IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Traitement du Signal
Vol. 39, No. 3, June, 2022, pp. 923-929
Journal homepage: http://iieta.org/journals/ts

Embedded Implementation of Social Distancing Detector Based on One Stage Convolutional Neural Network Detector

Yahia Said[1*], Riadh Ayachi[2]

[1] Electrical Engineering Department, College of Engineering, Northern Border University, Arar 1321, Saudi Arabia
[2] Laboratory of Electronics and Microelectronics (LR99ES30), Faculty of Sciences of Monastir, University of Monastir, Monastir 5019, Tunisia

Corresponding Author Email: Yahia.said@nbu.edu.sa

https://doi.org/10.18280/ts.390318

Received: 7 March 2022
Accepted: 16 May 2022

Keywords:
COVID-19 prevention, social distance detection, deep learning, convolutional neural networks (CNNs), embedded implementation

ABSTRACT

The recent COVID-19 is a very dangerous disease that intimidates humanity. It spreads very fast and many rules must be respected to reduce its prevalence. One of the most important rules is the social distance which means keeping a safe distance between two persons. A safe distance must be one meter or more. Respecting such rules in public spaces is a very challenging task that needs the assistance of artificial intelligence tools. In this paper, we propose a social distance detector using convolutional neural networks. The detector was based on the Yolo model with a custom-made backbone to guarantee real-time processing and embedded implementation. The backbone was optimized to make it suitable for embedded resources. The inference model was evaluated on the Pynq platform. The model was trained and fine-tuned using the MS COCO dataset. The evaluation of the proposed model proved its efficiency with a precision of 87.98% while running in real-time. The achieved results proved the efficiency of the proposed model and the proposed optimization for embedded implementation.

1. INTRODUCTION

Recently, the Corona Virus Disease 2019 (COVID-19) is affecting 215 countries and territories around the world. COVID-19 has infected more than 42 million people and killed more than one million persons [1]. The main problem of the coronavirus that it spreads through the air and it is very hard to control its prevalence. Many rules were proposed to reduce the impact of this disease. One of the most important rules was social distancing which means keeping a distance of one meter or more between every two persons. This rule was effective in reducing the infection probability. But most of the people are ignoring this rule and that makes the situation worse. To ensure social distancing protocol in public spaces such as supermarkets and transport stations, an artificial intelligent social distance detection must be involved in the loop to reduce the purser on the human operators.

An artificial intelligence breakthrough was made by the use of the deep learning technique for computer vision tasks. Deep learning [2] is defined using deep artificial neural networks to solve complex tasks such as image processing [3], natural language processing [4], and signal processing [5]. The main power of deep learning comes from the use of deep specific neural networks and the ability to learn features directly from input data without any handcrafted algorithms. Convolutional neural network (CNN) [6] is the most famous deep learning model for image and video processing applications [7, 8]. CNN was inspired by the biological cortex and its decision-making process mimics the biological brain. CNN is composed of different types of layers for different tasks to achieve trusted predictions. The most important is the convolution layers that are used to extract features and the non-linear activation layers that are used to extend the deep and the ability of the network to handle more complex tasks. in second place we find pooling layers that are used to compress the dimension of the feature maps to reduce the network computation complexity at the decision-making stage. Pooling layers can be eliminated and replaced by strided convolution layers [9].

CNN was successfully used to solve many daily live tasks such as indoor navigation [10, 11], scene recognition [12], indoor object recognition [13], traffic signs classification and detection [14, 15], pedestrian detection [16, 17] and many other applications [18, 19]. The main problem of CNN that is computationally expensive and need high-performance computers to achieve the desired performance. Recently, many techniques were proposed to make CNN suitable for low-power devices.

Pruning [20] was one of the most important techniques that allow the implementation of CNN models on embedded devices. The main idea of the pruning technique was to remove redundant parameters that are not relevant to the network by setting its weight to 0. This allows reducing the number of neurons that accelerate the processing time and reduce memory usage. Many types of pruning were proposed such as pruning from scratch and post-training pruning. In this paper, we will focus on pruning the model for inference [21] since the training will be performed on a high-performance computer and the inference will be implemented on a low power device. Our main goal was to reduce memory usage and computation complexity without dramatically damage the accuracy.

Quantization [22] was a very important optimization technique that allows implementing CNN models on

embedded devices. This technique aims to reduce the number of bits used to represent the weights on CNN. Also, it replaces floating-point representation with fixed-point representation. Many works [23] have proved that CNN weight can be represented using an 8-bits integer instead of a 32-bits floating point without incurring a significant loss of accuracy. Also, the use of 4/2/1-bits integers for weight representation was investigated [24] and great progress was achieved.

In this work, we proposed a social distance detector based on CNN model. The CNN model was designed to be implemented on low-power devices to be implemented on public spaces surveillance cameras without the need for high-performance computers. To reach real-time processing, we propose the use of one-stage detectors. This type of detector was developed for speed purposes with accepted accuracy. You only look once (Yolo) [25] was the most powerful one stage detector. As its name means it processes the input image in a single pass and generates predictions. Yolo solved the detection task as a regression task to speed up the processing time. The second version Yolo v2 [26] was used in this work because it presents more performance in the detection task compared to the first version of Yolo. To make the proposed detector suitable for embedded implementation, a custom backbone was proposed based on CNN. The proposed backbone was built without using pooling layers for embedded implementation purposes [9]. CNN models with only convolution layers are more adaptable for implementation on hardware devices such Field Programmable Gate Arrays (FPGA).

The social distance detector is composed of two steps. The first step is to detect pedestrians and the second step is to calculate the distance between every two pedestrians. Based on the calculated distance, we predict if the social distance was respected or not. To validate the performance of the proposed detector, the MSCOCO dataset [27] was used for training and fine-tuning after applying the optimization techniques. To focus on the desired task, we consider only the class person from the dataset and the other classes was used as negative data. The evaluation of the proposed detector proved its efficiency with a precision of 87.98% and a processing speed of 17 FPS.

The remainder of the paper is the following: section 2 presents the related works. The proposed approach was detailed in section 3. In section 4 experiment and results were discussed. The paper was concluded in section 5.

2. RELATED WORKS

Since the first appearance of the COVID-19, many related research topics were investigated. Enforcing safety rules are the most important and many works were proposed to build automatic systems. Talking about social distance detection, it is divided into two main parts. The first part is the person detection which the most important and needs a lot of attention. The second part is the distance calculation between every two persons and interrupts if the social distance is respected. For both parts, many works were proposed.

2.1 Person detection

Person detection can be considered as a separate task for a specific application or can be included with general object detection. Most famous datasets such as Pascal VOC [28] and MS COCO [27] are considering persons as a relevant class.

Faster RCNN [29] was the first object detection model based entirely on CNN mode. It is composed of two jointly CNN models. The first model was used for feature extraction which is based on existing models such as VGG [30] and ResNet [31]. And the second model named region proposal network (RPN) was used for region proposal generation for the detection task. The RPN shares the features extracted by the first model to reduce computation and generate more trusted regions to reduce the processing time and enhance the detection accuracy. The faster RCNN achieve good detection precision, but it was very slow in processing time with 0.2 seconds per image.

In the context of processing speed, the Yolo [25] was proposed. It was designed to be very fast without losing much accuracy. In Yolo, the RPN was eliminated, and a single CNN model was used to perform both tasks, detection, and identification, simultaneously. Both tasks were solved as a regression problem by a single pass through the CNN model. The Yolo model was very fast with a speed of 0.02 seconds per image, but it suffers from low precision and struggles in detecting small objects. More versions of Yolo are then proposed to balance the person and speed. The Yolo v2 comes with a new idea to enhance precision such as using predefined anchors like those used in the RPN [29] and it was effective. The Yolo v3 [32] enhances the precision but it was computationally extensive.

The single shot multi-box detector (SSD) [33] was proposed to achieve a balance between speed and precision. In the SSD, extra layers were added to detect objects at different scales through a pyramid structure. The proposed contribution was very useful and good results were achieved in terms of precision and processing speed. The SSD was computationally extensive even when using lightweight backbones such as MobileNet [34]. Besides, it suffers from class imbalance and the precision for some classes was very low compared to other classes.

RetinaNet [35] was proposed as a solution to the class imbalance problem. It proposes a focal loss function by applying a modulating term to the original loss function to focus on learning hard examples for low precision classes. The RetinaNet is composed of a CNN backbone and two specific outputs. The first output is used for classification and the second one is used for object localization through a linear regression layer. The proposed focal loss function was very effective, and state of the art performance was achieved. The RetinaNet is computationally extensive and must run on a high-performance GPU [36].

The mentioned object detection models were trained on large-scale datasets and can be directly integrated or fine-tuned for social distance detection.

2.2 Social distance detection

Artificial intelligence applications can be very useful for the fight against the COVID-19 by eliminating the human operator from the loop and reducing physical contact. Enforcing social distancing have a great impact on reducing the spreading speed of the virus. Many works were proposed to enforce social distancing and detect non-social distancing behaviors.

Narinder et al. propose a social distance detector based on the Yolo v3 model [32] and Deepsort techniques. The Yolo v3 was used to detect persons by processing images. The

Deepsort techniques were used to assign an ID to each detected person and track it in the video. The Yolo model was fine-tuned for person detection. The model achieved an mAP of 84.6% and a processing speed of 23 FPS when implemented on Nvidia GTX 1060 GPU. The proposed approach needs a high-performance computer to be implemented for real-world use and this is not available for everyone because of its high cost.

An artificial intelligence-based social distance detector was embedded in a drone in the study of Ramadass et al. [37]. The proposed detector was based on the Yolo v3 model [32] for person detection. The Yolo v3 was used to process the images provided by the drone camera and calculate the distance between persons to check whether the social distance is respected or not. Also, the model was used to detect face masks.

The mentioned methods were proposed to detect social distancing but none of them is available for use in public spaces due to its limitation. In the next section, we will present in detail our social distance detector and discuss its availability for implementation in existing surveillance cameras at a low cost.

3. PROPOSED APPROACH

The COVID-19 is causing a world crisis and because of the absence of any cure, social distancing is considered as an important rule that people must respect to reduce the spreading impact of the virus. Since people are ignoring this rule, a warning for all Violators must be performed. To detect Violators, an automatic social distance detector was proposed based on deep learning. The proposed detector was based on the Yolo v2 model, and it was suitable for embedded implementation. The performance was validated using a Pynq board [38].

The main idea of the Yolo models is to divide the input image into an s x s grid. Then for each grid cell, it predicts one object with a fixed number of bounding boxes with a confidence score each and conditional class probabilities for each class. The bounding boxes were randomly guessed. Each predicted bounding box has 5 parameters which are the (x, y) coordinate of the offset of the corresponding cell, the width (w), the height (h), and the confidence score. x, y, w, and h were normalized by the height and width of the input image. So, the value of those parameters is between 0 and 1. To generate the final prediction Yolo uses two linear regression layers to predict bounding box parameters. As a final output only bounding boxes with a confidence score greater than 0.25 were presented. For the classification task, the class confidence score was calculated by multiplying the bounding box confidence score and the conditional class probabilities. The one object prediction proposed in the first version of Yolo limits its detection rate by ignoring close objects. Besides, the initial training was not stable because of randomly guessed bounding boxes.

The Yolo v2 was proposed to enhance the detection precision. In real life, objects of the same class have similar shapes with different sizes. So, Yolo v2 proposed to use a predefined anchor and generate bounding boxes by predicting the offset of those anchors. Thus, if the offset values are fixed then the diversity of the predictions can be maintained, and each prediction will focus on a specific shape. So, the initial training will be more stable and better results can be achieved. For better predictions, the fully connected layers were removed and replaced by a 1x1 convolution layer. Also, the last convolution layer was replaced by three 3x3 convolution layers with a 1024 output channel each. The size of the input image was changed to a multiple of 32. In addition, one pooling layer was removed to generate a 13x13 output grid. The main concept of Yolo v2 is presented in Figure 1.

The Yolo v2 model used a k-mean clustering algorithm by processing the training data and generating the predefined anchors. The original Yolo v2 was trained on the MS COCO dataset and 5 predefined anchors were used. In this work, we run the k-mean clustering algorithm on the person class and new 5 predefined anchors were proposed. Figure 2 presents the difference between the original anchors (old) and the proposed anchors (new). The shape of the new anchors is corresponding to the shape of the person at different positions and sizes. The proposed anchors allow focusing on the person class to achieve better performance for the studied task. For social distance detection, the only person must be detected to measure the distance between them and detects social distancing.

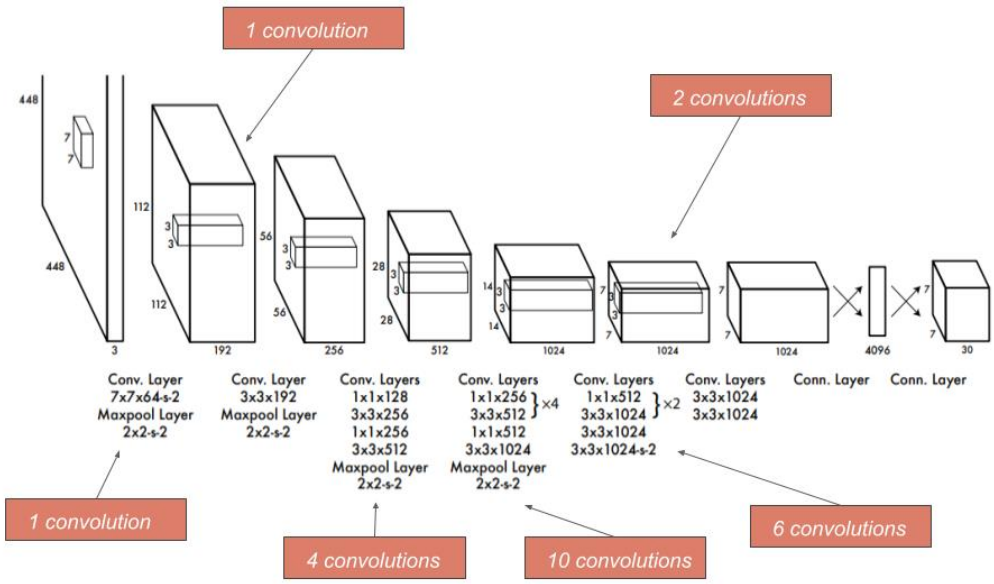

Figure 1. The main concept of Yolo v2

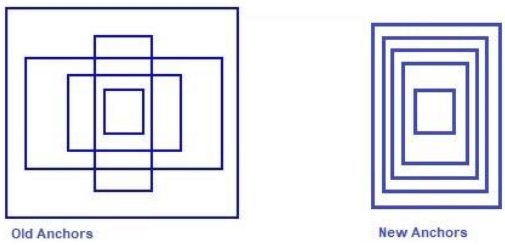

Figure 2. Proposed anchors vs original anchors

To train the model a special loss function was proposed that combines the detection loss and classification loss. The proposed loss can be computed as (1).

$$loss = \lambda_{coord} \sum_{i=1}^{s^2} \sum_{j=1}^{k} q_{ij}^{obj} (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + \lambda_{coord} \sum_{i=1}^{s^2} \sum_{j=1}^{k} q_{ij}^{obj} (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 + \sum_{i=1}^{s^2} \sum_{j=1}^{k} q_{ij}^{obj} (c_i - \hat{c}_i)^2 + \lambda_{noobj} \sum_{i=1}^{s^2} \sum_{j=1}^{k} q_{ij}^{noobj} (c_i - \hat{c}_i)^2 + \sum_{i=1}^{s^2} q_i^{obj} (p_i - \hat{p}_i)^2 \quad (1)$$

q_i^{obj}: the grid cell *i* contains a person.

q_{ij}^{obj}: the *jth* bounding box that contains a person.

q_{ij}^{noobj}: the *jth* bounding box that does not contain a person.

(x_i, y_i): defines the *ith* center coordinate of the ground truth bounding box.

$(\hat{x}_i, \hat{y}_i)$: defines the *ith* center coordinate of the predicted bounding box.

$\hat{w}_i$: the width of the *ith* ground truth bounding box.

w_i : the width of the *ith* predicted bounding box.

$\hat{h}_i$: the height of the *ith* ground truth bounding box.

h_i : the height of the *ith* predicted bounding box.

$\hat{c}_i$: the confidence score of the bounding box j in the cell *i*.

$\hat{p}_i$: the conditional probability of the presence of a person in cell *i*.

The Yolo v2 model was originally deployed on a high-performance GPU and cannot be used for embedded implementation. To solve this, a custom lightweight backbone was proposed. The proposed backbone was composed of 7 convolution layers followed each by a non-linear activation layer and batch normalization layer. No pooling layers was used, and the subsampling process was performed using the strided convolution layers with a stride of 2 to optimize the number of operation and avoid the decrease of accuracy. Table 1 present the different configuration of the proposed backbone. For all convolution layers a kernel size of 3x3 with a stride of 2. The number of filters starts with 16 filters in the first convolution layer and doubled in each next layer except for the fifth and sixth layers and for the last layer, 125 filters were used.

Table 1. Configuration of the proposed backbone

Layer	Number of filters	Kernel size	Stride
Convolution 1	16	3 x 3	2
Convolution 2	32	3 x 3	2
Convolution 3	64	3 x 3	2
Convolution 4	128	3 x 3	2
Convolution 5	128	3 x 3	2
Convolution 6	256	3 x 3	2
Convolution 7	125	3 x 3	2
Detection		1 x 1	1

The input image was resized to 128x128 to speed up the processing time. This may degrade the precision, but this degradation can be ignored and does not affect the total performance and we must focus on speed over precision.

The proposed model must be more optimized to fit the embedded device. First, the proposed model was pruned by removing redundant and weak connections. It was proved in the study of Han et al. [39] that pruning the model can reduce its size 9 times without a big loss in precision. The proposed model is composed only of convolution layers and ordinary pruning cannot be useful. So, we propose to prune the whole filter instead of selecting weights. To achieve better results, we propose to apply quantization alongside the pruning. Using fixed-point representation speed up the processing time and reduce memory usage. But it degrades the precision. The optimization techniques allow to speed up the processing speed and allow to reduce the needed computation resources but affect the precision. To recover the precision the model was fine-tuned using the same training data after applying the optimization techniques.

To measure the distance between two persons, we first define the detection area, and then we define the social distance in vertical and horizontal directions. Figure 3 presents how to define social distance. Points 1, 2, 3, 4 are used to define the detection area, and the distance between point 5 and point 6 defines the social distance in the horizontal direction, and the distance between point 5 and point 7 defines the social distance in the vertical direction.

Figure 3. Definition of detection area and social distance in a different direction

4. EXPERIMENT AND RESULTS

For the training and testing of the proposed model, a desktop works with a Linux operating system equipped with an Intel i7 CPU with 32 G of RAM, and an Nvidia GTX960 GPU was used. The proposed model was developed based on the PyTorch deep learning framework. The open cv library was used to process images and to measure the social distance.

The inference of the proposed model was deployed on a Pynq bord. The Pynq board is a Xilinx product equipped with a zynq MPSOC suitable for python programming. In this work, we used the Pynq z1 which is based on the ZYNQ XC7Z020. The Pynq z1 is equipped with a dual-core ARM Cortex A9, 512 MB of RAM, and Artix-7 family programmable logic. The programmable logic is composed of 13300 logic slices, each with four 6-input LUTs and 8 flip-flops, 630 KB of fast block RAM, 220 DSP slices, and many other components. Also, the board is equipped with Gigabit Ethernet PHY for internet connection and an input and output HDMI connection that allows the process and display visual data easily. A Linux image disk can be loaded to the Pynq for fast configuration

and easy use. The Pynq board is a low-cost device and can be used for all commercial RGB cameras and display monitors. It is considered a good solution for social distancing detection which provides a good trade-off between cost and performance. Figure 4 presents an illustration of the Pynq z1 board.

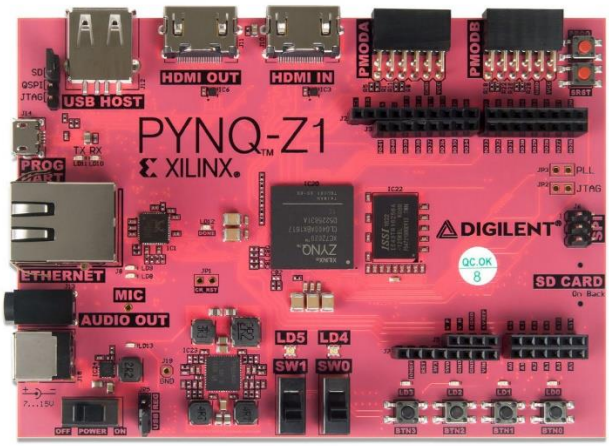

Figure 4. Pynq z1 board

To train and evaluate the proposed model the MSCOCO 2017 dataset was used. It is a dataset built for many tasks such as object detection and key point estimation. The dataset contains more than 118000 images from 80 classes. In this work, we only consider the class person, and the other classes were used as negative data to make the model focus on the desired task.

The evaluation of the proposed model was performed using the MSCOCO dataset by taking into account only the class person. The Adam optimizer was used to train the proposed model. The Adam optimizer has many advantages compared to other optimizers which are not limited to accelerating the convergence of the model and optimizing the learning rate while optimizing the model. An initial learning rate of 0.01 was used with a weight decay of 0.005. The model was trained for 110 K iterations. The backbone was initially pretrained on the imageNet dataset and the weights were used to initialize the detection model. The curve of loss function optimization is presented in Figure 5.

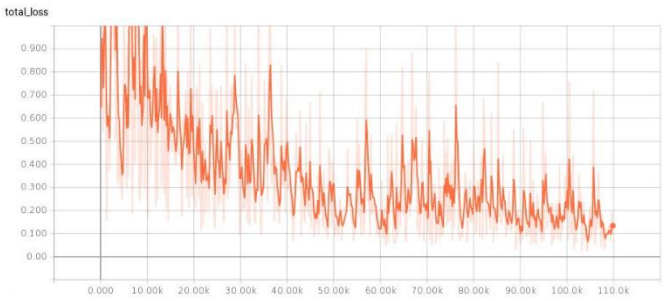

Figure 5. Loss function optimization curve

A mean average precision (mAP) of 87.98%. After compressing the model, an mAP of 83.45% was achieved. The model compression degrades the mAP but accelerates the processing time from 15 FPS to 21 FPS on the testing desktop. The proposed model was very effective compared to the state-of-the-art models. Table 2 present a comparison against state-of-the-art models in social distance detection.

Table 2. Comparison against state-of-the-art models

Model	mAP (%)	Speed (FPS)
Ramadass et al. [37]	54.73	7
Punn et al. [36]	84.6	23
Ours	87.98	15
Ours compressed	83.45	21

The inference of the model was implemented on the Pynq z1 board using the xf DNN module for neural network implementation on Xilinx hardware. Also, the xf open cv library was used for fast image processing on the hardware device. The xf DNN allows the partition of the neural network on the available resources (hardware and software). In this work, the input and output layers were implemented on the software part and the hidden layers (convolution) were implemented on the hardware where a data reuse technique was enabled to eliminate the connection with the global memory and work on the memory of the programmable arrays. Such a technique allows speeding up the processing speed and takes advantage of the full hardware device. A processing speed of 17 FPS was achieved. Figure 6 presents an example of social distance detection using the proposed model.

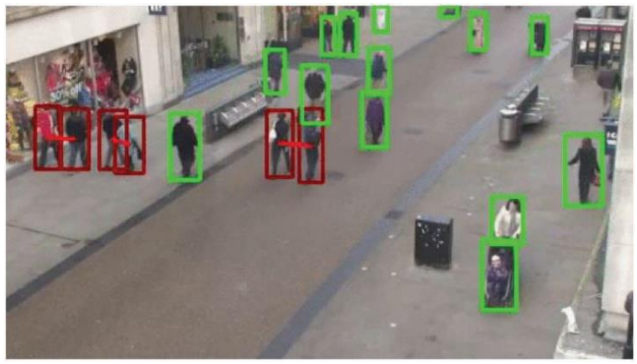

Figure 6. Demo of social distance detection

The achieved results proved the efficiency of the proposed model for social distance detection. It was useful to propose a lightweight backbone and compress for embedded implementation. The Yolo v2 detection methodology was very effective and has allowed achieving good results. The proposed anchors have enhanced the precision by focusing on a predefined shape and ignoring other objects.

The limitation of the proposed technique is caused by the occlusion that degraded the model's capability to measure the distance between persons. Maybe using a multi-cameras view can be a solution to this problem.

5. CONCLUSIONS

COVID-19 is causing a global crisis and without an efficient cure, many important rules must respect to reduce the spreading speed. Social distancing was defined as a principal rule to fight this disease. Unfortunately, people are not responding, and social distancing is not respected. In this work, we propose a social distance detector to warn violators in public spaces and workspaces. The proposed approach was based on the Yolo v2 model for person detection the distance between them was measured. Good results were achieved with 87.98% of mAP and 17 FPS of processing speed. The proposed model was implemented on the Pynq board.

ACKNOWLEDGMENT

The authors extend their appreciation to the Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia for funding this research work through the project number "IF_2020_NBU_219".

REFERENCES

[1] COVID-19 CORONAVIRUS PANDEMIC. Available at: https://www.worldometers.info/coronavirus/?utm_campaign=homeAdvegas1, Last accessed on 20 Feb. 2022.

[2] Goodfellow, I., Yoshua, B., Aaron C. (2016). Deep Learning. 1. Cambridge: MIT Press.

[3] Razzak, M.I., Naz, S., Zaib, A. (2018). Deep learning for medical image processing: Overview, challenges and the future. Classification in BioApps, 323-350. https://doi.org/10.1007/978-3-319-65981-7_12

[4] Young, T., Hazarika, D., Poria, S., Cambria, E. (2018). Recent trends in deep learning based natural language processing. IEEE Computational Intelligence Magazine, 13(3): 55-75. https://doi.org/10.1109/MCI.2018.2840738

[5] He, M., He, D. (2020). A new hybrid deep signal processing approach for bearing fault diagnosis using vibration signals. Neurocomputing, 396: 542-555. https://doi.org/10.1016/j.neucom.2018.12.088

[6] Albawi, S., Mohammed, T.A., Al-Zawi, S. (2017). Understanding of a convolutional neural network. In 2017 International Conference on Engineering and Technology (ICET), pp. 1-6. https://doi.org/10.1109/ICEngTechnol.2017.8308186

[7] Guo, T., Dong, J., Li, H., Gao, Y. (2017). Simple convolutional neural network on image classification. In 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA), pp. 721-724. https://doi.org/10.1109/ICBDA.2017.8078730

[8] Hou, R., Chen, C., Shah, M. (2017). Tube convolutional neural network (T-CNN) for action detection in videos. In Proceedings of the IEEE International Conference on Computer Vision, pp. 5822-5831.

[9] Ayachi, R., Afif, M., Said, Y., Atri, M. (2018). Strided convolution instead of max pooling for memory efficiency of convolutional neural networks. In International Conference on the Sciences of Electronics, Technologies of Information and Telecommunications, pp. 234-243. https://doi.org/10.1007/978-3-030-21005-2_23

[10] Afif, M., Ayachi, R., Said, Y., Pissaloux, E., Atri, M. (2020). An evaluation of RetinaNet on indoor object detection for blind and visually impaired persons assistance navigation. Neural Processing Letters, 51(3): 2265-2279. https://doi.org/10.1007/s11063-020-10197-9

[11] Afif, M., Said, Y., Pissaloux, E., Atri, M. (2020). Recognizing signs and doors for Indoor Wayfinding for Blind and Visually Impaired Persons. In 2020 5th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), pp. 1-4. https://doi.org/10.1109/ATSIP49331.2020.9231933

[12] Afif, M., Ayachi, R., Said, Y., Atri, M. (2020). Deep learning based application for indoor scene recognition. Neural Processing Letters, 51(3): 2827-2837. https://doi.org/10.1007/s11063-020-10231-w

[13] Afif, M., Ayachi, R., Said, Y., Pissaloux, E., Atri, M. (2018). Indoor image recognition and classification via deep convolutional neural network. In International Conference on the Sciences of Electronics, Technologies of Information and Telecommunications, pp. 364-371. https://doi.org/10.1007/978-3-030-21005-2_35

[14] Ayachi, R., Afif, M., Said, Y., Atri, M. (2020). Traffic signs detection for real-world application of an advanced driving assisting system using deep learning. Neural Processing Letters, 51(1): 837-851. https://doi.org/10.1007/s11063-019-10115-8

[15] Said, R.A.Y.E., Atri, M. (2019). To perform road signs recognition for autonomous vehicles using cascaded deep learning pipeline. Artificial Intelligence Advances, 1(1): 1-58. https://doi.org/10.30564/aia.v1i1.569

[16] Ayachi, R., Afif, M., Said, Y., Abdelaali, A.B. (2020). Pedestrian detection for advanced driving assisting system: A transfer learning approach. In 2020 5th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), pp. 1-5. https://doi.org/10.1109/ATSIP49331.2020.9231559

[17] Ayachi, R., Said, Y., Ben Abdelaali, A. (2020). Pedestrian detection based on light-weighted separable convolution for advanced driver assistance systems. Neural Processing Letters, 52(3): 2655-2668. https://doi.org/10.1007/s11063-020-10367-9

[18] Ayachi, R., Said, Y., Atri, M. (2021). A convolutional neural network to perform object detection and identification in visual large-scale data. Big Data, 9(1): 41-52. https://doi.org/10.1089/big.2019.0093

[19] Şüyun, S.B., Taşdemir, Ş., Biliş, S., Milea, A. (2021). Using a deep learning system that classifies hypertensive retinopathy based on the fundus images of patients of wide age. Traitement du Signal, 38(1): 207-213. https://doi.org/10.18280/ts.380122

[20] Zhao, C., Ni, B., Zhang, J., Zhao, Q., Zhang, W., Tian, Q. (2019). Variational convolutional neural network pruning. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2775-2784. https://doi.org/10.1109/CVPR.2019.00289

[21] Molchanov, P., Tyree, S., Karras, T., Aila, T., Kautz, J. (2016). Pruning convolutional neural networks for resource efficient inference. arXiv preprint arXiv:1611.06440. https://doi.org/10.48550/arXiv.1611.06440

[22] Wu, J., Leng, C., Wang, Y., Hu, Q., Cheng, J. (2016). Quantized convolutional neural networks for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4820-4828.

[23] Wang, K., Liu, Z., Lin, Y., Lin, J., Han, S. (2019). Haq: Hardware-aware automated quantization with mixed precision. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8612-8620. https://doi.org/10.48550/arXiv.1811.08886

[24] Banner, R., Nahshan, Y., Soudry, D. (2019). Post training 4-bit quantization of convolutional networks for rapid-deployment. Advances in Neural Information Processing Systems, 32: 7950-7958.

[25] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You only look once: Unified, real-time object detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788. https://doi.org/10.1109/CVPR.2016.91

[26] Redmon, J., Farhadi, A. (2017). YOLO9000: Better,

faster, stronger. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6517-6525. https://doi.org/10.1109/CVPR.2017.690

[27] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L. (2014). Microsoft coco: Common objects in context. In European Conference on Computer Vision, pp. 740-755. https://doi.org/10.1007/978-3-319-10602-1_48

[28] Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A. (2010). The pascal visual object classes (VOC) challenge. International Journal of Computer Vision, 88(2): 303-338. https://doi.org/10.1007/s11263-009-0275-4

[29] Ren, S., He, K., Girshick, R., Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell., 39(6): 1137-1149. https://doi.org/10.1109/TPAMI.2016.2577031

[30] Simonyan, K., Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556. https://doi.org/10.48550/arXiv.1409.1556

[31] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770-778. https://doi.org/10.1109/CVPR.2016.90

[32] Redmon, J., Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767. https://doi.org/10.48550/arXiv.1804.02767

[33] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C. (2016). SSD: Single shot MultiBox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds) Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science(), vol 9905. Springer, Cham. https://doi.org/10.1007/978-3-319-46448-0_2

[34] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4510-4520. https://doi.org/10.1109/CVPR.2018.00474

[35] Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P. (2017). Focal loss for dense object detection. 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2999-3007. https://doi.org/10.1109/ICCV.2017.324

[36] Punn, N.S., Sonbhadra, S.K., Agarwal, S., Rai, G. (2020). Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLO v3 and Deepsort techniques. arXiv preprint arXiv:2005.01385. https://doi.org/10.48550/arXiv.2005.01385

[37] Ramadass, L., Arunachalam, S., Sagayasree, Z. (2020). Applying deep learning algorithm to maintain social distance in public place through drone technology. International Journal of Pervasive Computing and Communications, 16(3): 223-234. https://doi.org/10.1108/IJPCC-05-2020-0046

[38] PYNQ is an open-source project from Xilinx that makes it easier to use Xilinx platforms. Available at: http://www.pynq.io/, last accessed on 28 Oct. 2021.

[39] Han, S., Mao, H., Dally, W.J. (2015). Compressing deep neural networks with pruning, trained quantization and Huffman coding. arXiv 2015. arXiv preprint arXiv:1510.00149.