Annals of R.S.C.B., ISSN:1583-6258, Vol. 24, Issue 2, 2020, Pages. 165 - 178
Received 24 October 2020; Accepted 15 December 2020.

Implemention of a two Layer (Phase) Covid Mask Detector and Social Distancing Detector

1Reena Thakur , 2Mayuri Wankhede,3Shewta Dandhare,4Nishadraj Kakde, 5Gagandeep Chana

[1]Assistant Professor, Department Of Computer Science & Engineering, Jhulelal Institute Of Technology, Nagpur, Maharashtra, India.

[2345]BE Students, Department of Computer Science & Engineering, Jhulelal Institute Of Technology, Nagpur, Maharashtra, India.

Abstract

COVID-19 may be aunwellness caused by a severe metabolism syndrome coronavirus. it had been known in Dec 2019 in Wuhan, China. it's resulted in associate current pandemic that caused infected cases as well as several deaths. Coronavirus is primarily unfold between folks throughout shut contact. Motivating to the current notion, this analysis proposes a synthetic intelligence system for social distancing classification of persons victimisation thermal pictures. By exploiting YOLOv2 (you cross-check once) approach, a completely unique deep learning detection technique is developed for detection and pursuit folks in indoor and out of doors situations. associate rule is additionally enforced for measure and classifying the gap between persons and to mechanically check if social distancing rules square measure revered or not. Hence, this work aims at minimizing the unfold of the COVID-19 virus by evaluating if and the way persons go with social distancing rules. The projected approach is applied to photographs noninheritable through thermal cameras, to determine an entire AI system for folks pursuit, social distancing classification, and temperature observance. The coaching part is completed with 2 datasets captured from completely different thermal cameras. Ground Truth Labeler app is employed for labeling the persons within the pictures. The projected technique has been deployed during a low-priced embedded system (Jetson Nano) that consists of a hard and fast camera. The projected approach is enforced during a distributed police work video system to examine folks from many cameras in one centralized observance system. The achieved results show that the projected methodology is appropriate to line up a closed-circuit television in good cities for folks detection, social distancing classification, and temperature analysis.

Introduction

COVID-19 could be a sickness caused by a replacement coronavirus that appeared in China in Gregorian calendar month 2019. COVID-19 symptoms embrace chiefly fever, cough, chills, and shortness of breath, body aches, loss of style, and smell. COVID-19 will be severe, and in several cases, it's caused death. The coronavirus will unfold from one person to a different as diagnosed by researchers in laboratories. This pandemic has unfold to over 188 countries round the world [1]. On Gregorian calendar month fifteen, 2020, UN agency (World Health Organization) declared that there are thirty eight,394,169 confirmed COVID-19 cases and 1089,047 deaths [2] round the world. The uncertainty, underpinning, and complexness of the coronavirus have created it tough to predict the length and unfold of this pandemic. As of yet, there's no immunizing agent on the market. bar involves carrying masks and laundry hands oftentimes. AN infected person ought to lodge in home once individuals ar sick to forestall spreading this pandemic to the others. this example forces the worldwide community and governments to seek out the simplest mitigation decide to stop the unfold of coronavirus. Nations stopped their business and closed the border and public places like faculties and workplaces to avoid people’s interactions. it's been according that each one infected countries UN agency applied the lock-down for his or her communities achieved a discount of range|theamount|the quantity} of COVID-19 cases and also the number of deaths from this pandemic.

Fever or chills ar common symptoms of coronavirus. Researchers in China found that ninety nine of individuals infected with the coronavirus given with a extreme temperature. Thermal cameras and non-contact infrared thermometers, that ar non-contact instruments, will be wont to live blood heat. This approach will monitor a person’s surface temperature to limit the unfold of coronavirus infections.

Based on the knowledge from the globe Health Organization, social distancing is that the best follow wherever people will minimize physical contact with potential COVID-19 carriers by maintaining an exact distance between one person and another. the most target is to produce a comprehensive tool and effective technologies which will be utilised to enforce social distancing. Technologies might play a crucial role to facilitate social distancing follow. In such a context, computer science (AI) and knowledge and communication technology (ICT) will be utilized in addressing this challenge.

This analysis aims at mitigating the unfold of this virus in communities and saving the lives of individuals. during this work, we have a tendency to propose a deep learning object detection model for individuals detection together with AN enforced formula for social distancing classification on thermal pictures. Hereafter, the paper is organized as follows: once the introduction of COVID-19 in Sects. 1, two presents the analysis background and connected work. Section three shows an summary of object detection. Section four presents the planned methodology to outline a measuring instrument for individuals detection and social distancing check. Section five shows the experimental results and Sect. half dozen describes the implementation of the planned approach on embedded hardware. Conclusions ar drawn in Sect. 7.

Research background and connected work

Social distancing and temperature screening ar effective tools for preventing the unfold of unwellness. they need been urged by several organizations, together with the globe Health Organization (WHO) [3]. Russel et al. [4] studied the results of social distancing techniques on the unfold of coronavirus. This paper bestowed scientific location contact patterns to supply the mechanical phenomenon of an epidemic by utilizing inclined exposed infected removed (SEIR) strategies. The authors conjointly mentioned that the unforeseen lifting of social distancing might increase the infection and unfold of the virus between individuals. Nabil Kahale [5] highlighted the impact of social distancing measures. The study aimed to derive AN approximation that shows however early social distancing measures will scale back economic loss and therefore the variety of recent infections considerably. At the time once coronavirus is begun spreading across the people and society, analysis and scientists ar beginning to verify the simplest resolution to eliminate the unfold of this pandemic [6, 7]. Jennifer Berglund [8], urged following an individual infected with COVID-19 victimisation GPS and integral applications in smartphones. However, this technology has limitations on following people WHO haven't any Wi-Fi or cell signals. On the opposite hand, some authorities utilize drones with mounted video cameras to trace the gathering of people within the outside space [9, 10]. Such technology is appropriate for watching COVID-19 that might amid the coronavirus irruption.

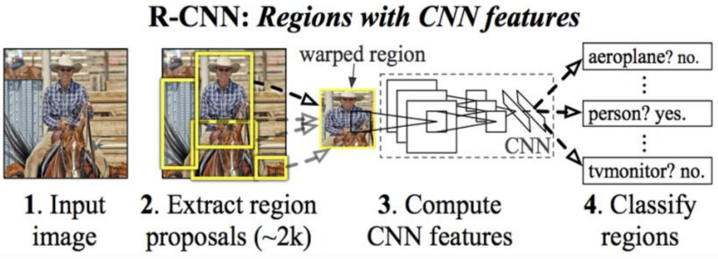
Recently, the matter of classifying and police work the objects in a picture is solved , because of the enhancements in pc vision and deep learning generally. consequently, pc vision development has centered on varied fascinating and difficult topics, like neural vogue transfer, segmentation, and following, and in fact object detection [11].Deep learning is an artificial intelligence function (AI) that emulates the tasks of the human brain in data processing and object detection. It can be referred to as a neural network with a sophisticated algorithm. The history of the neural network dates to 1940 [12]. The original intention of the neural network is to solve learning problems ethically [13]. A convolutional neural network (CNN) is widely used in deep learning models for object detection. CNN is a deep learning algorithm that takes an input image and assigns the learnable weights and biases for various classes in an image and differentiates them from one to another. The convolutional neural network has been made evolution which can be implemented on an embedded system with a low-resolution input and low complexity [14]. There are various deep learning models such as R-CNN, Single-shot detector (SSD), and YOLO which are applied in different applications for object detection. These models are efficient algorithms for movement estimation in video scenes. Ebrahim et al. [15], proposed a technical approach for detecting people using video frames. The author utilized a background subtraction and Gaussian mixture with a deep learning detector for people detection. In method [16], the authors presented a deep learning (CNN) technique for human detection. They utilized a combination of deep learning and machine learning methods to achieve high accuracy and less computation for people detection. Unfortunately, this method had problems with low speed for real-time detection. In method [17], researchers suggested a method on static crowds for a group of people that stayed in the same location for a long time. They utilized the mean of class as support vector machine (SVM) to categorize patches as essential crowds and these patches are extracted by text features.

Recent developments showed that the identification of individuals through video surveillance cameras can be achieved by face [18], and a person's manner of walking. However, the detection of a person under crowds' technique is difficult and hard to optimize.

In method [19], the authors presented a solution for detecting pedestrians with a low-resolution camera by utilizing background subtraction by extracting foreground silhouettes and classifying them in real-time.

Overview of object detection

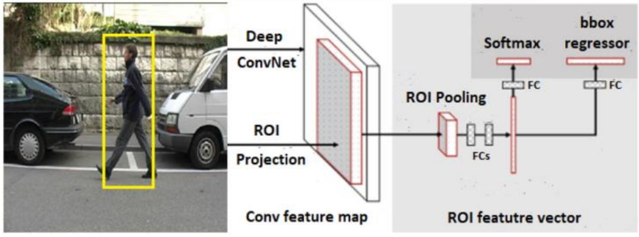
Object detection systems place a bounding box around the objects and associate the correct object's category with each bounding box. Deep learning is an effective method to perform object detection. In [20], Ross Girshick explored a regional convolutional neural network detector (R-CNN). This model consists of four stages. It starts with introducing the images into the input layer, then it extracts the regional proposals, after that it computes the features by CNN, and finally, it classifies these features, see Fig. 1. R-CNN uses selective search algorithms to generate region proposals. It takes a huge amount of time as it would have to classify the regions per image. R-CNN cannot be implemented in real-time object detection as it takes 47 s for each image. R-CNN cannot be trained at one time. Rather, it needs to train every part independently.



Fast R-CNN is another version of the regional proposal algorithm, which was presented by the same author of R-CNN model [21]. Fast R-CNN enhanced the drawbacks from R-CNN to

Received 24 October 2020; Accepted 15 December 2020.

build faster object detection algorithm. It is similar to the R-CNN algorithm. However, the input image is fed to the convolutional neural network to generate a convolutional feature map instead of feeding region proposals to CNN. The region proposal is warped into squares in this model. Using region of interest (ROI) pooling layer, these regions are reshaped into a fixed size which can be fed into a fully connected layer. Softmax layer is used in this architecture to predict the class of region and the offset values of the bounding box. Figure 2 shows the schematic diagram for fast R-CNN detector.

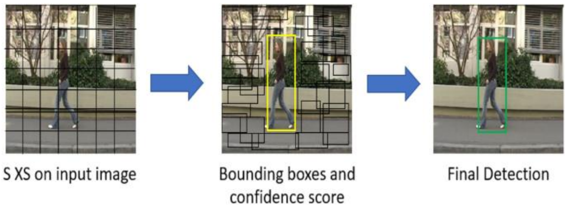


Both algorithms (R-CNN and Fast R-CNN) use selective search to find the region proposals. This process a slow and time-consuming which is affecting the performance for neural network algorithm.

Recent improvements in object detection deep learning include other algorithms such as YOLO and YOLOv2. You look at one or (YOLO) is a state-of-the-art deep learning object detection. It was presented by Joseph Redmon et al. [22]. YOLO uses a single neural network to the whole image. It divides the image into regions and predicts the bounding boxes and the probabilities for each region. These bounding boxes are weighted by predicted probabilities.

YOLO detector looks full image at one time; therefore, its predictions are informed by the context in the image. It predicts with single network evaluation, unlike other object detectors such as (R-CNN) which requires thousands for a single image. YOLO algorithms take the input image and split into $S \times S$ grids. It extracts the features from each grid. It predicts the bounding boxes with confidence scores for the predicted classes in the bounding boxes, see Fig. 3. Each grid cell detects bounding boxes and confidence scores. The bounding box consists of five predictions which are represented with (x, y, w, h) and the confidence score. The (x, y) coordinates reflect the center of the bounding box of the grid cell. The (w, h) represents the width and the height of the full image. The confidence scores represent the measurement of how confident the detector is that the box contains the object to be predicted.

Fig. 3



Schematic diagram for YOLO: input image which splits into S×S grids, each grid predicts the bounding boxes and the confidence scores and finally, the score encodes the probability with bounding box on the detected class

YOLO predicts several bounding boxes for each grid cell. In the training stage, it only requires one predictor of the bounding box to be responsible for each class. The predictor is assigned to predict an object which has the highest Intersection over Union value (IoU) for the ground truth. This process leads to specialization within the bounding boxes prediction. YOLO algorithms use sum-squared error between the ground truth and predictions of bounding boxes for loss. This sum squared error computes the classification, localization, and confidence losses for the model. Therefore, YOLO is optimized with the following loss function to enhance its performance during the training process, see Eq. (1). YOLOv2 is the second version of YOLO. It is an object detection system targeted for real-time processing. It

Received 24 October 2020; Accepted 15 December 2020.

has several improvements to YOLO as explored in YOLO9000 paper [23]. YOLOv2 resolved the issues which were encountered with YOLO, thus improving the processing accuracy and speed for the architecture. It enhanced the errors of localization for the classes to be predicted in the images. It uses batch normalization in all convolutional layers.

Proposed methodology

Social distancing detector steps

This section discusses the essential steps which are attempted to establish a workflow for monitoring social distancing on thermal images as seen in Fig. 4:

1.
Prepare the thermal images or streaming a video from a thermal camera which contains people.

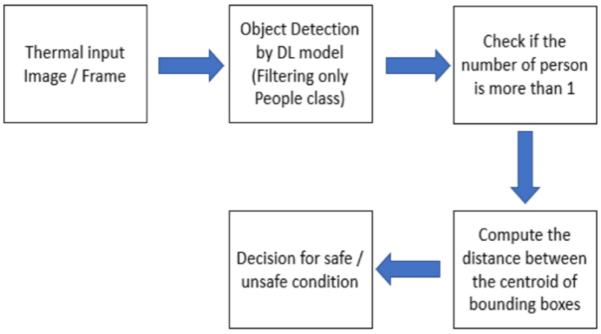
2.
Applying the deep learning object detector to detect people in thermal images or video streams.

3.
Check the number of persons that are in the images or video stream.

4.
Compute the distance between the centroid of the bounding boxes which are enclosed to the detected people.

5.
Finally, the algorithm will decide for safe or unsafe social distancing based on the number of persons and the measured distance between the centroid of bounding boxes.

Fig. 4


Thermal input Image / Frame
Object Detection by DL model (Filtering only People class)
Check if the number of person is more than 1
Compute the distance between the centroid of bounding boxes
Decision for safe / unsafe condition

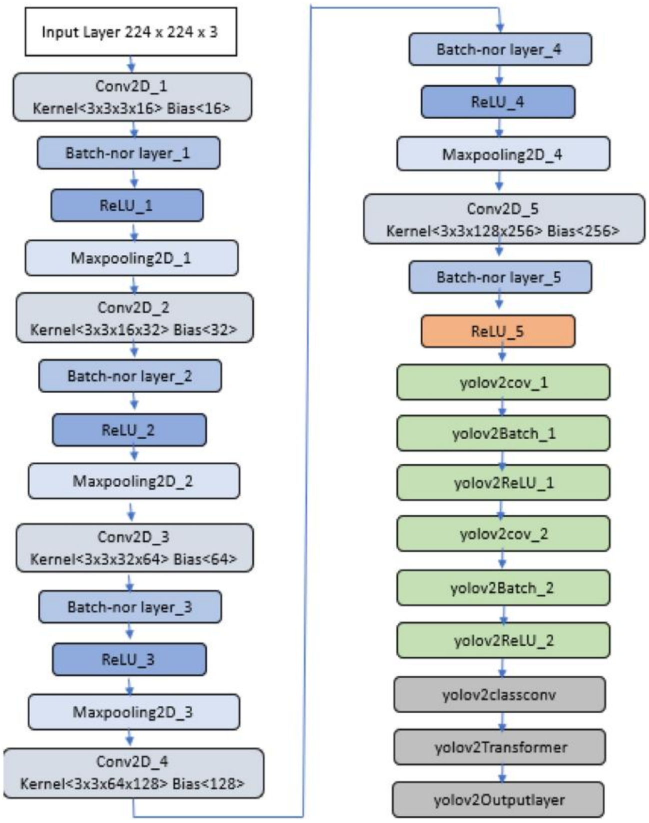
The steps involved for people detection and social distancing classification on thermal images

Neural network design

A Deep Neural Network (DNN) application is used in MATLAB to construct YOLOv2 neural network layers. Then the designed DNN is ported in embedded platforms like NVIDIA Jetson Nano. We built a CNN with 29 layers, see Fig. 5. This is to establish a light-weight model to fit the real-time implementation of CNN inference also in low-cost embedded platforms, such as those of IoT nodes. The neural network layers include the input layer, middle layers, and subnetwork of YOLOv2 layers.

Fig. 5

Received 24 October 2020; Accepted 15 December 2020.



Architecture of YOLOv2 Neural Network

The proposed approach starts with the input image layer, which introduces the input image with a size of (224 × 224 × 3) for our detector. A set of middle layers was used, which includes batch normalization, convolutional, max-pooling, and Relu (rectified linear unit) layers. Convolutional layers were used to map the features for the images. The size of the filter was set to (3 × 3). It defines the height and width of the regions in the input image. Batch normalization layers were used to regularize the model and eliminate the overfitting problem. ReLU activation functions were utilized to introduce the non-linearity to the neural network. Maxpooling layers were used to downsample the images into pooling regions. We applied (2 × 2) for the size of pooling with a stride of (2 × 2) for all max-pooling layers in a neural network. 'ReLU_5' was used as the feature extraction layer. This is to extract the features from neural network layers and then given as input to YOLOv2 subnetwork layers. YOLOv2 layers were used in this detector which constructs YOLOv2 detection network. YOLOv2 Subnetwork consists of a batch of layers that include convolutional (yolov2cov), batch normalization (yolov2Batch), ReLU (yolov2ReLU), transform, and output layers. The transform layer was utilized in YOLOv2 detector to stabilize the network for object localization. This layer transforms the raw CNN output into a form required to produce object detections. YOLOv2 output layer was used which refines the location of bounding boxes to the detected objects. The model was examined with a neural network analyzer and reported zero errors.

Training

The designed network was trained with two different datasets of thermal images. Dataset I consists of 775 thermal images of humans captured in various scenarios while walking, running, or sneaking and in different body positions, as well as different motion speeds, maximizing the simulated conditions for detecting people in the surveillance and monitoring areas. These images were collected from different sources on the internet. Dataset II consists of 800 images. These images are infrared images that were created by FLIR company for thermal cameras . We used ground truth labeler application in MATLAB for labeling the

Received 24 October 2020; Accepted 15 December 2020.

persons in the thermal images . We split the images into 70% for training, 20 for validation, and 10% for testing for each dataset. The model has been trained with stochastic gradient descent (sdgm) [26]. The learning rate parameter in the training option was used to control the model change in response to the error [27]. We started the learning rate with 10^{-2}. However, we noticed that the model was unstable during the training process. The learning rate was fine-tuned at 10^{-3}, and the loss curve for mini-batch was steady with small fluctuation, see Fig. 6. Table 1 shows Training Hyper-Parameters for the proposed neural network.

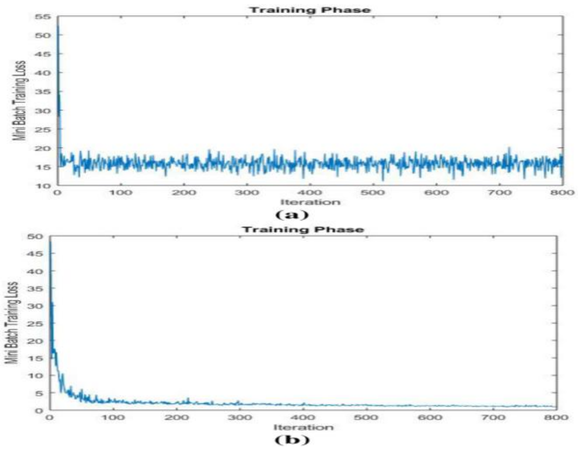


Fig. 6a Mini-Batch Loss Curve before fine-tuning, b Mini-Batch Loss Curve after fine-tuning

Table 1 Training Hyper-Parameters for the proposed neural network

Algorithm for distancing classification

We also implemented code in MATLAB to work with bounding boxes of a detected person in the thermal images. This code classifies and decides if persons in the image are within safe distancing or not. We assigned a green color for safe social distancing and red color for unsafe social distancing for the bounding boxes. First, we find the number of persons in the images. If it is one person, a green color is assigned for a bounding box of detected persons. When we have two or more persons, then color is decided from the function which is called find Color. This function will determine if the bounding box is 2 or more and in addition to that, it will calculate the distance between the centers of bounding boxes for the detected person. The centerpoints, *C* (*x*, *y)* of bounding boxes is measured using the equation as seen in Eq. (3).

C(X,Y)=Xmin+Xmax2,Ymin+Ymax2,C(X,Y)=Xmin+Xmax2,Ymin+Ymax2,

(3)

where: CC is the center point of the bounding box. $X_{\min}$ and $X_{\max}$ are the minimum and maximum values for the corresponding width of the bounding box. $Y_{\min}$ and $Y_{\max}$ are the minimum and maximum values for the corresponding height of the bounding box.

To measure the distance C_1 ($X_{\max}$–$X_{\min}$), and C_2 ($Y_{\max}$–$Y_{\min}$), between the center of each bounding box, we used the Euclidean formula, see Eq. (4), where the distance between pixels is translated in a metric distance (knowing the range and field of view covered by the camera) and then compared to a threshold value. In case of finding- color function detects two bounding boxes and the distance is less than the threshold value, these boxes will have a red color. If this function detects two bounding boxes and the distance is more than the threshold value, the color will be green for these boxes. Figure 7 provides the measured distance (D) between the center of each bounding box for a detected person.

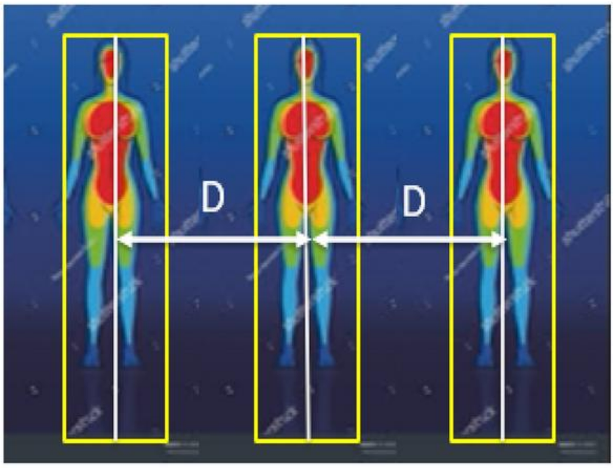
D(C1,C2)=(Xmax−Xmin)2+(Ymax−Ymin)2,−−−−−−−−−−−−−−−−−−−−−−−−−−−−√D(C1,C2)=(Xmax−Xmin)2+(Ymax−Ymin)2,

(4)

Received 24 October 2020; Accepted 15 December 2020.

where: *D* is the distance between the centers of bounding boxes.

Fig. 7



The measured distance (*D*) between the center of each bounding box for a detected person

Closed-circuit television (CCTV) cameras are installed in such a way that they provide angle views on the ground plan. To calculate the distance between the people effectively, a top view of the ground plane is required. This can be performed by applying a homography transformation to the four points coordinates in the angled view. These four points can be transformed as shown in the Eq. (5).

⎡⎣⎢⎢⎢ Xcorn.topYcorn.top1 ⎤⎦⎥⎥⎥=M∗⎡⎣⎢ Xcorn.angYcorn.top1 ⎤⎦⎥,[Xcorn.topYcorn.top1]=M∗[Xcorn.angYcorn.top1],

(5)

where: $X_{\text{corn.ang}}$ and $Y_{\text{corn.ang}}$ represent the pixel coordinates of one of the four points in the CCTV view image $X_{\text{corn.top}}$, $Y_{\text{corn.top}}$ represent the same point after transformed to the top view. MM is the homography matrix.

To estimate the distance between people in the real world, the distance is calculated between the individuals using Eq. 4 and four points coordinates with homography matrix value. This distance is then scaled by factor SS to have the real-world distance between the individuals. The scaling factor SS is obtained by measuring a number of pixels in an image that represents 1 m in the real-world.

Experimental results

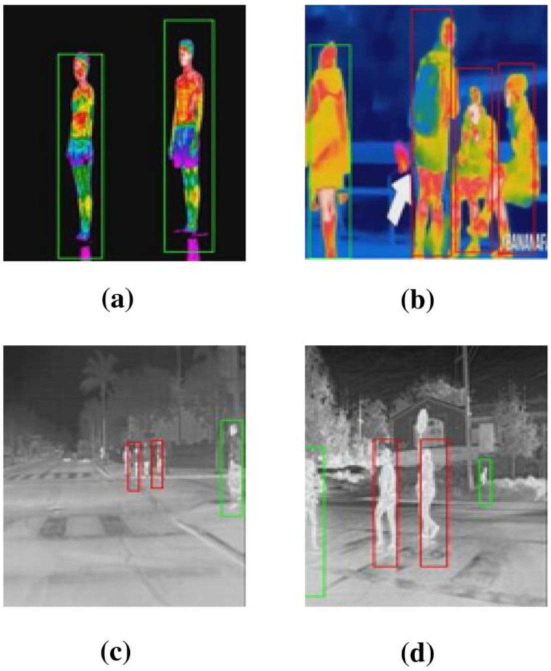
The technique proposed in Sect. 4 was examined with two testing datasets to evaluate the capability of detection and localization of persons in the thermal images. These datasets have been made challenging, which encountered a realistic situation by capturing body temperature on people from real thermal cameras. Motivating to that, we selected these datasets for our experiments. YOLOv2 and distance classification algorithms were applied to these thermal images. YOLOv2 model detects people and provides the bounding box information. After people detection, the Euclidean distance between each detected centroid pair is computed using the detected bounding box and its centroid information based on dimensions of (x, y) for each bounding box. As a further step, we designed and trained R-CNN and Fast R-CNN models for people detection with the same training datasets. We compared these R-CNN and Fast R-CNN architectures with the technique proposed in Sect. 4 using the same testing datasets of thermal images. To measure the efficiency of the proposed approach, the parameters on which the three architectures are evaluated include accuracy, precision, and recall values using confusion matrix criteria, see Eq. (6). Based on the results from these experiments, the new proposed detector showed good performance for people detection, social distancing classification on thermal images in both datasets, see Fig. 8. It achieved significant results with two datasets and overcomes R-CNN and Fast R-

CNN detectors see Table 2. YOLOv2 neural network looks the entire image at one time, unlike R-CNN and Fast R-CNN methods which see only the generated region proposals. Therefore, the proposed technique reduces the problem of background mistakes and improves the localization of detected persons in the image. In addition to that, the proposed approach shows better accuracy in comparison to other methodologies [28, 29], and [30], see Table 3. According to these results, the methodology proposed in Sect. 4 is a promising one for people detection and social distancing classification on thermal images.

Accuracy = TP + TNTP + FN + TN + FP,Precision = TPTP + FP,Recall = TPTP + FN,Accuracy = TP + TNTP + FN + TN + FP,Precision = TPTP + FP,Recall = TPTP + FN,
(6)

where TP stands for the number of true positive; TN stands for the number of true negative; FP stands for the number of false positive; FN stands for the number of false negative.

Fig. 8



Sample Images from **a**, **b** Dataset I, **c**, **d** Dataset II

Table 2 Performance of this work vs. other object detectors

Table 3 Performance of the proposed approach vs. other methodologies, averaging dataset I and II

Experiments were carried on a computer with Intel® Core TM I3-6006U CPU @ 2 GHz. MATLAB2020a was adopted with its built-in applications such as Ground Truth Labeler, Neural Network Designer. Jetsonnano was used as an embedded system test platform in Sect. 6.

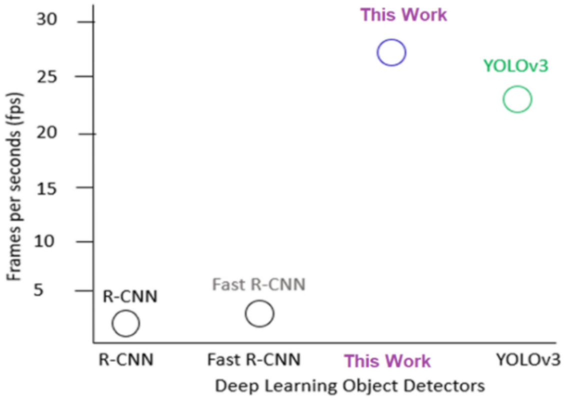
Real-time measurement of this work vs other object detectors

The main objective of this research is to detect and recognize individuals in real-time. We have to monitor and track people’s movements by utilizing a video camera. The invention and evolution of deep learning have improved the traditional ways of object detection and recognition systems. This technology is applied in several applications to identify and locate the objects in images, and it showed encouraging results for real-time detection [31]. To understand further, experiments were carried to compare the proposed approach and other deep learning detectors such as R-CNN and Fast R-CNN. MATLAB was used with our test bench of videos that were captured from a thermal camera. The three models run simultaneously while frames per second were calculating for each model. Based on results from this experiment, the neural network proposed in this work runs faster than the other two

Received 24 October 2020; Accepted 15 December 2020.

detectors (Fast R-CNN and R-CNN). Note that this work showed better results for real-time detection comparable to the method [32], which proposed YOLOv3 detector. It is observed that R-CNN and Fast R-CNN have low frames per second, which make them not suitable for real-time applications. Figure 9 shows the comparative real-time detection of this work versus other deep learning object detectors.

Fig. 9



The comparison of this work vs other competing deep learning detectors (R-CNN, Fast R-CNN, and YOLOv3) for real-time detection

Implementation of the proposed approach on embedded hardware

Jetsonnano (NVIDIA device)

Jetsonnano NVIDIA system is a low-cost embedded device but a powerful computer. It costs approximately $100 [33]. Jetsonnano can run various advanced neural networks including a full version of most popular deep learning (DL) and machine learning (ML) frameworks such as Pytorch, Caffe, Keras, and TensorFlow. This embedded device uses TensorRT accelerator libraries which include Jetpack packages. Jetsonnano is suitable for real-time applications in different scenarios and is capable to process multiple high-definition video streams.

Jetsonnano includes CPU QUAD-core ARM A57 at 1.43 GHz and GPU 128-core Maxwell. The memory of the device is 4 GB, 64-bit, LPDDR4 25.6 GB/s. Jetsonnano has a USB 2.0 Micro-B, 4 × USB 3.0. A standard camera module with 8 M pixel resolution has been used in our experiments. The camera was connected to the camera serial interface (CSI) in Jetsonnano. The trained neural network model and social distancing classification algorithm defined in Sect. 4 has been deployed in Jetsonnano and it runs as a standalone application.

MATLAB environments and third-party packages were utilized to generate the C code of the proposed approach in the NVIDIA device. GPU coder was used for converting MATLAB code into an optimized CUDA code. Compute unified device architecture or CUDA is an extension of C programming language which is designed for NVIDIA frameworks. Jetsonnano was connected to the host computer using an ethernet cable. MATLAB coder was utilized to generate C code to Jetsonnano. We used a parallel computing toolbox to solve complex computational and data processing using a multicore processor and GPU. A deep learning toolbox was utilized to provide a framework to implement the neural network and algorithms in Jetsonnano. GPU support package for NVIDIA is used to deploy the proposed algorithms in Jetsonnano. This support package application enables the communication remotely to the targeted NVIDIA hardware. Embedded coder was used for code generation on Jetsonnano. This tool improves the code generation on hardware effectively. A JetPack developer AI tool was installed in the NIVIDIA device. It is an environment variable application which is to be applicable for code generation of the proposed deep learning architecture in Jetsonnano. Microsoft visual studio 2019 was installed as a compiler generate GPU code in Jetsonnano. CUDA Deep Neural Network libraries were used to accelerate primitives for neural network architecture.

Test the proposed algorithm on Jetsonnano

The proposed algorithm was deployed in Jetsonnano and run as a standalone application to evaluate its performance. A Raspberry Pi camera model V2 was exposed to another personal computer that simulated a number of videos that were captured from a thermal camera. While the proposed algorithm was running in the NIVIDIA device, we recorded various parameters. We measured the average frames per second for the proposed approach on Jetsonnano and we compared the achived results with other different methods, see Table 4. According to the results from this experiment, our approach showed the best result for the real-time which reached up to 27 fps.

Table 4 The real-time measurement for the proposed approach vs other methods

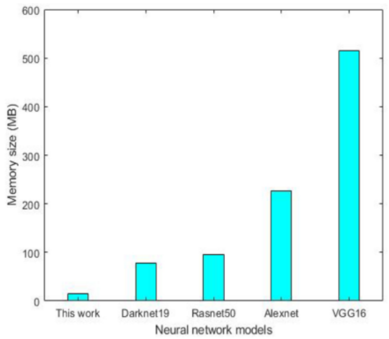
We measured also the power consumption for Jetsonnano. We removed all Jetsonnano accessories such as a mouse, monitor, and keyboard. We measured the power consumption at 1.24 W when the deployed algorithm is off. When the distance classification algorithm was executed, the power consumption was recorded at 4.4 W. Table 5 shows the power measurement of the NVIDIA device in different scenarios.

Table 5 Power consumption measurement in different scenarios

We recorded the measurement of Graphics Processing Unit (GPU) and the Central Processing Unit (CPU) % resource utilization in Jetson device. The GPU is designed to process the graphic operations and the CPU runs the operating system and applications. These characteristics are essential to assess its computation processing. The table shows the measured values for the GPU and CPU processors while the proposed algorithm was executed in the targeted hardware. Moreover, we measured the temperature of Jetsonnano while the proposed approach was in execution. The temperature was measured at 54.5 °C for GPU and 54.1 for CPU, see Table 6. Further to our experiments, we measured the memory size for the deployed algorithm in Jetsonnano, which is 14 MB. This is the advantage of the proposed approach in comparison to the other pre-trained models such as VGG16, Alexnet, and Resnet50 in method [29]. Figure 10 shows the comparison between the proposed approach with other pre-trained models in terms of memory size. These architectures use massive CNN layers which need a large disk size for the deployment on the targeted embedded system. This could affect real-time performance while the algorithm is running on low-cost embedded devices. This is the advantage of the proposed approach versus the pre-trained CNN models and it can be superior for real-time detection.

Table 6 The % resource utilization and temperature measurement for the GPU and CPU in Jetsonnano while our method is running on it

Fig. 10



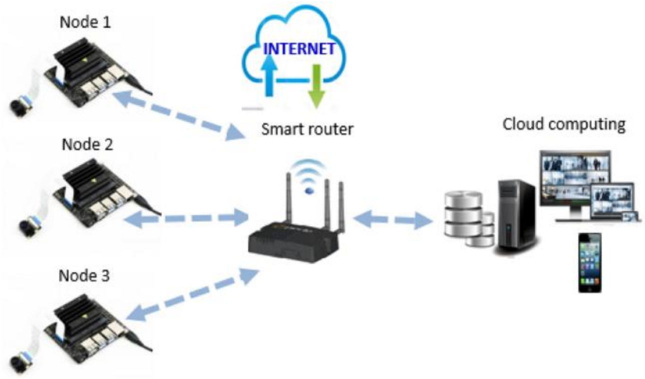
Comparison of the proposed approach verses other pre-trained models in terms of memory size

Distributed surveillance video system for social distancing

Video surveillance cameras are an effective monitoring system for authorities to visualize how people are acting and their compliance with social distancing. In our research, we implemented a distributed surveillance camera system based on embedded devices.

The proposed system is composed of multiple Jetsonnanodevices with, each combined with a video camera. Each camera is connected to one Jetsonnanodevice, which represents a smart node in the system architecture. Jetsonnanodevices were upgraded with Wi-Fi for internet connection. All Jetsonnanodevices were connected to the computer through a router using a static IP address from each node. The router directs the video streaming of each node and serves as a networking device to the centralized surveillance management system (personal computer), see Fig. 11.

Fig. 11



Smart Surveillance distributed video system for people detection and social distancing classification

MobaXterm application in Windows 10 was used to establish communication between the centralized surveillance management system (personal computer) and NvidiaJetsonnano nodes. The communication was established using an OpenSSH application with respect to the defined *IP address* of each node. Secure Shell or OpenSSH is a remote information communication technology protocol that allows users to control and transfer data between computers. The system is built with a multi-access point of *IP addresses* through OpenSSH sessions in the MobaXterm software. Each OpenSSH session communicates with Jetsonnano node through its defined *IP address*. The latency time was measured between the computer and Jetson boards at 0.3 ms.The proposed approach is suitable for a distributed surveillance system that can visualize people detection and social distancing classification on thermal images from several Jetsonnanodevices in one centralized surveillance management system.

Conclusion and future work

This analysis bestowed AN intelligent closed-circuit television for individuals following and social distancing classification supported thermal pictures. The planned technique achieved promising results for individuals detection in terms of analysis the accuracy and exactitude of the detector cherish the opposite deep learning models. a particular rule was enforced on bounding boxes to tell apart between safe and unsafe conditions, severally, marking as inexperienced and red the bounding box for detected persons. The planned technique showed higher results for period performance vs different object detectors. The planned approach is enforced in a very distributed video police work system; so, it's an acceptable resolution for the authorities to ascertain the compliance of individuals with social distancing and at identical time screening their vital sign. within the future, we are going to utilize this technique on mobile cameras, e.g., mounted on AN autonomous drone system, and thence drones area unit less complicated to work and more practical to capture quick actions of the detected objects from totally different angles. weare going to extend our analysis to use and experiment individuals detection by conjointly applying three-D dimensions to possess 3 parameters (x, y, z), within which we will understand uniform distribution distance within the entire image and eliminating the angle impact. additionally to it, the fresh free YOLOv4 detector [37] are going to be conjointly thought of.

References

1. M. Cristani, A. D. Bue, V. Murino, F. Setti and A. Vinciarelli, "TheVisual Social Distancing Problem," in IEEE Access, vol. 8, pp. 126876-126886, 2020, doi: 10.1109/ACCESS.2020.3008370.
2. A. H. Ahamad, N. Zaini and M. F. A. Latip, "Person Detection for Social Distancing and Safety Violation Alert based on Segmented ROI," 2020 10th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), Penang, Malaysia, 2020, pp. 113-118, doi: 10.1109/ICCSCE50387.2020.9204934.
3. C. T. Nguyen et al., "A Comprehensive Survey of Enabling and Emerging Technologies for Social Distancing—Part I: Fundamentals and Enabling Technologies," in IEEE Access, vol. 8, pp. 153479-153507, 2020, doi: 10.1109/ACCESS.2020.3018140.
4. S. Sharma, M. Bhatt and P. Sharma, "Face Recognition System Using Machine Learning Algorithm," 2020 5th International Conference on Communication and Electronics Systems (ICCES), COIMBATORE, India, 2020, pp. 1162-1168, doi: 10.1109/ICCES48766.2020.9137850.
5. D. Garg, P. Goel, S. Pandya, A. Ganatra and K. Kotecha, "A Deep Learning Approach for Face Detection using YOLO," 2018 IEEE Punecon, Pune, India, 2018, pp. 1-4, doi: 10.1109/PUNECON.2018.8745376.
6. H. Filali, J. Riffi, A. M. Mahraz and H. Tairi, "Multiple face detection based on machine learning," 2018 International Conference on Intelligent Systems and Computer Vision (ISCV), Fez, 2018, pp. 1-8, doi: 10.1109/ISACV.2018.8354058.
7. M. Matsumoto, K. Kaneta, M. Naruoka, H. Tanaka and K. Takano, "Tracking Positions of Human Body Parts Based on Distance Measurement with Sound Wave," 2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA), Taipei, 2017, pp. 514-518, doi: 10.1109/WAINA.2017.135.
8. M. Hatt, C. Parmar, J. Qi and I. El Naqa, "Machine (Deep) Learning Methods for Image Processing and Radiomics," in IEEE Transactions on Radiation and Plasma Medical Sciences, vol. 3, no. 2, pp. 104-108, March 2019, doi: 10.1109/TRPMS.2019.2899538.
9. M. Sewak, S. K. Sahay and H. Rathore, "Comparison of Deep Learning and the Classical Machine Learning Algorithm for the Malware Detection," 2018 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), Busan, 2018, pp. 293-296, doi: 10.1109/SNPD.2018.8441123.
10. P. P. Shinde and S. Shah, "A Review of Machine Learning and Deep Learning Applications," 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 2018, pp.1-6, doi: 10.1109/ICCUBEA.2018.8697857.
11. O. Jukić, I. Špeh and I. Heđi, "Cloud-based services for the Internet of Things," 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, 2018, pp. 0372-0377, doi: 10.23919/MIPRO.2018.8400071.
12. W. Tärneberg, V. Chandrasekaran and M. Humphrey, "Experiences Creating a Framework for Smart Traffic Control Using AWS IOT," 2016 IEEE/ACM 9th

International Conference on Utility and Cloud Computing (UCC), Shanghai, 2016, pp. 63-69.

13. M. Aleisa, A. A. Hussein, F. Alsubaei and F. T. Sheldon, "Performance Analysis of Two Cloud-Based IoT Implementations: Empirical Study," 2020 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2020 6th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), New York, NY, USA, 2020, pp. 276-280, doi: 10.1109/CSCloud-EdgeCom49738.2020.00055.
14. Sohan, M.F. (2020). So You Need Datasets for Your COVID-19 Detection Research Using Machine Learning?ArXiv, abs/2008.05906
15. Vaid, S., Kalantar, R. &Bhandari, M. Deep learning COVID-19 detection bias: accuracy through artificial intelligence. International Orthopaedics (SICOT) 44, 1539–1542 (2020). https://doi.org/10.1007/s00264-020-04609-7.
16. D. Veeraiah and J. N. Rao, "An Efficient Data Duplication System based on Hadoop Distributed File System," 2020 International Conference on Inventive Computation Technologies (ICICT), 2020, pp. 197-200, doi: 10.1109/ICICT48043.2020.9112567.
17. Rao, J. Nageswara, and M. Ramesh. "A Review on Data Mining & Big Data." *Machine Learning Techniques. Int. J. Recent Technol. Eng* 7 (2019): 914-916.
18. Karthik, A., MazherIqbal, J.L. Efficient Speech Enhancement Using Recurrent Convolution Encoder and Decoder. Wireless PersCommun (2021). https://doi.org/10.1007/s11277-021-08313-6
19. S. N. Ajani and S. Y. Amdani, "Probabilistic path planning using current obstacle position in static environment," 2nd International Conference on Data, Engineering and Applications (IDEA), 2020, pp. 1-6, doi: 10.1109/IDEA49133.2020.9170727.
20. S. Ajani and M. Wanjari, "An Efficient Approach for Clustering Uncertain Data Mining Based on Hash Indexing and Voronoi Clustering," 2013 5th International Conference and Computational Intelligence and Communication Networks, 2013, pp. 486-490, doi: 10.1109/CICN.2013.106.