using efficient GPU researchers found a value of 59 FPS. Which is very normal because of the processing power and energy consumption of the CNN based model.

All things considered, we summaries all aspects of the proposed prototype in Table 2, including technical details and results.
Table 2. Technical summary of the proposed prototype

| Materials                  | - Raspberry Pi 3  
|                           | - Camera Pi Rev.1.3  
|                           | - Speaker  
| Object detection model    | - SSD MobileNet  
|                           | - Image input size $= 300 \times 300$  
|                           | - FPS $\approx 1, 1$  
| Input                     | - Live stream from Camera Pi  
| Analysis process          | - Detect persons  
|                           | - Locate detected persons’  
|                           | - Compute distance between them  
| Output                    | - Vocal alerts in case of disrespect of social distance  
|                           | - Alert type 1: Two persons  
|                           | - Alert type 2: Group of persons  

Otherwise, we test the prototype in real-time. An extract of the demonstration is given in Fig. 7 and 8. As you can see in Fig. 7, two persons are detected and they do not respect the social distance. So automatically, the alert message is launched. Then in Fig. 8, the two persons are distance so there isn’t any alert.

![Fig. 7. Demonstration 1: disrespect of social distance](image_url)
6 Conclusion and Perspectives

To conclude, in this paper we proposed a smart surveillance prototype based on the Raspberry Pi and Advanced AI algorithms for social distancing monitoring during coronavirus disease (COVID19). Our main goal is to warn people using vocal alerts in real-time while they are ignoring social distance protocols. To do so, we make a comparative study of some pretrained object detection models. As a result the SSD-MobileNet is doing a satisfying job. So the final prototype implemented the SSD model to detect persons on real-time, then we calculate the distance between each two detected persons. If two persons are too close, not respecting social distance a vocal alert is generated. Also if a group of persons are not respecting the mentioned distance a big vocal alert is generated.

The most challenging part in this project is to fit the CNN based model to the Raspberry Pi which is limited in term of computing resources. At this time, the result covers our needs. However we are looking forward to improve the object detection model in order to increase the accuracy and speed of the system. In addition, it is recommended to use the latest version of the Raspberry Pi 4 which has a much more powerful processing power than the Raspberry Pi 3. So we suppose that it would improve the system quality and accuracy.

As perspective, we are looking forward to improve our deep learning model taking on consideration 3D situations. For more innovation, other details could be added. For example, facemask detection systems to verify either people are wearing their mask or not. Even more, a thermal camera connected to the raspberry will check people with common symptoms, and many others additions we’re working on to prevent spread of the COVID-19.
Acknowledgments. This project is subsidized by the MENFPESRS and the CNRST as part of the program to support scientific and technological research related to “COVID-19” (2020). Also, we acknowledge financial support for this research from CNRST.

References

1. Ahmadi, M., Ouarda, W., Alimi, A.M.: Efficient and fast objects detection technique for intelligent video surveillance using transfer learning and fine-Tuning. Arab. J. Sci. Eng. 45(3), 1421–1433 (2019). https://doi.org/10.1007/s13369-019-03969-6
2. chuanqi305: chuanqi305/MobileNet-SSD (2020)
3. Elhoseny, M.: Multi-object Detection and Tracking (MODT) machine learning model for real-time video surveillance systems. Circ. Syst. Signal Process. 39(2), 611–630 (2019). https://doi.org/10.1007/s00034-019-01234-7
4. Fedorov, A., et al.: Traffic flow estimation with data from a video surveillance camera. J Big Data 6(1), 73 (2019). https://doi.org/10.1186/s40537-019-0223-z
5. Hellewell, J., et al.: Feasibility of controlling COVID-19 outbreaks by isolation of cases and contacts. Lancet Glob. Health. 8(4), e488–e496 (2020). https://doi.org/10.1016/S2214-109X(20)30074-7
6. Kaymak, C., Ucar, A.: Implementation of object detection and recognition algorithms on a robotic arm platform using raspberry pi. In: 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), pp. 1–8 (2018). https://doi.org/10.1109/IDAP.2018.8620916
7. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C.: SSD: Single Shot MultiBox Detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
8. Mao, H., et al.: Towards real-time object detection on embedded systems. IEEE Trans. Emerg. Top. Comput. 6(3), 417–431 (2018). https://doi.org/10.1109/TETC.2016.2593643
9. Mao, Q.-C., et al.: Mini-YOLOv3: real-time object detector for embedded applications. IEEE Access. 7, 133529–133538 (2019). https://doi.org/10.1109/ACCESS.2019.2941547
10. Mehmoood, F., et al.: Object detection mechanism based on deep learning algorithm using embedded IoT devices for smart home appliances control in CoT. J Ambient Intell. Hum. Comput. (2019). https://doi.org/10.1007/s12652-019-01272-8
11. Oh, S., et al.: Investigation on performance and energy efficiency of CNN-based object detection on embedded device. In: 2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT), pp. 1–4 (2017). https://doi.org/10.1109/CAIPT.2017.8320657
12. Redmon, J., Farhadi, A.: YOLOv3: an incremental improvement. arXiv:1804.02767 [cs]. (2018)
13. Sun, C., Zhai, Z.: The efficacy of social distance and ventilation effectiveness in preventing COVID-19 transmission. Sustain. Cities Soc. 62, 102390 (2020). https://doi.org/10.1016/j.scs.2020.102390
14. Wu, B., et al.: SqueezeDet: unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving. Presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (2017)
15. Zhao, H., et al.: Mixed YOLOv3-LITE: a lightweight real-time object detection method. Sensors 20(7), 1861 (2020). https://doi.org/10.3390/s20071861
16. picamera — Picamera 1.13 Documentation. https://picamera.readthedocs.io/en/release-1.13/
17. Social distance and SARS memory: impact on the public awareness of 2019 novel coronavirus (COVID-19) outbreak. medRxiv. https://www.medrxiv.org/content/10.1101/2020.03.11.20033688v1

18. Teach, Learn, and Make with Raspberry Pi – Raspberry Pi, https://www.raspberrypi.org/

19. YOLO: real-time object detection, https://pjreddie.com/darknet/yolo/