Information Fusion Research (2024) Volume 2 Issue 1
doi: 10.59429/ifr.v2i1.6556

RESEARCH ARTICLE

Implementing Anchor Free Model for Social Distancing Detection on FPGA Board

Riadh Ayachi[1*], Mouna Afif[1], Yahia Said[2], and Abdessalem Ben Abdelali[1]

[1] *Laboratory of Electronics and Microelectronics, Faculty of Sciences of Monastir, University of Monastir, Monastir, Tunisia*

[2] *Electrical Engineering Department, College of Engineering, Northern Border University, Arar, Saudi Arabia*

***Corresponding author:** Riadh Ayachi, riadh.ayachi@fsm.rnu.tn

ABSTRACT

Since 2019, the world has known a global pandemic caused by the COVID-19 virus. The epidemic was spreading very fast and many precautions should be respected to fight the disease. In order to limit the CVID-19 spread, different ways can be adopted. Social distancing is one of the most important cautions that should be respected. It means keeping a safe social distance between persons in order to avoid COVID-19 contagion. The social distance is about one meter at least. Building new tools used to perform social distance system present a very challenging task.

We propose in this paper to build a social distancing system based on one-stage neural networks. The proposed system is developed based on an improved version FCOS model. In order to ensure an embedded implementation of the proposed work, we used EfficientNet v1 as a network backbone and we applied compression techniques to reduce the model complexity and computation resources. The inference stage of the model has been performed on a ZCU 102 board. Training and testing experiments have demonstrated the efficiency of the proposed work in terms of accuracy as well as processing time.

Keywords: COVID-19, social distance detection, Deep convolutional neural networks, Xilinx ZCU 102

1. Introduction

Since 2019, the world has known a global pandemic caused by the COVID-19 virus. It firstly appeared in Wuhan, China. The World Health Organization referred to this virus as "severe acute respiratory syndrome coronavirus 2." (SARS-COV-2). To date, more than 575 million cases have been confirmed around the world [1]. This number is rising as there is no accurate detection systems are developed. Different researchers have worked on developing new COVID-19 detection to control and fight the epidemic.

The biggest problem of COVID-19 is its fast spread through the air which makes it very hard to be controlled. One of the most significant precautions that have been implemented to prevent the outbreak is social withdrawal. Maintaining a minimum of one meter between two people helps lower the likelihood of

ARTICLE INFO
Received: 13 July 2024 | Accepted: 27 August 2024 | Available online: 9 September 2024

CITATION
Ayachi R, Afif1 M, Said Y, et al. Implementing Anchor Free Model for Social Distancing Detection on FPGA Board. *Information Fusion Research* 2024; 2(1): 6556. doi: 10.59429/ifr/v2i1.6556

COVID-19 infection. To build such a system used to control the social distancing in public spaces such as supermarkets, airports, stations, and so on… artificial intelligence should be involved.

Since their appearance, deep learning-based architectures have presented great performances in solving different artificial intelligence and computer vision tasks. Deep learning involves the use of deep convolutional neural networks (DCNN) to solve different problems such as indoor objects detection [2, 3], scene recognition [4], fatigue warning [5], road sign detection [6], wayfinding assistance [7], and medical imaging [8]. The main strength of such algorithms is their great ability on learning features directly from the input and without using hand-crafted algorithms.

Deep convolutional neural networks (DCNNs) are the most famous models among deep learning architectures. They were inspired by the human brain and mimic its way to process things. CNNs are generally composed of input layers that present the data, a convolution layer used to extract the main features from the input data, and a non-linear activation layer used to enable the network to work on more complex tasks. Additionally, pooling layers are employed to lower the complexity of the network computation and the size of the feature map, and the output layer displays the class prediction.

Designing FPGA-based systems for social distancing detection requires custom hardware architecture. This involves creating a tailored logic design for data processing, which can be highly complex and time-consuming. FPGA devices have a limited number of logic blocks, memory, and other resources. Efficiently utilizing these resources while meeting the performance requirements for real-time video processing is challenging. Ensuring that all components of the FPGA design work synchronously is crucial. Timing issues can lead to incorrect data processing, especially when dealing with high-speed data like real-time video streams. Although FPGAs are more energy-efficient than general-purpose processors for specific tasks, they still consume significant power, especially when processing high-resolution video data in real-time. This can be a challenge in portable or battery-operated systems.

Social distancing detection requires processing video feeds in real-time, which demands high data throughput. Achieving this on an FPGA requires careful optimization of data paths and memory access patterns. Minimizing latency is critical in real-time applications. Designing an FPGA system that processes frames quickly enough to provide real-time feedback without delays is challenging.

FPGAs are well-suited for real-time processing tasks because of their deterministic nature. Unlike CPUs and GPUs, which operate under an OS with varying performance due to multitasking, FPGAs execute tasks with predictable timing, making them ideal for social distancing detection where real-time feedback is critical. FPGAs can be configured to perform parallel processing efficiently. This allows for the simultaneous processing of multiple video frames or channels, which is essential for monitoring large areas with multiple cameras.

FPGAs can handle data directly from sensors or cameras without the need for intermediate processing steps typical in CPU/GPU-based systems. This direct handling reduces latency, providing quicker detection and response times. By implementing critical parts of the social distancing detection algorithm in hardware, FPGAs can accelerate processing, leading to lower overall latency compared to software-based solutions.

While FPGAs consume significant power, they can be optimized for specific tasks, potentially leading to lower overall energy consumption compared to a general-purpose processor running similar algorithms. This makes FPGAs suitable for applications where power efficiency is important, such as in edge devices deployed in remote locations.

In order to ensure lightweight implementation of such CNNs models different compression techniques should be applied such are weight pruning [9]. Its main aim is to remove the unimportant network weights that do not contribute to network efficiency. This technique allows for reducing the network's neurons number which contributes to reducing the computation complexity as well as the processing time. There are various pruning methods that can be used, including pre-training pruning and post-training pruning. Due to the fact that the training phase will be carried out on a high-performance desktop and the inference portion will be performed on a low-end device, the proposed study will concentrate on trimming the model for inference [10].

A second compression technique that can be applied to ensure CNN models implementation on embedded devices is quantization [11]. The number of bits of weight representation is reduced by using this technique. It changes the floating-point representation to a fixed-point representation. In Banner, Ron, Yury Nahshan, and Daniel Soudry [12], the authors proved that changing CNN weights representation from 32-bits float to 8-bits integer does not lead to a significant accuracy drop. The DCNN model is optimized to operate on low-end devices and to be implemented in public spaces to control social distancing without the need for huge calculators. The system will be charged to calculate the social distance and based on the measured value, it predicts if the social distance is respected or not. In order to train and test the proposed work, MS COCO dataset [13] was used. To focus on the desired task, we only consider the class person in the dataset while the rest is considered negative data. Experiments achievements have demonstrated the efficiency of the proposed work with a precision of 89.37 % and a processing time of 17 FPs.

The proposed social distancing system is composed of two main parts: first, it detects persons, and second, it calculates the distance between persons. The proposed social distancing system is developed based on a lightweight version of FCOS (fully convolutional one-stage object detector) [14] with efficientNet v1 [15] as a backbone to ensure its mobile implementation on embedded devices.

The following are the significant advancements made by the proposed work:

- The developed work presents the first work evaluating the fully convolutional one-stage (FCOS) neural network to build a social distancing system.
- The developed social distancing system can widely reduce the number of COVID-19 infections by detecting social distance violations.
- The proposed work introduces the use of two compression techniques that have widely reduced the

model size and the computation complexity.

- We developed a social distancing system based on an improved version of FCOS with efficientNet v1 as a network backbone.
- The proposed system presents the first work evaluating a social distancing system on a Xilinx ZCU 102 board.

The remainder of the rest of the paper is the following: section 2 will review the related works, and section 3 will detail the proposed architecture used to develop the proposed social distancing system. Experiments and discussions are presented in section and section 5 will conclude the paper.

2. Related works

Since its appearance in 2019, COVID-19 disease caused a big crisis around the world. Many works have been proposed in order to fight the epidemic and control the spread of this virus. In order to ensure more safety, different rules have been imposed. One of the rules that should be fully respected is social distancing. This task can be divided into two main parts: person detection and social distance calculation. For the two parts, different related works have been proposed.

Person detection

Person detection presents one of the most important tasks of computer vision and artificial intelligence fields. Different datasets considered the person as a relevant class in MSCOCO [13] and Pascal VOC [16].

Deep learning-based architectures have been widely used in this field. Faster RCNN [17] presents the first deep convolutional neural networks (DCNNs) used for object detection purposes. The two primary components of this model are region proposal network (RPN) for region proposal generation for object detection and features extraction based on various models, including VGG [18] and ResNet [19]. The faster RCNN model achieved good detection accuracy while it was very slow in processing time. In order to accelerate the processing time, you only look once (YOLO) network has been proposed [20]; it was essentially designed to accelerate the processing time without a big accuracy drop. The RPN region has been removed in this architecture and the model was designed to perform features extraction and object detection at one stage. Identification and detection tasks have been performed simultaneously and were solved as a regression problem. YOLO model was very fast and achieved a very interesting processing time. This model also enhances the detection accuracy but this model struggles in detecting small objects. To address this problem more versions of YOLO have been proposed as YOLOV2 [21] and YOLO v3 [22]. YOLO v2 comes with new composition and it widely enhances the detection accuracy. YOLO v3 improved the detection accuracy but it was computationally excessive.

To achieve a fair trade-off between model precision and processing time, a single shot multi-box detector (SSD) [23] was developed. In SSD design, more layers have been added to create a pyramid-shaped structure for detecting objects at various scales and levels. This new design was really helpful and successfully struck a balance between accuracy and processing speed. SSD network was very computationally extensive. Also, it suffers from the class-imbalance problem.

Another famous one-stage neural network named RetinaNet [24] has been proposed to address the problem of class imbalance. It proposes a new rethinked loss function named "focal loss" which applies a modulating term that focuses on learning hard examples for low precision classes. RetinaNet is composed of two regions: one for features extraction and the second for the detection consisting of classification and localization heads. The focal loss was very effective to achieve new state-of-the-art results. RetineNet network is very computation extensive and must be implemented on high-performance GPU.

As mentioned above, all these models were used for object detection purposes and can be reused and fine-tuned for social distance detection.

Social distance detection

Deep learning and artificial intelligence are very useful to provide new solutions to fight the COVID-19 epidemic. One of the rules that should be fully respected is the social distance to break the spread of this virus. Many works have been proposed to address this problem.

Narinder et al. [25] proposed to build a new social distancing system based on the YOLO v3 [22] neural network and deep sort techniques. The YOLO v3 program is used to detect persons in the input data. The goal of the deep sort technique is to give each individual in the image an ID so that they can be tracked in the movie. YOLO v3 was initially adjusted for person detection after being used for general object detection. With the NVIDIA GTX 1060 GPU used for implementation, this work reached 84.6% as mAP and a processing speed of 23 FPS. High-performance processors are required to process the system in this method.

A new artificial intelligence-based method used for social distancing is proposed by Ramadass et al. [26]. The developed system was implemented on an embedded drone. The developed system was built based on YOLO v3 neural network [22]. YOLO v3 is used to process the data captured by the camera mounted in the drone and to calculate the distance between persons. After that, it checks whether or not the social distance is respected. In this work, the also authors used YOLO v3 for face mask detection.

To automate the task of monitoring social distancing and face mask recognition using video sequences, a real-time four-stage model with a monocular camera and a framework based on deep learning was proposed [30]. Utilizing a deep association metric method, this study builds upon the Scaled-You Only Look Once (Scaled-YOLOv4) object identification model, as well as Simple Online and Real-time Tracking. To get the distance between boxes, we first utilize the perspective transformation to approximatively use the three-dimensional coordinates with the Euclidean metric. People that breach or cross the social distance can have their faces detected using the Dual Shot Face Detector (DSFD) and the MobileNetv2 face mask model. With training on the MS COCO and Google-Open-Image datasets, the Social-Scaled-YOLOv4 (Social-YOLOv4-P6) model achieves 56.2% accuracy and 32 frames per second real-time performance. Mean-Average-Precision, frame rate, and loss of values are the metrics used to compare the outcomes to other well-known state-of-the-art models. An impressive 99.3% accuracy rate was attained by the DSFD&MobileNetv2 facemask detectors when trained on the Wider Face and Real Face mask datasets.

When it comes to automating mundane manual processes, the ability to recognize social distance automatically is crucial. Using a surveillance camera is one of various ways to identify people's level of social distance in public spaces.

Nevertheless, it is not a simple operation to identify social distance using a camera. While detecting, issues including illumination, occlusion, and low camera resolution are possible. The goal of the work in [31] is to use deep learning (specifically, the YOLOv4 architecture with the Darknet framework) and the CrowdHuman dataset to create a physical distance detection system that is tailored to Indonesian regulations and situations, with a focus on Jakarta. In order to accomplish the detection, the source video is read, the distance between people is detected, and the number of people in close proximity to one another is determined. Training using CSPDarknet53 and VGG16 backbone in YOLOv4 and YOLOv4 Tiny architecture is done utilizing several hyperparameters in order to complete the detection.

To identify and optimize the optimal mix of designs, the research undertakes many investigations. With a mAP50 of 71.59% (74.04% AP50) and 16.2 FPS shown online, the study effectively identifies crowds at the 16th training. The precision and speed of the model are highly dependent on the input size.

Rapid, precise, and practically useful assessments of compliance with lockdown and social distancing policies are essential for policymakers at the Greater London Authority—the UK's regional governing body for London—to lessen the impact of the present COVID-19 epidemic. During the peak of the first wave of the epidemic, our platform provided crucial data that the local transportation agency, Transport for London, used to execute over 700 interventions, including more signage and the extension of pedestrian zoning. Acquiring large, well-defined, varied pedestrian footfall and physical proximity datasets is challenging, but essential, for monitoring city-wide activity (busyness) and, by extension, for discernible policy choices. We overcome this difficulty by processing over 900 camera feeds in near real-time using our current large-scale data processing infrastructure for urban air quality machine learning. This allows us to develop novel estimates of social distance adherence, group recognition, and camera stability. A computer vision and machine learning pipeline that we developed and deployed is detailed in this study. It uses real-time traffic camera feeds to give a sample of, and context for, physical separation and activity on London's streets almost immediately. YOLOv3 and v4 were used for the detection. They were both pre-trained on the massive COCO dataset, which contains objects with 80 different labels. Our goal restricts us to six categories: human, vehicle, bus, motorcycle, bicycle, and truck. Using a training set of custom-made, manually-labeled JamCam-specific data and combined datasets from COCO and MIO-TCD [33] the model was fine-tuned on six labels.

The methods mentioned above addressed the social distancing problem but none of them is ready for use in public spaces due to many limitations. Traditional methods often use multi-stage detection frameworks, which can be complex and slower due to multiple processing stages. The proposed method uses a one-stage

FCOS model, which simplifies the detection process and enhances speed. The use of an improved FCOS model eliminates the need for anchor boxes, a common requirement in many object detection methods, thus reducing computational overhead and improving detection efficiency. EfficientNet v1 is integrated as the backbone, which is known for achieving state-of-the-art performance with fewer parameters and lower computational costs compared to other commonly used backbones like ResNet or VGG. Advanced model compression techniques are employed to ensure the model can run efficiently on the ZCU 102 board, addressing the challenge of deploying deep learning models on resource-limited devices. This is a significant improvement over existing methods that may not prioritize or achieve such efficient embedded implementation. The implementation on a ZCU 102 board showcases the system's capability to operate in real-time, which is essential for practical social distancing applications. Many existing techniques might struggle to achieve real-time performance on similar hardware.

In the next section, we will detail the proposed method used for social distancing based on improved and compressed versions of a fully convolutional one-stage neural network (FCOS).

3. Proposed Approach

From its appearance to date, the COVID-19 disease is causing a real crisis around the world. In order to limit its spread, different precautions should be fully respected as porting the mask and social distancing. A new warning tool used to detect the offenders should be developed.

To mitigate the impact of the fast spreading of COVID-19disease, we propose to build a new methodology based on an improved version of FCOS network to perform a social distancing network. This work is developed to alert people to maintain a safe distance from each other to avoid infection risk.

The FCOS model was selected due to its anchor-free design, which significantly simplifies the object detection process. Traditional anchor-based models require careful tuning of anchor sizes and aspect ratios, which can be computationally intensive and prone to errors, especially when detecting objects at varying scales and orientations. The FCOS model, by eliminating the need for anchors, reduces the computational complexity and improves generalization across different environments, making it particularly well-suited for real-time social distancing detection. The FCOS model has demonstrated competitive accuracy in various object detection benchmarks. Its architecture leverages fully convolutional layers that enable precise localization and classification of objects, which is critical for accurately measuring social distances between individuals in crowded scenes.

EfficientNet v1 was chosen as the network backbone for its ability to achieve a state-of-the-art balance between accuracy and computational efficiency. Unlike traditional backbones such as ResNet or VGG, which can be computationally expensive, EfficientNet v1 scales network width, depth, and resolution in a balanced manner, resulting in a more efficient model with fewer parameters. This makes it ideal for deployment on resource-constrained embedded systems like FPGAs, where power consumption and processing capabilities are limited. EfficientNet v1 is inherently well-suited to compression techniques, such as pruning and quantization, further reducing the model size and making it more suitable for embedded deployment without sacrificing performance.

The ZCU 102 board was selected due to its ability to handle real-time processing demands. FPGAs are known for their parallel processing capabilities, which are crucial for tasks like real-time video analysis and social distancing detection. The ZCU 102 board, built around the Xilinx Zynq UltraScale+ MPSoC, offers a balance of high-performance processing with the flexibility of reconfigurable logic, making it an ideal platform for deploying the proposed neural network model. The FPGA's ability to implement custom hardware accelerators for critical tasks (such as convolutional operations and non-maximum suppression) enables the system to achieve real-time performance. This hardware acceleration is essential for processing high-resolution video feeds in real-time, which is a key requirement for effective social distancing monitoring. The Pynq ZCU 102 provides ample resources for future scalability, allowing the integration of additional features or the processing of multiple video streams simultaneously. Its reconfigurable nature also allows for updates and optimizations without the need for entirely new hardware, providing long-term flexibility.

In order to detect offenders, we propose in the following to build a social distancing system by taking advantage of deep learning techniques. The proposed social distancing system is developed based on FCOS (fully convolutional one-stage objects detector) [14] with efficientNet v1 [15] as a network backbone to ensure a lightweight implementation of the proposed system on limited resources devices. Inference evaluation has been performed using a pynq ZCU 102 board.

FCOS neural network presents an anchor-free neural network used for object detection purposes. By avoiding the anchors, the network will avoid a lot of complex computation. It also eliminates the use of hyper-parameters related to the use of anchor boxes.

Since the appearance of deep learning-based models object detection models have been dominated by the use of anchor-based methods such as faster RCNN [17], YOLO [20], SSD [23], or RetinaNet [24]. Generally, these methods rely on a large number of anchors predictions which require huge computation resources to be involved.

Recent types of models have addressed more attention to anchor-free methods. FCOS network is proposed to directly predict objects based on points tiled on the input data. The architecture of FCOS is based on three main parts: backbone, features pyramid network, classification, center-ness, and regression heads.

Network backbone: We used EfficienNet v1 to guarantee a lightweight implementation of the developed social distancing system. This neural network offers rethought scalability of the model. It offers a network scaling that is balanced in terms of depth, width, and resolution. EfficientNet v1 network presents a new way of network scaling by scaling the network dimensions by using a powerful agent named "compound coefficient". This model provides a new way for obtaining better network accuracy. In contrast to existing CNN modes, which typically scale up the network by adding more layers, like in the ResNet family, the EfficientNet network offers a novel kind of scaling among several network dimensions. Following a lot of grid searches and testing, the compound scaling falls under the following coefficients:

Depth= 1.20

Width= 1.10

Resolution= 1.15

The optimal design for embedded implementations is EfficientNet because it offers fewer parameters. Increasing the network depth is crucial for high-resolution photos in order to capture the key features that are present in the input data In order to catch more of the fine-grained patterns in images with high resolutions, it was also crucial to enhance the network depth. The various model scaling strategies used in the EfficientNet architecture are shown in **Figure 1.**

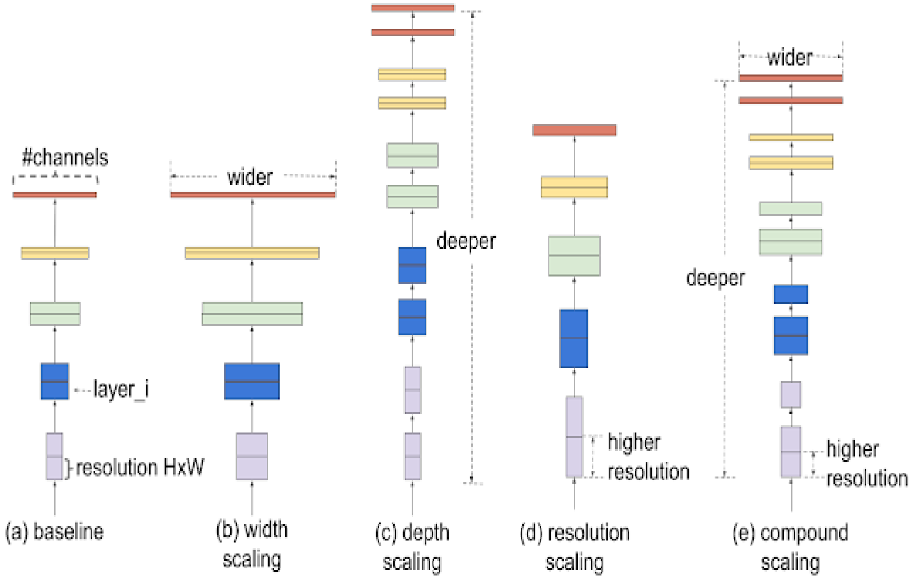

Figure 1. compound scaling technique used in EfficientNet

To balance the parameters scaling, the compound scaling method includes a compound coefficient Φ which aims to uniformly scale the network dimension by new parameters:

Depth =d= α^{Φ}

Width =w= β^{Φ}

Resolution =R= γ^{Φ}

While $\alpha\beta^2\gamma^2 \approx 2$

A>=1, β>=1, γ>=1.

Where α,β, and γ are constantly determined by the grid search

Table 1 provides the EfficientNet B0 architecture.

Table 1: EfficientNet-B0 architecture

Layer	Number of layers
Conv 3x3	1
MBConv 1, 3x3	1
MBConv 6, 3x3	2
MBConv 6, 5x5	2
MBConv 6, 3x3	3
MBConv 6, 5x5	3
MBConv 6, 5x5	4
MBConv 6, 3x3	1
Conv 1x1	1
Pooling	1
FC	1

As presented in table1, this architecture is composed of a set of standard convolutions followed by bottleneck convolution (MBConv). MBConv convolution is used to encode and highlights in a low-dimensional sub-space to decrease the computation complexity. **Figure 2** gives the MBConv design.

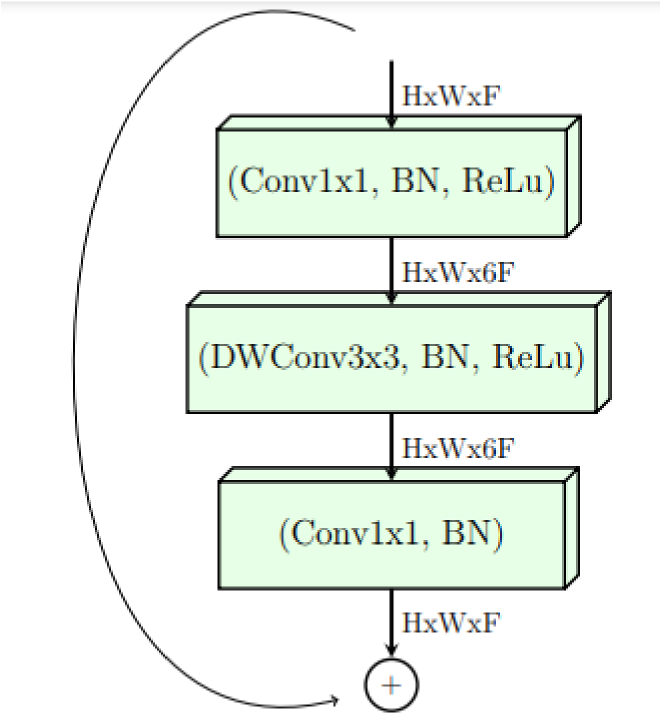

Figure 2. mobile convolution architecture

Features pyramid network (FPN): FCOS network ensures a multi-level prediction using FPN network [27]. FPN predictions are obtained across 5 feature levels. FPN network detects objects across different scales. Deep layer features generally encode lower resolution and are rich in semantic information and shallow layers present higher resolutions with low semantic features. Lateral connections are used to fuse the features between deep layers and shallow layers. This process enhances the detection and localization accuracy of different object sizes: small, medium, and large. FPN detects objects at five levels {P3, P4, P5, P6, P7} with 8, 16, 32, 64 and 128 strides respectively. The outputs of FPN network are fed through a subnetwork that consists of 3 main branches: classification head, center-ness, and regression head.

Classification head: this head predicts a per-pixel probability of the class weighted by the center-ness score with the class probability.

Center-ness head: it aims to calculate the object's center deviation from the location. FCOS introduces the use of center-ness to reduce the low quality of the predicted bounding boxes without using new hyperparameters. The center-ness mind is the adding of a new layer branch in parallel to the classification branch. This parameter presents the normalized distance from the location of the center of the object. The center-ness can be calculated as follows:

$$\sqrt{\frac{\min(l^*,r^*)}{\max(l^*,r^*)} * \frac{\min(t^*,b^*)}{\max(t^*,b^*)}} \quad (1)$$

L*, r*, t*, b* present the regression targets for the location as mentioned in **figure 3**.

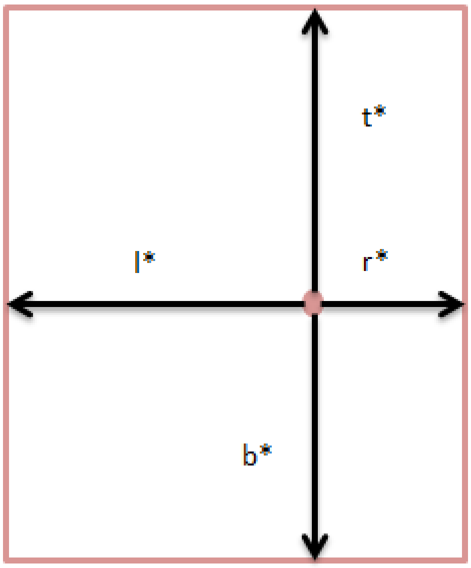

Figure 3. Center-ness calculation technique

The loss function used during the training process is as follows:

$$L(\{P_x,y], \{t_x,y\}) = 1/N_{pos} \sum_{x,y} l_{cls}(P_{x,y}, C^*_{x,y}) + {}^{\lambda}/_{N_{pos}} \mathbb{1}\big(C^*_{x,y} > 0\big)(l_{reg}(t_{x,y}, t^*_{x,y}) \quad (2)$$

l_{cls} present the focal loss.

l_{reg} present the IOU loss.

N_{pos} present the number of positive samples.

$\lambda = 1$, balance weight.

$\mathbb{I}(C_{x,y}^* > 0)$: indicator function =1 if $C_i^* > 0$ and 0 otherwise.

Figure 4 provides a detailed architecture of the Fully connected one-stage object detector used in this work.

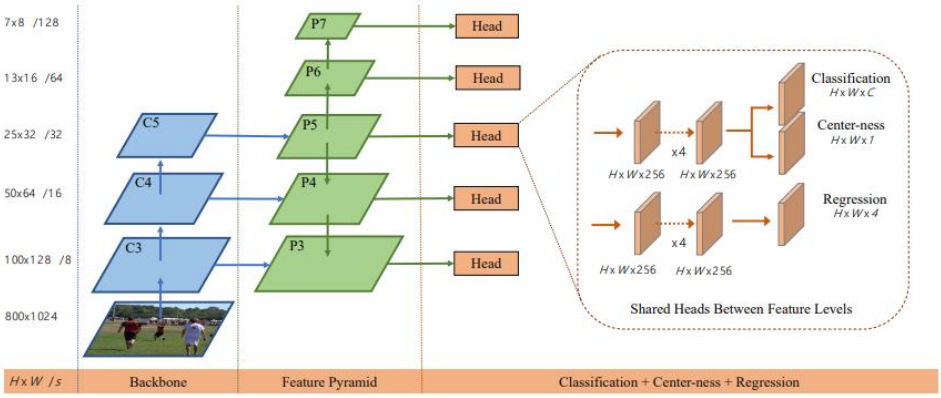

Figure 4. FCOS architecture

In order to build the proposed social distancing system, we applied the transfer learning technique which consists of reusing the first task as a starting point for the neural network for a second task. Transfer learning is a popular approach used in deep learning where a pre-trained model is used as a first point for a new task completely different from the first one. It aims to adopt the pre-trained model weights for the second task. The transfer learning technology enables the network to be generalizable to new datasets and tasks.

In the proposed work, we first retrained the FCOS model with its pre-trained weights on the MS COCO dataset. Generally, applying transfer learning techniques reduces the model training complexity as well as processing time.

We note that transfer learning application is not an easy process as we need to identify which parts and aspects of the model knowledge need to be transferred and that can be relevant for the new task. This process employs identifying the sources and the target that are in common in the two tasks. **Figure 5** presents an illustration of the transfer learning approach.

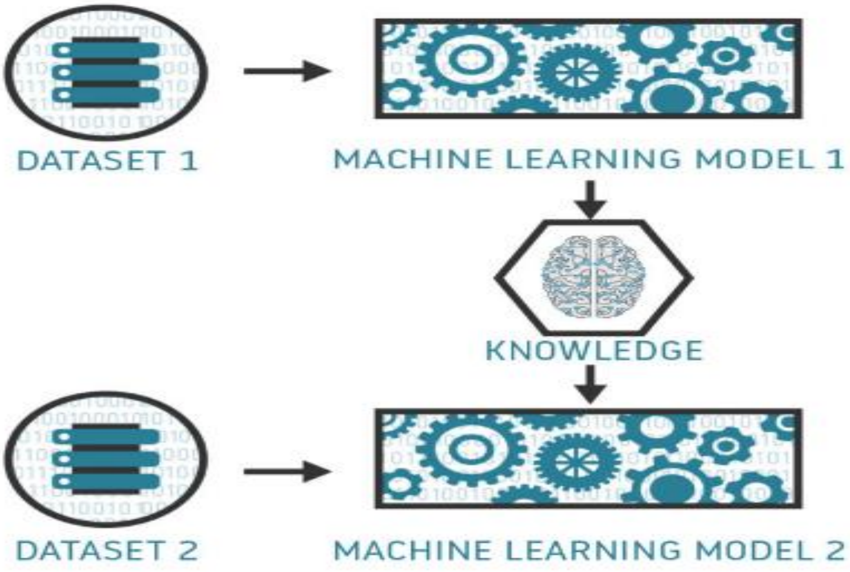

Figure 5: Transfer learning approach illustration

In order to ensure a lightweight implementation of the proposed work, we applied two compression techniques: weights pruning and quantization. The weights pruning techniques aim to remove the unimportant weights that do not contribute to model performances and accuracy. **Figure 6** provides the weights pruning technique.

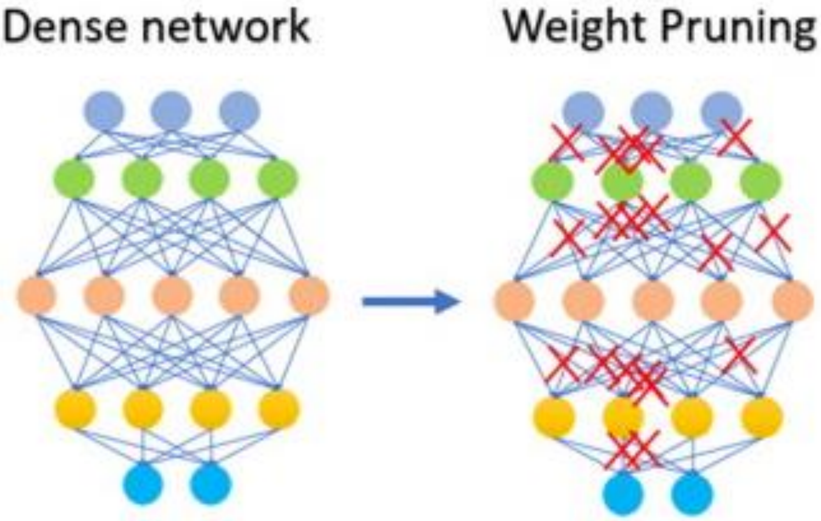

Figure 6. Weights pruning technique

Quantization technique aims to change the parameters presentation from 32floiting point bit to a fixed-point representation. After the application of theses compression technique, the model can be implemented in low-end devices. Figure 7 provides an illustration of the quantization technique used in this work. We note that we changed the parameters presentation from 32 floating point to 8 bits fixed points in the proposed experiments.

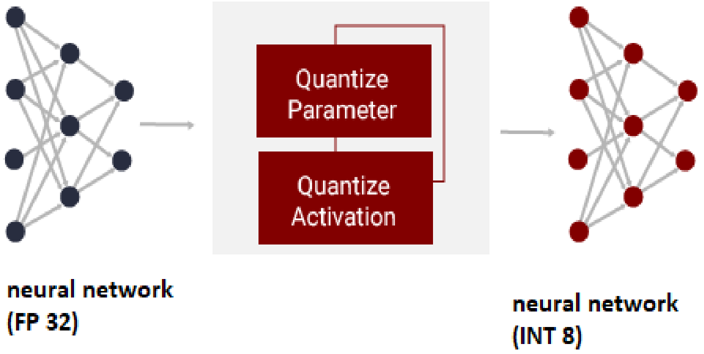

Figure 7. Quantization technique applied in the proposed work

4. Experiments and results

For network training and testing, we used a computer desktop with an intel i7 CPU with 32 GB of RAM and a GTX960 GPU with 2 GB of graphic memory. We used the OpenCV graphic library for image manipulation, display, and to calculate of the social distance.

To ensure a mobile implementation of the proposed social distancing system on an embedded implementation, we used a ZCU 102 board. The Xilinx ZCU 102 evaluation kit is equipped with an ultra-scale MPSoc with an ARM Cortex-A53 quad-core, RF real-time processor, and a mali-400 MP2 GPU. It presents about 600K logic system cells. ZCU 102 board presents 4GB of memory and DDR4 memory with 64-bit and with a second memory component DDR4 of 512 MB. In order to implement the proposed system, we used the VITIS-AI development tool [28] which is developed by the Xilinx hardware platform. This AI tool provides an acceleration library used to implement deep learning models. VITIS AI provides a development kit that presents an AI quantizer, compiler, and optimizer. The AI quantizer aims to change the model parameters presentation from 32 floating points to 8,4 or 2 fixed point presentations. Figure 8 provides the ZCU 102 board.

Figure 8. ZCU 102 board

Training and testing experiments have been performed using the challenging dataset MSCOCO. This dataset can be used to address different problems such as object detection, classification, and segmentation. **Figure 9** provides a subset from the MS COCO dataset.

Figure 9. MS COCO subset

Table 2 depicts the experiment settings used in the proposed work. The proposed model was trained for 100000 iterations with a learning rate of 0.01. Data were divided into training and testing sets where 65% were used for training and 35 % were used for testing. The batch size was fixed to 16.

Table 2. Experiment parameters' settings

Training iterations	100000
Learning rate	0.01
Training set	65%
Testing set	35%
Train batch size	16
Loss function	Focal loss

To train and test the developed social distancing system, we used the MSCOCO 2017 dataset. This dataset is initially built for object detection purposes. It contains more than 11800 images underclasses. We note that the developed social distancing system is efficient enough as it was trained using low-quality images taken under very challenging conditions such as different lighting conditions, day and night images, images with high contrast, and high intra-class variation. In the proposed work, only the class person is considered while the rest of the data is considered as negative data to make the model only focus on the social distancing task.

Table 3 provides the evaluation metrics used in the proposed work.

$$Precision = \frac{TP}{TP+FP} \tag{3}$$

$$Recall = \frac{TP}{TP+FN} \tag{4}$$

$$F1 - score = 2 * \frac{Precision*Recall}{Precision+Recall} \tag{5}$$

Table 3: Evaluation metrics

Precision	85.08
F1-score	87.79
Recall	86,37
Accuracy	89.37

As presented in table 3, we obtained high detection performances for the developed system. We obtained a high precision rate coming up to 85.08 % with a high sensitivity of 86.37%.

Table 4 provides the mean average precision mAP obtained for FCOS network before and after compression techniques application. In order to assess the model performances, we evaluated the work using different evaluation metrics, precision, recall, and F1 score.

Table 4: Obtained results

Model	**Accuracy (%)**	**FPS**
FCOS improved	89,37	17
FCOS Compressed	86.74	24

According to the obtained results, we obtained very encouraging results for the improved and the compressed versions. We obtained 89.37% as the mean average precision for FCOS improved version and 86,74 % for the FCOS compressed version. So, despite we applied two compression techniques, accuracy does not undergo a big drop. Also, we obtained very impressive processing time for the two versions of implementations that outperform the results obtained by the state-of-the-art models. **Table 4** provides a comparison against the state-of-the-art works.

Table 4: Comparison against state-of-the-art models

Model	**mAP (%)**	**Speed (FPS)**
Lalitha et al. [29]	54.73	7
Narinder et al. [25]	84.6	23
Ours	89.37	17
Ours compressed	86.74	24

As presented in **table 4**, our proposed work achieved better results than those of the state-of-the-art. The proposed work outperform works presented in [29] and [25] in terms of detection accuracy and also it outperforms the state-of-the-art method [29] in terms of processing time. The proposed compressed version outperforms methods in [25] and [29] in terms of detection precision as well as processing time. The compressed version of the proposed work demonstrated high efficiency in terms of processing speed. We obtained very competitive detection accuracy despite we applied two compression techniques that can affect the model detection performances.

The inference of the model was performed using ZCU 102 board using the VITIS AI Xilinx tool. We used the OpenCV library for image processing and to calculate social distance. By taking advantage of the hardware resources a processing time of 20 FPS was achieved.

Figure 9 provides a social distance detection demo. We note that the obtained results proved the efficiency of the developed system for social distancing detection. It was very useful to apply two compression techniques: weights pruning and quantization. This method allows obtaining good detection results as fast processing time.

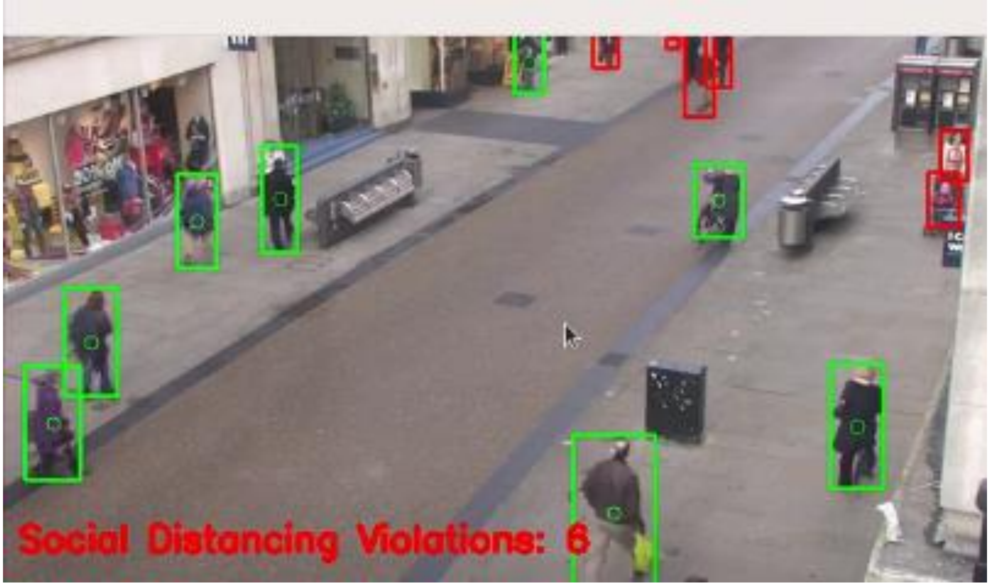

Figure 9. Model detection example

4. Conclusions

Since its emergence, the COVID-19 pandemic has caused a global crisis, affecting millions of people worldwide. To curb the rapid spread of the virus, various preventative measures were enforced, with social distancing recognized as one of the most critical strategies. In this work, we developed a social distancing detection system utilizing a lightweight version of the Fully Convolutional One-Stage (FCOS) network. To enable efficient deployment on the Xilinx ZCU 102 board, we applied two compression techniques: weight pruning and quantization.

Our experimental results demonstrated that both the improved and compressed implementations achieved high detection accuracy, with the system reaching a precision of 89.37%. These results underscore the effectiveness of the proposed method in accurately detecting social distancing while maintaining computational efficiency suitable for real-time embedded applications. As future works, integrating the system with other edge AI devices to create a network of smart sensors could provide comprehensive monitoring solutions for public spaces.

Adapting the system for broader applications beyond COVID-19, such as general crowd management or safety monitoring in public areas, would extend its utility.

Conflict of interest

The authors declare no conflict of interest.

References

1. https://www.worldometers.info/coronavirus/
2. Afif, M., Ayachi, R., Said, Y. et al. An evaluation of EfficientDet for object detection used for indoor robots assistance navigation. J Real-Time Image Proc 19, 651–661 (2022). https://doi.org/10.1007/s11554-022-01212-4
3. Afif, M., Ayachi, R., Said, Y. et al. An efficient object detection system for indoor assistance navigation using deep learning techniques. Multimed Tools Appl 81, 16601–16618 (2022). https://doi.org/10.1007/s11042-022-12577-w
4. Afif, M., Ayachi, R., Said, Y. et al. Deep Learning Based Application for Indoor Scene Recognition. Neural Process Lett 51, 2827–2837 (2020). https://doi.org/10.1007/s11063-020-10231-w
5. R. Ayachi, M. Afif, Y. Said and A. Ben Abdelali, "Drivers Fatigue Detection Using EfficientDet In Advanced Driver Assistance Systems," 2021 18th International Multi-Conference on Systems, Signals & Devices (SSD), 2021, pp. 738-742, doi: 10.1109/SSD52085.2021.9429294.
6. Ayachi, R., Afif, M., Said, Y. et al. An edge implementation of a traffic sign detection system for Advanced driver Assistance Systems. Int J Intell Robot Appl 6, 207–215 (2022). https://doi.org/10.1007/s41315-022-00232-4
7. Afif, M., Ayachi, R., Said, Y. et al. Deep learning-based application for indoor wayfinding assistance navigation. Multimed Tools Appl 80, 27115–27130 (2021). https://doi.org/10.1007/s11042-021-10999-6
8. Marwa Fradi, Mouna Afif and Mohsen Machhout, “Deep Learning based Approach for Bone Diagnosis Classification in Ultrasonic Computed Tomographic Images” International Journal of Advanced Computer Science and Applications(IJACSA), 11(12), 2020. http://dx.doi.org/10.14569/IJACSA.2020.0111210
9. RETSINAS, George, ELAFROU, Athena, GOUMAS, Georgios, et al. Weight pruning via adaptive sparsity loss. arXiv preprint arXiv:2006.02768, 2020.
10. Molchanov, Pavlo, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. "Pruning convolutional neural networks for resource efficient inference." arXiv preprint arXiv:1611.06440 (2016).
11. Wu, Jiaxiang, Cong Leng, Yuhang Wang, Qinghao Hu, and Jian Cheng. "Quantized convolutional neural networks for mobile devices." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4820-4828. 2016.
12. Banner, Ron, Yury Nahshan, and Daniel Soudry. "Post training 4-bit quantization of convolutional networks for rapid-deployment." In Advances in Neural Information Processing Systems, pp. 7950-7958. 2019.
13. Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. "Microsoft coco: Common objects in context." In European conference on computer vision, pp. 740-755. Springer, Cham, 2014.
14. TIAN, Zhi, SHEN, Chunhua, CHEN, Hao, et al. Fcos: Fully convolutional one-stage object detection. In : Proceedings of the IEEE/CVF international conference on computer vision. 2019. p. 9627-9636.
15. TAN, Mingxing et LE, Quoc. Efficientnet: Rethinking model scaling for convolutional neural networks. In : International conference on machine learning. PMLR, 2019. p. 6105-6114.
16. Everingham, Mark, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. "The pascal visual object classes (voc) challenge." International journal of computer vision 88, no. 2 (2010): 303-338.
17. Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks." In Advances in neural information processing systems, pp. 91-99. 2015.
18. Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).
19. He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. 2016.
20. Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi. "You only look once: Unified, real-time object detection." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779-788. 2016.
21. REDMON, Joseph et FARHADI, Ali. YOLO9000: better, faster, stronger. In : Proceedings of the IEEE conference on computer vision and pattern recognition. 2017. p. 7263-7271.

22. REDMON, Joseph et FARHADI, Ali. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767, 2018.
23. Liu, Wei, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. "Ssd: Single shot multibox detector." In European conference on computer vision, pp. 21-37. Springer, Cham, 2016.
24. Lin, Tsung-Yi, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. "Focal loss for dense object detection." In Proceedings of the IEEE international conference on computer vision, pp. 2980-2988. 2017.
25. Singh Punn, Narinder, Sanjay Kumar Sonbhadra, and Sonali Agarwal. "Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLO v3 and Deepsort techniques." arXiv (2020)
26. Ramadass, Lalitha, Sushanth Arunachalam, and Z. Sagayasree. "Applying deep learning algorithm to maintain social distance in public place through drone technology." International Journal of Pervasive Computing and Communications (2020).
27. LIN, Tsung-Yi, DOLLÁR, Piotr, GIRSHICK, Ross, et al. Feature pyramid networks for object detection. In : Proceedings of the IEEE conference on computer vision and pattern recognition. 2017. p. 2117-2125.
28. https://www.xilinx.com/products/design-tools/vitis/vitis-ai.html last accessed: 28/06/2022
29. Ramadass, Lalitha, Sushanth Arunachalam, and Z. Sagayasree. "Applying deep learning algorithm to maintain social distance in public place through drone technology." International Journal of Pervasive Computing and Communications (2020).
30. Mokeddem, Mohammed Lakhdar, Mebarka Belahcene, and Salah Bourennane. "Real-time social distance monitoring and face mask detection based Social-Scaled-YOLOv4, DeepSORT and DSFD&MobileNetv2 for COVID-19." Multimedia Tools and Applications 83, no. 10 (2024): 30613-30639.
31. Chowanda, Andry, Ananda Kevin Refaldo Sariputra, and Ricardo Gunawan Prananto. "Object Detection Model for Web-Based Physical Distancing Detector Using Deep Learning." CommIT (Communication and Information Technology) Journal 18, no. 1 (2024): 89-98.
32. Walsh, James, Oluwafunmilola Kesa, Andrew Wang, Mihai Ilas, Patrick O'Hara, Oscar Giles, Neil Dhir, Mark Girolami, and Theodoros Damoulas. "Near real-time social distance estimation in London." The Computer Journal 67, no. 1 (2024): 95-109.
33. Luo, Zhiming, Frederic Branchaud-Charron, Carl Lemaire, Janusz Konrad, Shaozi Li, Akshaya Mishra, Andrew Achkar, Justin Eichel, and Pierre-Marc Jodoin. "MIO-TCD: A new benchmark dataset for vehicle classification and localization." IEEE Transactions on Image Processing 27, no. 10 (2018): 5129-5141.