Multimedia Tools and Applications (2023) 82:23569–23593
https://doi.org/10.1007/s11042-022-14251-7

COVID-19 risk reduce based YOLOv4-P6-FaceMask detector and DeepSORT tracker

Mohammed Lakhdar Mokeddem[1] · Mebarka Belahcene[1] · Salah Bourennane[2]

Received: 26 March 2022 / Revised: 30 May 2022 / Accepted: 4 November 2022 /
Published online: 25 November 2022

Abstract

Wearing masks in public areas is one of the effective protection methods for people. Although it is essential to wear the facemask correctly, there are few research studies about facemask detection and tracking based on image processing. In this work, we propose a new high performance two stage facemask detector and tracker with a monocular camera and a deep learning based framework for automating the task of facemask detection and tracking using video sequences. Furthermore, we propose a novel facemask detection dataset consisting of 18,000 images with more than 30,000 tight bounding boxes and annotations for three different class labels namely respectively: face masked/incorrectly masked/no masked. We based on Scaled-You Only Look Once (Scaled-YOLOv4) object detection model to train the YOLOv4-P6-FaceMask detector and Simple Online and Real-time Tracking with a deep association metric (DeepSORT) approach to tracking faces. We suggest using DeepSORT to track faces by ID assignment to save faces only once and create a database of no masked faces. YOLOv4-P6-FaceMask is a model with high accuracy that achieves 93% mean average precision, 92% mean average recall and the real-time speed of 35 fps on single GPU Tesla-T4 graphic card on our proposed dataset. To demonstrate the performance of the proposed model, we compare the detection and tracking results with other popular state-of-the-art models of facemask detection and tracking.

Keywords Detection · Localization · Deep learning · Scaled-YOLOv4 · Tracking

✉ Mohammed Lakhdar Mokeddem
mohammedlakhdar.mokeddem@univ–biskra.dz

Mebarka Belahcene
mebarka.belahcene@univ-biskra.dz

Salah Bourennane
salah.bourennane@fresnel.fr

[1] RB_IAIM, LI3C, M.Khider University, Biskra, Algeria

[2] GSM, Fresnel Institut, Ecole Centrale, Marseille, France

1 Introduction

The report n.48 of the World Health Organization (WHO) noted that COVID-19 disease 2019 has globally infected over 58 million people and caused over 1.4 million deaths (9 April 2021). With this outbreak of COVID-19 coronavirus, many countries, or we can say that all nations, were obliged to commence new rules for social distance and face mask-wearing. The governments have obliged hospitals and different organizations to use new infection interference measures to prevent the spreading of COVID-19 because its transmission rate is increasing. However, the transmission rate could vary per the government's measures and policies. As COVID-19 is transmitted through airdrops and shut contact, governments have started using new rules forcing individuals to prevent people from sitting close to each other and wearing a face mask to scale back the transmission and spreading rate. New variants of the coronavirus took hold after the relaxation of many countries in adhering to safety rules (India, Nigeria, UK, Brazilian …), which made the WHO recommend the usage of Personal Protective Equipment (PPE) among people and in medical care. The coronavirus (COVID-19) spreads quickly in close contact and in crowded environments. The spread of COVID-19 affected people's lives and disrupted the economy. It became classified as significant public health and economic problem. Countries need guidance and surveillance of people in crowded environments and public areas incredibly packed to ensure that wearing face masks laws are applied. This could be used through video surveillance systems and deep learning (DL) models. However, most mask detection applications and current research of mask detection models target solving the masked face and no masked face detection problem but ignore wearing face mask incorrectly.

The face mask is the focus of this work to minimize the transmission and spreading of COVID-19.

Our main objective in this work is:

1. Detection and tracking of masked/incorrectly masked/no masked faces;
2. Tracking faces;
3. Save unmasked and incorrectly masked faces for building person risk dataset. This dataset is destined to identification/authentification and other applications.

The major contributions of this paper consisted in:

1. A general facemask detection and tracking;
2. Collection of a new masked/incorrectly masked/no masked faces dataset in video of an uncontrolled environment;
3. Propose a new model of face mask detection, namely the YOLOv4-P6-FaceMask detector;
4. Propose to use Simple Online and Real-time Tracking (Deep SORT) to track faces by ID assignment to save faces only once per person in a file;
5. Create a risk person database (no masked faces) for future identification/authentification or other applications;
6. The proposed approach is applied to multiple indoor and outdoor sequences in an uncontrolled environment.

The outcome of the proposed detection model of the masked faces /incorrectly masked/no masked face region in the image or video surveillance. This region is considered as an input for the DeepSORT tracker. The outcome of the Simple Online and Real-time tracker is also an image or sequence video, but with ID identification for every face. In the final step and after every tracking of the box, we save the image of the unmasked face only once (Fig. 2).

The superiority of proposed method proved on performance metrics mean average precision (**mAP**) and mean average recall (**mAR**) then the detection and tracking results are compared with recent works of facemask detection and tracking. This work is evaluated on proposed facemask dataset and public videos/images.

Detection of the masked face sequences and the results (cropped images) are evaluated on the video sequences of persons with masked/ incorrectly masked /no masked faces.

The remainder of this paper is organized as follows: Section 1 introduction and Section 2 exposes the most original recent works. Section 3 is devoted to the presentation and study of the proposed approach. The experimental results and discussions are presented in section 4 finally the work ends with a conclusion in section 5.

1.1 Related work

1.1.1 Recent detection and localization methods based on DL

We based on two survey papers focus on deep learning approaches for object detection and datasets, metrics, and fundamentals, which can be found in [60**,** 62]. For face detection, we based on [4**,** 5**,** 16]. State-of-the-art object detectors use deep learning approaches, usually divided into two categories (Fig. 1). The first category is called two-stage detectors, of which the famous models are RCNN (Recurrent Convolutional Neural Network) [21], Fast RCNN [19], Faster RCNN [44], which starts with Region Proposal Network (RPN) to generate regions of interests and then performs the classification and bounding box regression. The second category is called one-stage detectors, of which the famous models are YOLO (You Only Look Once) [43], YOLOv2 [41], YOLOV3 [8], YOLOv4 [53], Scaled-YOLOv4 Scaling: Cross Stage Partial (CSP) [32], Single Shot Multibox Detector (SSD) [30], RetinaNet [48], and EfficientDet [17]. The most popular one stage model is YOLO; Fig. 1 shows the timeline and the comparison between members of the YOLO family models and their performance. The evaluation of models was usually based on two datasets, Pascal VOC [17] and MS COCO (Microsoft Common Objects in Context) [29], the results are given in Table 1.

The usual object detector model is contained of several blocks:

- **Input block:** Image or Patches or Image Pyramid
- **Backbones block:** HematoNet [49], DaViT-G [14], EfficientNetV2 [47], CSPDarkNet53 [50], ConvNeXt-XL [33]
- **Neck block:** Additional blocks: SPP [23], Deeplab [9], SCPSPP, ASPP [31], RFB [31]. Path-aggregation blocks: FPN, BiFPN, PAN [20], BiFPN [10].
- **Heads:**

Dense Prediction (one-stage): RPN, SSD, YOLO, RetinaNet (anchor based) CornerNet [55].

Sparse Prediction (two-stage): Fast Faster R-CNN, R-CNN, R-FCN.

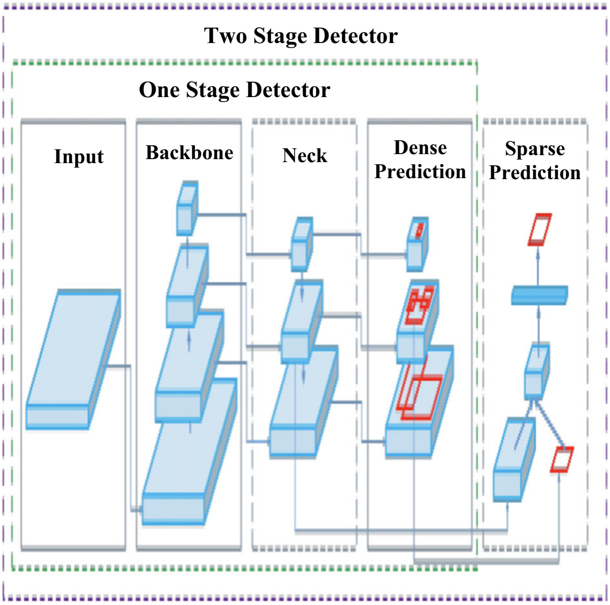


Fig. 1 Overall structure of one and two stages object detectors

1.2 Recent face detection methods

CNN based models show big progress in recent years. B. Yang et al. in [57] proposed a face detection, which adopted the feature aggregation model based on Convolutional Neural Network (CNN) to extracted features. Yang et al. [58] proposed a DL approach, from facial parts responses to face detection. Recently, Zhu et al. [61] proposed contextual multi-scale region-based CNN (MS RCNN) for unconstrained face detection, which considered contextual information. Luo et al. [35] proposed Small Faces Attention (SFA) detector, this proposed work is a multiple-branches framework to focus on the detection of small faces (accurate detection). LI Xiaochao et al. [28] proposed face detection based on receptive field enhanced multi-task cascaded CNN.

Table 1 Object detection models and their accuracies

Dataset	VOC [8]	COCO [53]	
Model Name	mAP	FPS	mAP
RCNN [21]	0.53	0.5	–
Fast RCNN [19]	0.68	7	0.19
Faster RCNN [44]	0.70	19	0.22
SSD [32]	0.75	45	0.27
YOLOv2 [41]	0.73	67	0.22
YOLOv3 [42]	0.75	47	0.33
YOLOv4 [8]	0.79	62	0.43
RetinaNet [30]	–	–	0.415
YOLOv4-CSP [53]	–	–	0.462
EfficientDet-D0 [48]	–	–	0.346

1.3 Recent DL methods COVID-19 facemask

In recent years, the majority of the publications center around face construction and identity recognition when wearing face-mask. In this paper, our emphasis is on recognizing individuals who are not wearing facemask and tracking all faces to save every no masked face once only to help in diminishing the transmission and spreading of the COVID-19.

Qin B et al. [40] have designed a method for identifying facemask-wearing conditions using image super-resolution with a classification network to prevent COVID-19. This proposed method achieved an accuracy of 98.7% in classifying pictures into three categories: the first is correctly facemask wearing, the second is incorrectly facemask-wearing, and the last is not wearing a facemask, but they used a small dataset named public dataset Medical Masks Dataset containing 3835 images in the stage of training of the model.

Paper [15] is proposed by Ejaz MS et al. They applied the PCA [6] to know the masked and unmasked faces. This technique is effective for face recognition without a mask with an accuracy of 96.25%, but its accuracy is decreased to 68.75% in face recognition using the mask.

Chowdary [11] proposed a facemask detection model using Transfer Learning (TL) of InceptionV3. This approach achieves an accuracy of 99.9% during training and 100% during testing, but the model is trained and tested on a small dataset, namely the SMFD [3] facemask dataset that contained 1570 images. Jiang M et al. [24] proposed the RetinaMask, which is based on the RetinaNet model's architecture. The authors of this work proposed a new dataset for mask detection made up of the Wider Face dataset and the masked faces dataset (MAFA). The model used ResNet as a standard backbone and FPN as a neck. Kumar. A et al. [27] proposed scaling up face mask detection with YOLO on a novel dataset. The authors present a novel dataset for facemasks detection consisting of 52,635 images in this work. Further, the proposed dataset was tested with the YOLOv3 and YOLOV4 detection models.

X Jiang et al. [25] proposed a new dataset, namely Properly Wearing Masked Face Detection Dataset (PWMFD), which included 9205 images, and proposed Squeeze and Excitation SE-YOLOv3 detection model. PeishuWu et al. FMD-Yolo [56] proposed an efficient facemask detection technique for COVID-19 counteraction and control out in the open area. In this work, the authors proposed a new facemask detection framework. The model FMD-YOLO is proposed to monitor whether people wear masks in a right way in public, which is compelling method for obstructing the infection transmission.

Specifically, the feature extractor utilizes Im-Res2Net-101 which consolidates Res2Net module and deep residual network.

Prasad et al. [39] proposed a YOLOv4 deep learning model and a deep transfer learning approach was employed to develop a real time facemask detector. Google Collab was utilized to run the simulations and make judgments. For facemask detection, the performance parameters were calculated, and an average precision of 0.86 was obtained, with an F1 score of 0.77 for the picture dataset and 90% accuracy for the video dataset. It is also a real time facemask detector that can correctly identify a person with and without a face mask.

Table 2 State-of-art facemask framework based deep learning (***Dm***: *detection model* ***Tr***: *tracking model* ***Imd***: *incorrectly mask detection* ***S_ no_ MF***: *Save no Masked faces)*

Model	*Dm*	*Tr*	*Imd*	*S_ no_ MF*
YOLOv2+ResNet50 [34]	YOLOv2	no	no	no
FaceMask SRCNet [X]	SRCNet	no	yes	no
SSDMNv2 [38]	SSD+MobileNetv2	no	no	no
RetinaFaceMask [18]	RetinaNet	no	yes	no
Goyal FaceMask [22]	Custom model	no	yes	no
Prasad YOLOv4 FaceMask [39]	YOLOv4	no	no	no

1.4 Proposed YOLOv4-P6-FaceMask detection and DeepSORT tracking

1.4.1 YOLO detection models

You Only Look Once (YOLO) is a highly accurate, real time one-stage object detection algorithm. It is developed to create a one-step process that involves detection and localization. The.

bounding box and class prediction are performed after an evaluation of the input image. The structure of the model (an example of structure) is shown in Fig. 3 and consists of Backbone, Neck, and Prediction. The fastest YOLO architecture can reach 96 Frame per Second (FPS), while the smaller version, the tiny YOLO, can reach up to 244 FPS on a GPU-equipped computer. YOLO'S idea is different from other traditional systems: the bounding box prediction and prediction category are done at the same time. The Table 2 illustrates a state of the art of the different facemask frameworks based deep learning.

YOLOv4 combines the characteristics of YOLOv1, YOLOv2, YOLOv3, etc., and reaches the current best in terms of detection speed and detection accuracy compensation.

Combining the characteristics of the ResNet structure, YOLOv3 integrates the residual module into itself and obtains Darknet53. On this basis, considering the superior learning ability of Cross Stage Partial Network (CSP-Net) [51], YOLOv4 constructed CSPDarkNet53 [51], in the residual module (input the feature layer and output the top-level feature information). As shown in Fig. 2, image input is divided into grids, and then a B bound box is defined for every grid cell, each with a confidence score.

Here, reliability represents the probability that an object will exist in each bounding box and is defined as:

$$C_S = P_r \times IOU \tag{1}$$

Were IOU (Intersection Over Union) is a fraction between zero and one, and Average Precision (AP):

$$AP = \sum_{k=1}^{n} P(k)\Delta r(k) \tag{2}$$

Where k is the precision at threshold k and $\Delta r(k)$ is the change in recall.

The DenseNet architecture, which takes the prior input and concatenates it with the current input before advancing into the dense layer, inspired the Cross Stage Partial architecture.

A dense block and a transition layer are present in each stage layer of a DenseNet, and each dense block is made up of k dense layers. The output of the ith dense layer will be concatenated

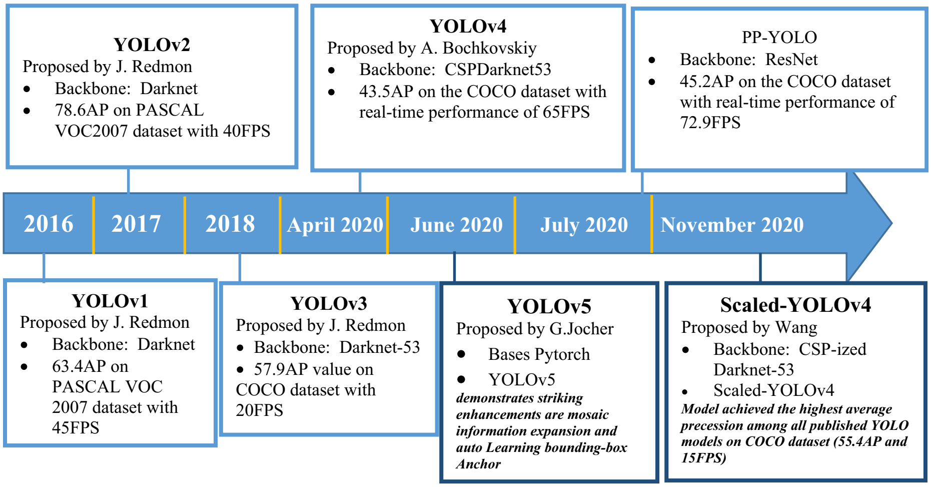

YOLOv2
Proposed by J. Redmon
Backbone: Darknet
78.6AP on PASCAL VOC2007 dataset with 40FPS
YOLOv4
Proposed by A. Bochkovskiy
Backbone: CSPDarknet53
43.5AP on the COCO dataset with real-time performance of 65FPS
PP-YOLO
Backbone: ResNet
45.2AP on the COCO dataset with real-time performance of 72.9FPS
2016
2017
2018
April 2020
June 2020
July 2020
November 2020
YOLOv1
Proposed by J. Redmon
Backbone: Darknet
63.4AP on PASCAL VOC 2007 dataset with 45FPS
YOLOv3
Proposed by J. Redmon
Backbone: Darknet-53
57.9AP value on COCO dataset with 20FPS
YOLOv5
Proposed by G.Jocher
Bases Pytorch
YOLOv5
demonstrates striking enhancements are mosaic information expansion and auto Learning bounding-box Anchor
Scaled-YOLOv4
Proposed by Wang
Backbone: CSP-ized Darknet-53
Scaled-YOLOv4
Model achieved the highest average precession among all published YOLO models on COCO dataset (55.4AP and 15FPS)

Fig. 2 Different YOLO models and their average precision

with the input of the ith dense layer, with the result being the (i + 1) dense layer’s input. The equations illustrating the aforementioned mechanism are as follows:

$$x_1 = w_1 {}^* x_0 \tag{3}$$

$$x_2 = w_2 {}^* [x_0, x_1] \tag{4}$$

$$\vdots$$

$$x_k = w_k {}^* [x_0, x_1, \ldots, x_{k-1}] \tag{5}$$

Where ∗ is the convolution operator, and [×0, ×1, …] means to concatenate ×0, ×1, …, and w_i and x_i the ith dense layer’s weights and outputs, respectively.

The CSP [51] is based on the same principle, except that instead of concatenating the ith output with the ith input, we divided the input ith into two-parts, x_0’ and x_0”, with one part passing through the dense layer x_0’ and the second part x_0” being concatenated at the end with the result at the dense layer’s output.

This is mathematically equivalent to the following equation:

$$x_k = w_k {}^* [x_0, x_1, \ldots, x_{k-1}] \tag{6}$$

$$x_T = w_T {}^* [x_0, x_1, \ldots, x_k] \tag{7}$$

$$x_U = w_U {}^* [x_0, x_1, \ldots, x_T] \tag{8}$$

This will result in different dense layers repeatedly learn copied gradient information.

1.5 YOLOv4 scaling

In traditional detection models, scaling means modifying the model's depth by adding more convolutional layers. For example, the VGGNet scaled to VGG11, VGG13, VGG16, and VGG19 architectures. But now, the scaling approach modifies the depth, width, resolution, and structure of the network, forming a scaled model, for example, Scaled YOLOv4. To prove the superiority of the selected model YOLOv4-P6 in terms of backbone, accuracy, real time and performance, in this paper we compare it with Fast-RCNN, Faster-RCNN, YOLOv3, YOLOv4, which are the state-of-the-art pedestrian detection models. In the proposed approach, illustrated in Fig. 2, we use the Scaled YOLOv4 detection technique to detect faces in single pictures, real-time video or online cameras.

This paper will not discuss the history or background of previous versions of YOLO (YOLOv1, YOLOv2, and YOLOv3). We trained a custom YOLOv4-P6 model for facemask detection and localization by using an extensive dataset, which consisted of about 18,000 images belonging to three classes: "masked," "incorrectly mask," and "unmasked (no-masked)". The proposed dataset is collected from:

- Wider face dataset [59] (all no masked faces and part of masked faces).
- FMD (face-mask dataset) [1] (masked-faces and incorrect-masked faces).
- RMFD (real-face-mask dataset) [52] (masked-faces and incorrectly-mask faces).

After, we take the face set detected by YOLOv4-P6-FaceMask, such as an input of the Deep SORT tracker. The Deep SORT creates a unique identification number ID for each initial.

detection. If this initial detection is an unmasked face or incorrectly mask wear, we crop the face and save it in a file using OpenCV. Then Deep SORT tracks each of the faces as they move around frames in a video, maintaining the assignment of a unique ID.

1.6 COVID-19 YOLOv4-P6-FaceMask detector

The CSPNet [51] proposed by Wang, can be applied to different CNN designs, while lessening how much boundaries and calculations. What's more, it additionally further develops exactness and decreases induction time.

The proposed facemask detector is illustrated in Fig. 3 with the re-design of YOLOv4 to YOLOv4-CSP to get the best speed/accuracy trade-off Network Architecture.

The Network Architecture of YOLOv4 detector used CSPDarknet53 as a backbone and YOLOv3 as a heads and SPP and PAN as a neck.

To achieve real-time facemask detection model we talk about the architecture scaling. To increase the performance of YOLO v4, we make some modifications in the architecture.

The network architecture of YOLOv4-P6-FaceMask Detector is illustrated in Fig. 4 and Table 3. The convolution layer is responsible of extracting features from the input using kernel (conv filter).

Backbone: The Backbone blocks of YOLOv4-P6-FaceMask can be shared into two parts: the kernel block (convolution building block) and the CSP-Block modules (see Table 3). The number of residual layer owned by each stage in CSP-Block is 1_2_8_8_4 respectively This means that (**1**x CSP-Block _ **2**x CSP-Block _ **8**xCSP-Block_ **8**xCSP-Block_**4**xCSP-Block).

The CSP-stage number one is converted to original-Darknet-Residual Layer.

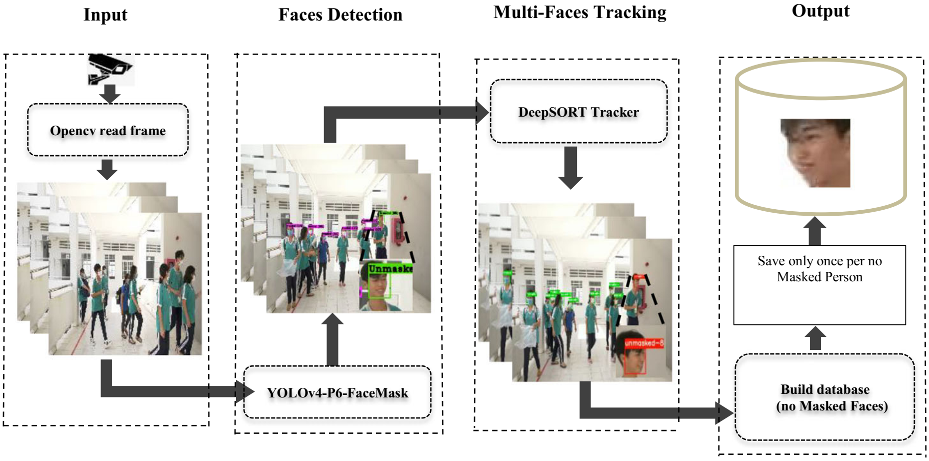


Fig. 3 Overall structure of proposed face mask detection and tracking system

Neck: The **PAN** architecture is CSP-ized (covert to Cross-stage-partial connections form) and Mish function is used to reduce computation, **SPP**: It was originally inserted in the middle position of the neck, the same idea is borrowed and implemented in CSPPAN too.

Mish function As for the backbone in YOLOv4-P6-FaceMask, we used the mish function as an activation function, instead of the activation function. The results presented by Misra D [37] in his research paper show that the Mish (Eq. (9)) activation function converged to the minimal loss faster and with higher accuracy than Swish and ReLU. The outcome was constant, especially when parameter initializers were varied, regularization methods were used, and the learning rate was decreased.

Mish:

$$softmaxplus(x) = \ln(1 + e^x) \tag{9}$$

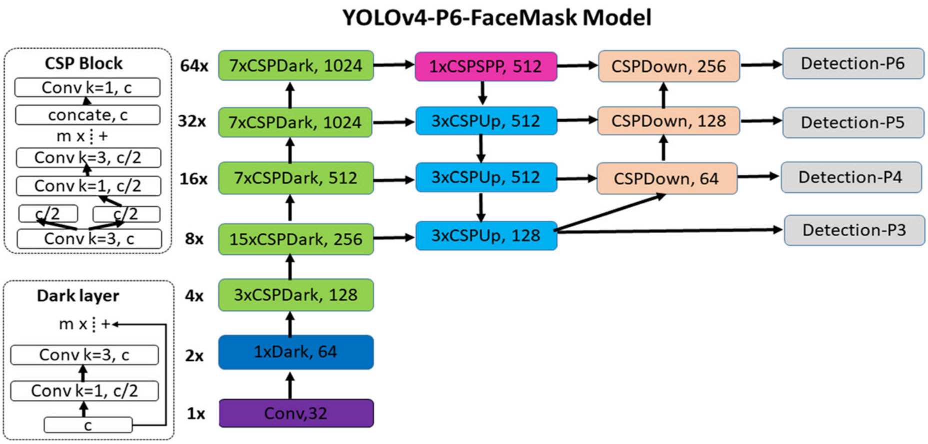


Fig. 4 Network architecture of YOLOv4-P6-FaceMask Detection Model

Table 3 Architecture of YOLOv4-P6-FaceMask (anchors, backbone, head and detect)

1: parameters	2: anchors	3: backbone
number of classes: 3 depth_multiple: 1.0 width_multiple: 1.0	- [14, 18, 25, 26, 32, 46, 52, 63] - [61,45, 48,102, 119,96, 97,189] - [97,189, 217,184, 171,384, 324,451] - [324,451, 545,357, 616,618, 1024,1024]	# [from, number, module, args] [[−1, 1, Conv, [1, 3, 33]], # 0 [−1, 1, Conv, [64, 3, 2]], # 1-P1/2 [−1, 1, BottleneckCSP, [64]], [−1, 1, Conv, [128, 3, 2]], # 3-P2/4 [−1, 3, BottleneckCSP, [128]], [−1, 1, Conv, [256, 3, 2]], # 5-P3/8 [−1, 15, BottleneckCSP, [256]], [−1, 1, Conv, [512, 3, 2]], # 7-P4/16 [−1, 15, BottleneckCSP, [512]], [−1, 1, Conv, [1024, 3, 2]], # 9-P5/32 [−1, 7, BottleneckCSP, [1024]], [−1, 1, Conv, [1024, 3, 2]], # 11-P6/64 [−1, 7, BottleneckCSP, [1024]], # 12
4: head	**5: detect**	
[[−1, 1, SPPCSP, [512]], [−1, 1, Conv, [512, 1, 1]], [−1, 1, nn.Upsample, [None, 2, ‘nearest’]], [−6, 1, Conv, [512, 1, 1]], [[−1, −2], 1, Concat, [1]], [−1, 3, BottleneckCSP2, [512]], [−1, 1, Conv, [256, 1, 1]], [−1, 1, nn.Upsample, [None, 2, ‘nearest’]], [−13, 1, Conv, [256, 1, 1]], [[−1, −2], 1, Concat, [1]], [−1, 3, BottleneckCSP2, [256]], [−1, 1, Conv, [128, 1, 1]], [−1, 1, nn.Upsample, [None, 2, ‘nearest’]], [−20, 1, Conv, [128, 1, 1]], [[−1, −2], 1, Concat, [1]], [−1, 3, BottleneckCSP2, [128]], [−1, 1, Conv, [256, 3, 1]],[−2, 1, Conv, [256, 3, 2]], [[−1, 23], 1, Concat, [1]], [−1, 3, BottleneckCSP2, [256]], [−1, 1, Conv, [512, 3, 1]],[−2, 1, Conv, [512, 3, 2]], [[−1, 18], 1, Concat, [1]], [−1, 3, BottleneckCSP2, [512]], [−1, 1, Conv, [1024, 3, 1]],[−2, 1, Conv, [512, 3, 2]], [[−1, 13], 1, Concat, [1]], [−1, 3, BottleneckCSP2, [512]], [−1, 1, Conv, [1024, 3, 1]],	[[30, 34, 38, 42], 1, Detect, [nc, anchors]], Detect(P6)	

$$f(x) = x.\tanh\Big(softmaxplus(x) \tag{10}$$

In the implementation process:

x: the input data.

The derivations is:

$$f^{'}(x) = \frac{e^{x}w}{\delta^{2}} \tag{11}$$

as a self-regularized non monotonic activation-function, where:

$$w = 4(x+1) + 4e^{2x} + e^{3x} + (4x+6) \tag{12}$$

1.7 Object tracking

The problem of Multi Object Tracking (MOT) consists in following the trajectory of different objects in a sequence, usually a video. With the rise of DL in the latest years, the algorithms that give a settling to this problem have benefited from the representational power of DL models. We based on survey paper [13] focus on MOT based DL approaches and state of the art MOTs use DL approaches.

We assess the accuracy and performance of DeepSORT [54] on the MOT-16 dataset [36]. This dataset evaluates tracking performance on seven challenging test video sequences, including frontal-view scenes with moving cameras as well as top-down surveillance setups.

The tracker is compared with state of art of tracking methods (see Table 4):

- AMIR [45]: Sadeghian et al. proposed a racking the untrackable: Learning to track multiple cues with long-term dependencies.
- IA [12]: Peng Chu et al. proposed an online MOT with Instance-Aware Tracker and Dynamic Model Refreshment.
- SORT (the Simple Online and Real-time Tracking) technique [7]: A. Bewley et al. proposed a Simple online and Real-Time Tracking.
- EAMTT [38]: Sanchez-Matilla, et al. proposed an online Multi-Target Tracking with strong and weak detections.

Table 4 Comparison of state-of- art Trackers (**MOT**: Mot16 dataset [27], **MT**: Mostly Tracked, **ML**: Mostly lost, **ID**: Identification number, **Acc**: Accuracy, **Pr**: Precision)

Tracking Model	MOT Acc	MOT Pr	MT %	ML %	IDs
ESNN [26]	33.4	72.1	11.7	30.9	1598
AMIR [45]	47.2	75.8	14	41.6	774
IA [12]	48.8	75.7	15.8	38.1	906
SORT [7]	59.8	79.6	25.4	22.7	1423
DeepSORT [54]	**61.4**	**79.1**	**32.8**	**18.2.**	**781**
EAMTT [46]	52.5	78.8	19	34.9	910

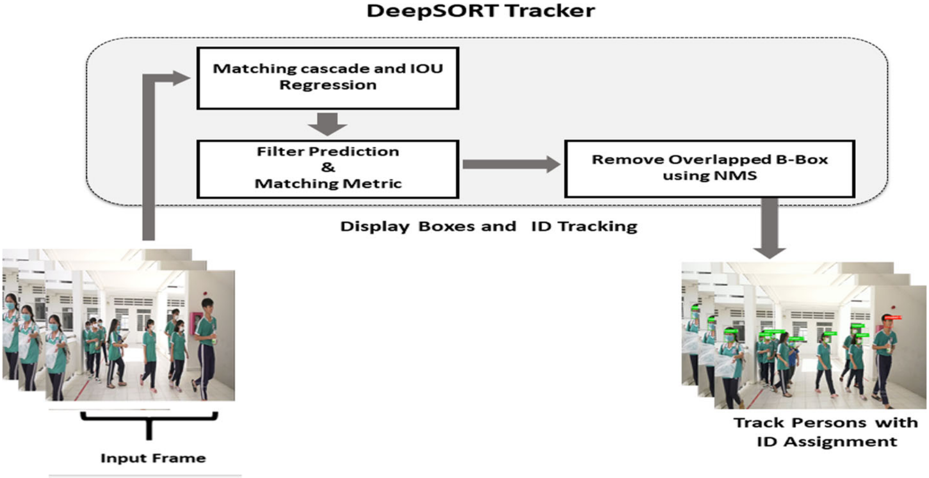

DeepSORT Tracker
Matching cascade and IOU Regression
Filter Prediction & Matching Metric
Remove Overlapped B-Box using NMS
Display Boxes and ID Tracking
Input Frame
Track Persons with ID Assignment

Fig. 5 DeepSORT Face Mask Tracking

1.7.1 Faces tracking

We track faces and ID assignments for each box in this phase using the DeepSORT technique (Fig. 5). DeepSORT is an online algorithm for the track of objects that considers both the information about the manifestation of the tracked objects and the bounding box parameters of the detection results to associate the detections in the frame at time t + 1 with tracked objects at a time t. Therefore, DeepSORT does not need to process the whole video at once. It only considers information about the current and previous frames to make predictions about the current frame. At the beginning of the sequence, i.e., in frame number one, the algorithm is assigned to each bounding box representing a pedestrian with a higher confidence value than a set threshold. The Hungarian algorithm is a combinatorial optimization algorithm used to assign the detections in a new frame to existing tracks so that the assignment cost function reaches the global minimum.

The cost function involves the Mahalanobis spatial distance $d^{(1)}(i, j)$ of the detected bounding box from the position predicted according to the known position at time t of that object, and a visual distance $d^{(2)}(i, j)$ that considers the appearance of the detected object and the history of the appearance of the tracked object. The expression of Mahalanobis $d^{(1)}(i, j)$ is given by:

$$d^{(1)}(i,j) = \left(d_j, y_i\right)^T S_i^{-1}\left(d_j - y_i\right) \tag{13}$$

λ: is a parameter that can be set to determine the influence of the visual distance $\boldsymbol{d^{(2)}(i, j)}$ and the Mahalanobis $\boldsymbol{d^{(1)}(i, j)}$. The cost function $\boldsymbol{c_{i,j}}$ of assigning a detected object j to a track i is given by the expression:

$$c_{i,j} = \gamma d^{(1)}(i,j) + (1-\gamma) d^{(2)}(i,j) \tag{14}$$

Where $\boldsymbol{y_i}$ represent the mean and Si represent the covariance matrix bounding box observations for the ith track $\boldsymbol{d_j}$ represents the jth detected bounding box.

The expression of visual $\boldsymbol{d^{(2)}(i, j)}$ that relies on appearance.

feature descriptors:

$$d^{(2)}(i,j) = min\left\{1-r_J^T r_k^{(i)} \middle| r_k^{(i)} \epsilon \Re\right\} \tag{15}$$

Where $\boldsymbol{r_j}$ is the appearance descriptor extracted from the part of the image within the j^th^ detected bounding box; $\mathscr{R}_i$ is the set of last 100 appearance descriptors $r_k^{(i)}$ associated with the track i.

The cosine distance uses by $\boldsymbol{d^{(2)}(i,\ j)}$ measure between the j^th^ detection and i^th^ track in the current detection to select the track where visually the most similar detection is.

previously found.

New track IDs are generated whenever:

- There are more detections in a frame than already tracked persons
- Detection cannot be assigned to any track, because the detection is too far from any track, or not visually similar to any previous detection.

1.8 Faces extraction

After detecting and tracking faces, we crop and save every person's face breaching face mask wear norms (red boxes). Faces are identified by the tracking ID number and frame number in the sequence. For more details, see Algorithm 1.

Algorithm 1. Saving of unmasked faces and incorrectly masked faces

```
Result: cropped and save unmasked/incorrectly masked faces
Input: bbox (ID, x, y, w, h), i_f_s = List to save ID of
unmasked/incorrectly mask faces.
for each box in bbox
    convert ID to int
    if ID not in i_f_s and class! = masked then
        i_f_s ← ID
        crop box
        save box in output file
    end
end
```

2 Experimentation

2.1 Dataset description

As described in the previous paragraphs, our emphasis is on recognizing individuals who are not wearing face mask, which prompted us to search for a dataset containing a big number of no masked faces to help us with that, and we found the solution in the famous database called wider face [56].

The proposed dataset contains 18,000 images. We alleviate the proposed dataset is collected from multiple datasets, 12,800 raw images selected from the Wider-Face-Dataset. The

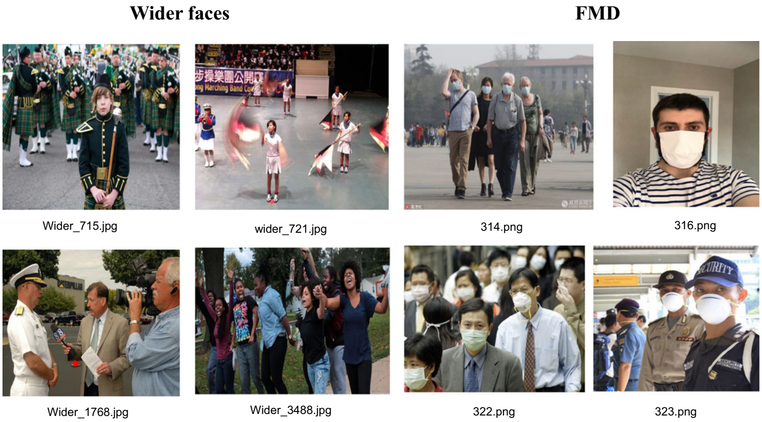


Fig. 6 Images from wider face dataset and FMD dataset

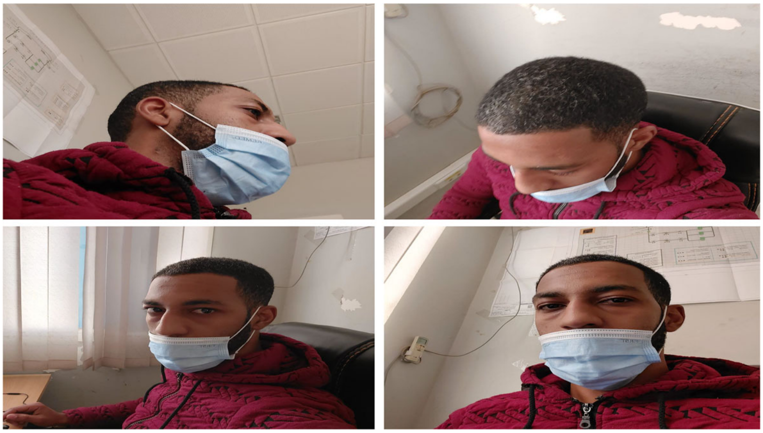
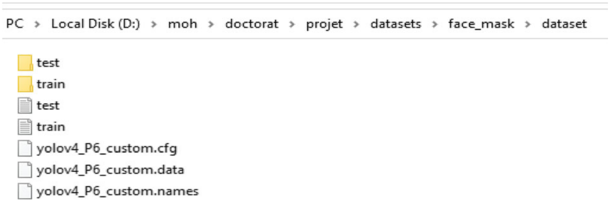


Fig. 7 Incorrectly masked faces from the proposed dataset



PC › Local Disk (D:) › moh › doctorat › projet › datasets › face_mask › dataset

test
train
test
train
yolov4_P6_custom.cfg
yolov4_P6_custom.data
yolov4_P6_custom.names

Fig. 8 Dataset pre-processing and converting to YOLO format

Wider- Face is a real-unmasked-face dataset, but many images contain a masked face. The remainder of 7200 pictures of our dataset is transferred from Kaggle, namely FMD and RMFD (Fig. 6), including masked faces and incorrectly masked faces, and some of our pictures of the incorrectly masked face (Fig. 7). The Wider-face dataset has 32,203 images and 393,703 faces with a high degree of variability in scale/pose/occlusion. The dataset is split into training, testing data. This will allow different DL based state-of-the-art object detection researchers to produce challenging and comparative results. Our dataset consists of three categories named masked faces, no masked faces and incorrectly masked faces.

2.2 Dataset pre-processing

In this subsection, we explain how to convert data from a given format to a YOLO format. The annotation of the Wider Face dataset, FMD and RMFD was different and did not meet our requirement (YOLO annotation format). We convert datasets annotation to Scaled-YOLOv4 format by steps:

- Create dataset file.
- Put all images inside the dataset file.
- Partition the dataset to train file and test file (80% of images for training and 20% for testing).
- Create file.txt (text file contain annotation of images).
- Create train.txt (annotation of training images) / test.txt (annotation of testing images) / file.names contain classes name / file.data.
- Create the configuration Yolov4-P6-custom.cfg.

See Fig. 8.

2.3 Detection results

Since mask detection is essentially a classification and localization task, it is evaluated using typical metrics, including true positive (TP), true negative (TN), false positive (FP), and false negative (FN), based on precision and recall defined as follows:

Table 5 YOLOv4-P6-FaceMask Model Parameters

Parameters	Value
Width	1280
Height	1280
Momentum	0.96
Learning rate	0.001
Batch_size	64
Subdivisions	8
Activation function	mish
Classes	3
Mini-batches	600
Weight decay	0.0004

Table 6 YOLOv4-P6-FaceMask Model 1280 × 1280 -Training Results

Iteration	mAP	Avg Loss
1000	47%	4.19
2000	67.16%	2.87
3000	79.05%	2.2
4000	87.6%	1.9
5000	89.03%	1.8
6000	93.02%	1.8

$$Precision = \frac{TP}{TP + FP} \tag{16}$$

$$recall = \frac{TP}{TP + FN} \tag{17}$$

In addition, the evaluation uses Intersection over Union (IoU), which gives the ratio of the overlapping area of the predicted boxes to the corresponding ground truth, larger IoU values reflect more accurate localization, so IoU = 1 is the best case. Combined with the IoU value, AP50 and AP75 are applied to report the average precision at IoU = 0.5 and IoU = 0.75 levels. mAP and mAR represent the means of the 10 precision and recall values at IoU, ranging from 0.5 to 0.95 with an interval of 0.05 for detailed performance in each category to further evaluate the overall performance of the facemask detection model. We selected the batch size value (set to 64) and subdivisions (set to 8: depending on the performance of GPU). The input image is set to width × height = 1280 × 1280 pixels. We used this prepared model to process the input image. A momentum of 0.96, a batch normalization = 1, activation function = mish,weight-decay = 0.0004 were used. The learning rate is L_r = 0.001 for 600 mini-batches; which is calculated using the following method C*2000 = 6000. We spent 14 hours more on the training of model using GPU Tesla-T4 of Google-Collaborator. The mean average precision equal to 93% with input-size 1280 × 1280 after 6000 iterations and average loss equal to 1.8%. Tables 5 and 6 show all training parameters and results.

Table 7 Comparison of our YOLOv4-P6-FaceMask face mask detector with state-of-the-art

Model-Name	AP50%	AP75%	mAP %	mAR %	FPS %
Faster RCNN [44]	80	72	80	70	35
YOLOv3 [42]	82	75	82	80	30
YOLOv4 416 [8]	81	74	80	75	57
YOLOv4 512 [8]	83	77	83	83	50
YOLOv4 608 [8]	86	80	86	82	44
EfficientDet [48]	89	84	87	87	32
RenitaFaceMask [18]	91	88	89	89	29
YOLOv4-P6-FaceMask (**Ours**)	**94**	**90**	**93**	**92**	**35**

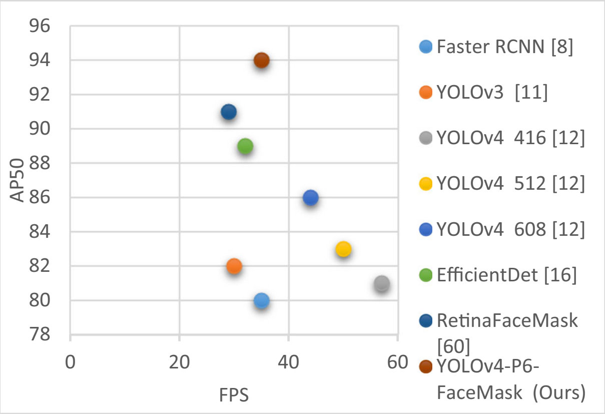


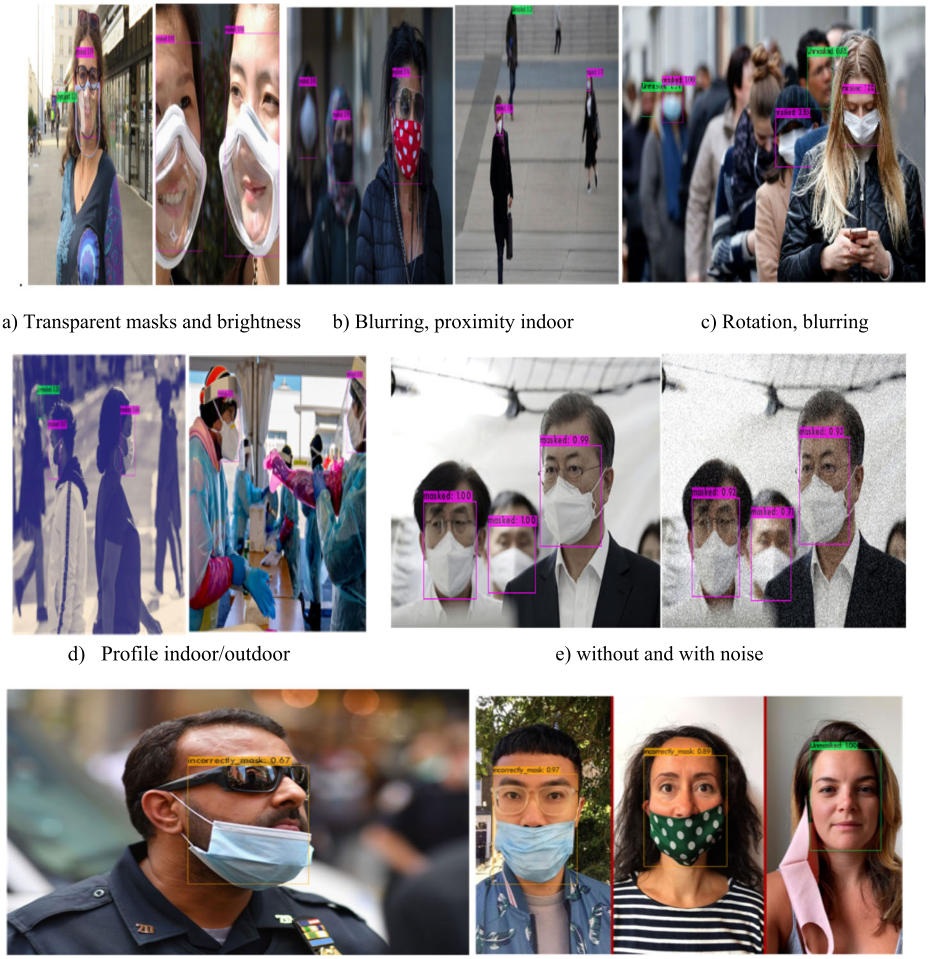
Fig. 9 Comparison of the proposed YOLOv4-P6-FaceMask with state-of-the-art object detection models

2.3.1 Comparison with state-of-the-art methods

To evaluate the proposed model, we trained some state-of-the-art models on our proposed dataset and in the same platform implementation (tesla T4). The classification accuracy and real time performance obtained by YOLOv4-P6-FaceMask model is compared with classification accuracy obtained by YOLOv4, YOLOv3, Faster RCNN, EfficientDet and RenitaFaceMask [18]. A comparison of the trained models in the proposed face mask dataset is presented in Table 7. We discovered that YOLOv4-P6-FaceMask can outperform Faster RCNN by 13% and can outperform YOLOv3 and YOLOv4 by 11% and 7%, respectively, and can outperform EfficientDet and RenitaFaceMask by 7% and 4% in term of mean average precision (mAP). The results are illustrated in Table 7 and Fig. 9.

2.3.2 Results and discussion on brightness, blurring, noise and proximity in images of YOLOv4-P6-FaceMask

In the first part of illustrated in Figs. 10 and 11, the experiments that we conducted is to validate the effectiveness and accuracy of the model (**Green bounding boxes**: unmasked faces without identification number ID; **Red bounding boxes**: masked faces without identification number ID; **orange bounding boxes**: incorrectly masked faces without identification number ID). We explored the performance of our YOLOv4-P6-FaceMask detector on images that contain difficulties and obstacles such as brightness, blurring, noise and proximity faces to the camera... to show the effectiveness and accuracy of the model. See Fig. 10a, b, c, d, e, and we can observe that our proposed model gives excellent results in all of the previously mentioned cases of difficulties. In other part of the experiments, we conducted to validate the effectiveness and accuracy of the model: we explored the performance of YOLOv4-P6-FaceMask detector on images that contain difficulties and obstacles such as different poses (rotation angles), profiles, and different formats and types of masks for example transparent masks... to show the effectiveness and accuracy of the model. See Fig. 10a, b, c, d and 11a, b and we can observe that our proposed model gives excellent performance



a) Transparent masks and brightness b) Blurring, proximity indoor c) Rotation, blurring

d) Profile indoor/outdoor e) without and with noise

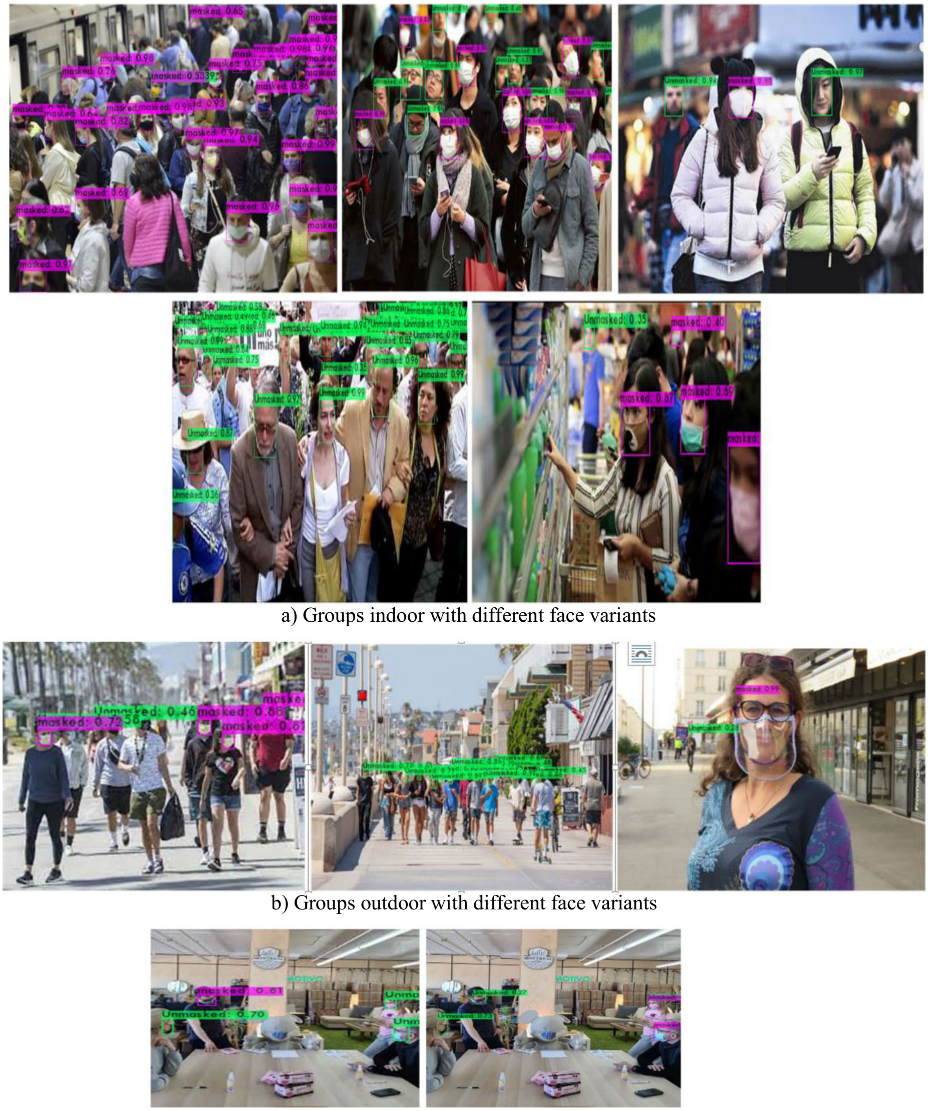
f) Correctly, incorrectly and without the mask

Fig. 10 Visual examples generated by YOLOv4-P6-FaceMask (first part). Green bounding boxes: unmasked faces without identification number ID; Red bounding boxes: masked faces without identification number ID; orange bounding boxes: incorrectly masked faces without identification number ID

in all the cases of difficulties mentioned above in indoor and outdoor except in the case of incorrectly mask. Where a lack of accuracy is obtained, due to the small number of images of this category in conjunction with other groups masked and unmasked in the training step. Figure 11c shows a different image resolution (low and medium). We can say that the model precision diminished in the low resolution of images.

2.3.3 Results and discussion of surveillance video

Indoor and outdoor videos are selected with difficulties and obstacles such as the different resolution of the video, brightness, blurring, different rotation angles of faces, profile, and



a) Groups indoor with different face variants

b) Groups outdoor with different face variants

c) Different images with low and medium resolution

Fig. 11 Visual examples generated by YOLOv4-P6-FaceMask (second part)

proximity of the faces... to show the effectiveness and accuracy of the model. The results demonstrated in Fig. 12.

As for the accuracy of the proposed model, we can notice that the precision diminished in the case of the image low resolution, and in term of real-time performance, we observe that the model gives an acceptable performance 35 fps.

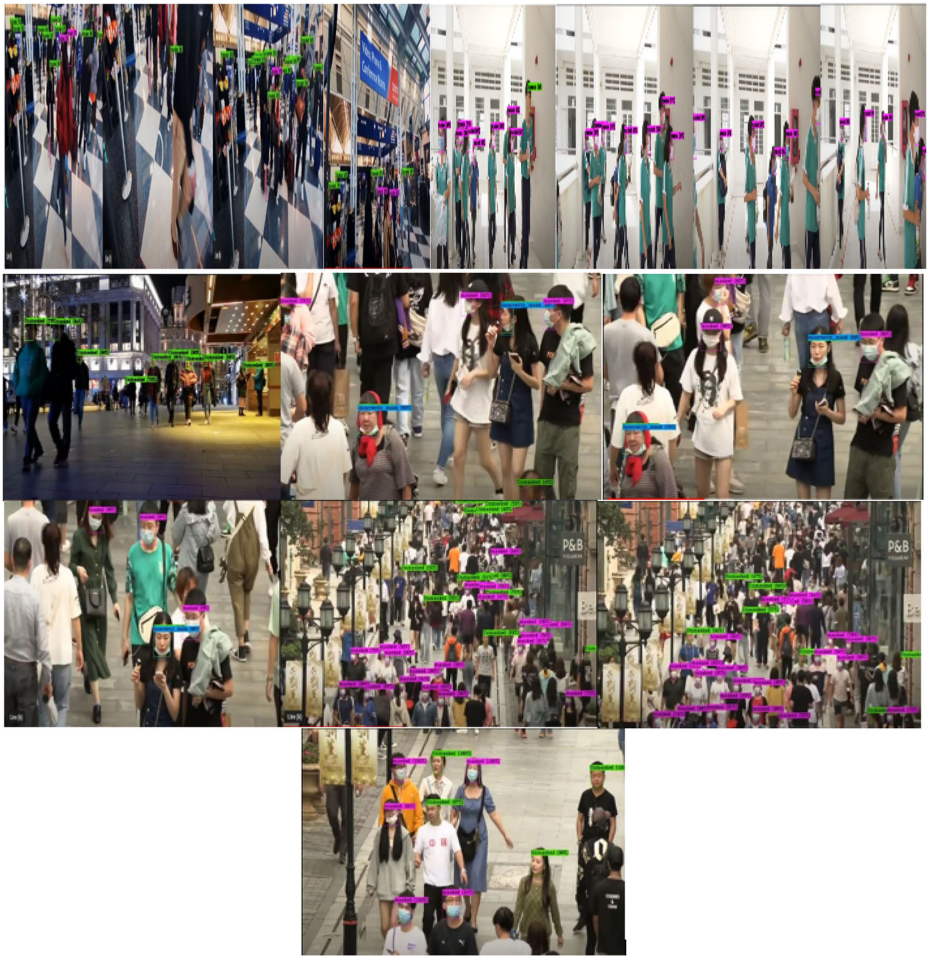
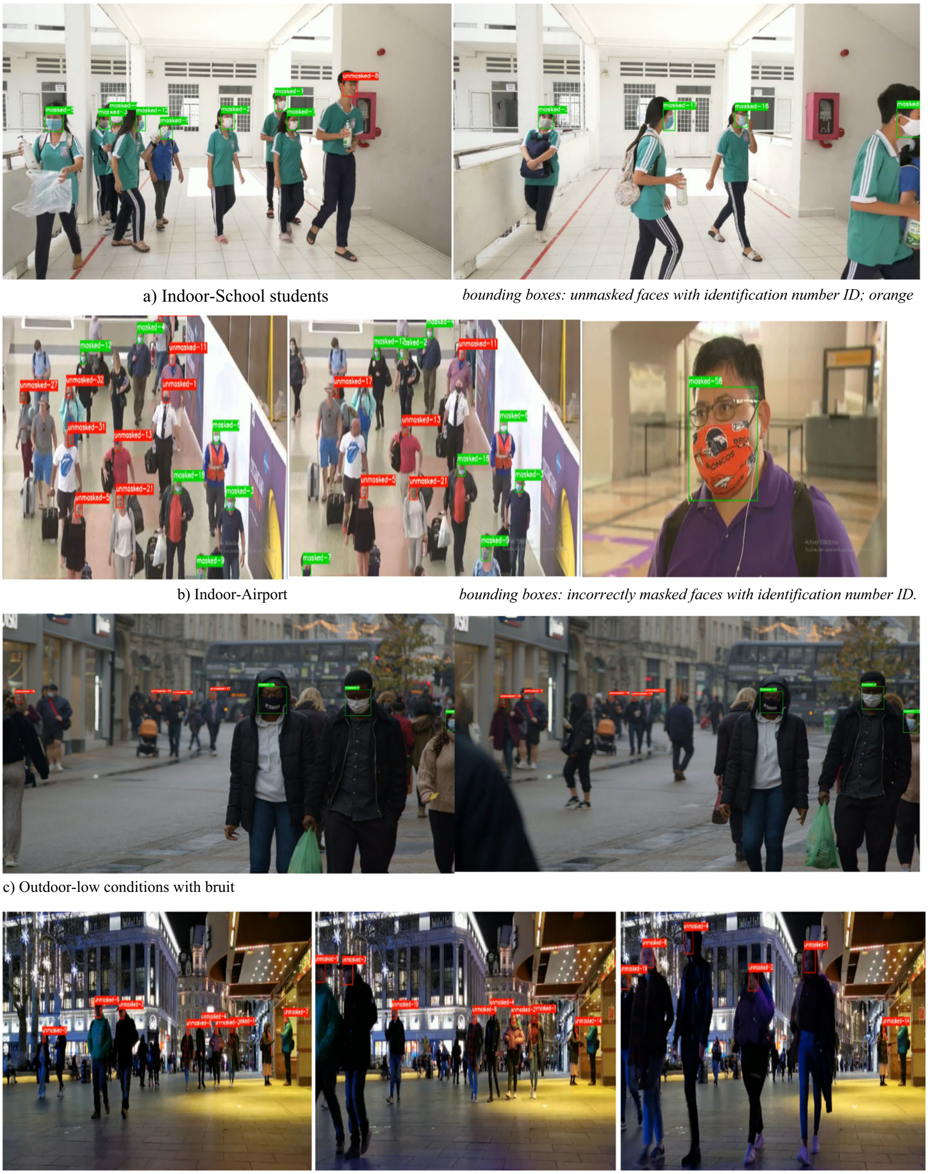


Fig. 12 Real-time surveillance video examples generated by YOLOv4-P6-FaceMask outdoor. Green bounding boxes: unmasked faces without identification number ID; Red bounding Boxes without identification number ID: masked faces: blue bounding boxes: incorrectly masked faces without identification number ID

2.4 Tracking and faces extraction results

To obtain the accuracy of the model presented in this research paper, we have experimented with a series of indoor and outdoor videos. The model gives acceptable results in tracking in both indoor and outdoor environments see Fig. 13, with good accuracy of the YOLOv4-P6-FaceMask. Figure 14 shows the result of extracted unmasked/incorrectly masked faces that saved in file. We notice only one error with save a masked face in all the sequences video.



a) Indoor-School students *bounding boxes: unmasked faces with identification number ID; orange*

b) Indoor-Airport *bounding boxes: incorrectly masked faces with identification number ID.*

c) Outdoor-low conditions with bruit

d) Outdoor-low conditions

Fig. 13 Real-time surveillance video examples generated by YOLOv4-P6-FaceMask detector and DeepSort tracker. Green bounding boxes: masked faces with identification number ID; Red bounding boxes: unmasked faces with identification number ID; orange bounding boxes: incorrectly masked faces with identification number ID

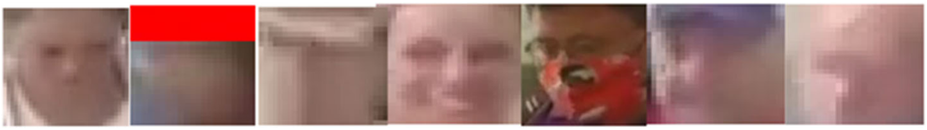


Fig. 14 Results with crop and save unmasked / incorrectly masked faces

Table 8 Comparison YOLOv4-P6-FaceMask detection with state-of-art models on different BDD and sizes

Model name	Publication	Dataset name	Dataset Size	Precision (%)
YOLO-V2+ResNet50 [34]	Feb 2021	MMD [2] + FMD	1415	81
SSDMNV2 [38]	Mar 2021	RMFD + PyImageSearch	5521	92.64
RetinaFaceMask [18]	Oct 2021	Face Mask Dataset	7971	93.4
SE-YOLOv3 [25]	Apr 2021	New collected dataset	9205	71.9
Scaling up FaceMasks detection [27]	2021	New dataset	52,635	71.69
FMD-YOLO [56]	2022	From kaggle website	7932	92
YOLOv4-P6-FaceMask (ours)	–	**WiderFace+FMD+ RMFD**	**18,000**	**93**

2.5 Implementation platform and libraries

To implement the facemask detection and tracking framework, the Python 3 language on Google-Colab notebook is used. In the first stage, we based on darknet project to train the YOLOv4-P6-FaceMask detector. The detection model trained and performed in a single GPU Tesla-T4 of Google-Collab. The libraries used in the implementation processes: darknet, Keras, Os, OpenCv, NumPy, MatPlotLib and pillow (Table 8).

2.6 Limits of the work

Among the limitations of the model:

- The imperfection in the accuracy of detection model of people who wear the medical mask incorrectly. This is due to the training with a lack of the number of incorrectly masked faces images.
- The inability to compare the face-tracking model with other models due to the lack of a standard database dedicated to this task.

3 Conclusion

In this paper an efficient automatic two-stage framework of face-mask detection and tracking based our YOLOv4-P6-FaceMask detection model and pre-trained DeepSORT tracker are proposed. In particular, we proposed a new dataset for facemask detection with 18,000 images and trained the YOLOv4-P6-FaceMask face-mask detection model, which can contribute to public healthcare. The network architecture of the proposed model consists of CSP-ized-CSPDarknet53 as the backbone, CSPSPP and PAN as the neck and CSP-ized-YoloV3 module as the heads and we scaling up the model network and mish as activation function. After that, the DeepSORT tracker is used to track faces, this method helps us to crop and save faces only once per person in all sequences. In order to extract, more robust features we believe that our work propose a dataset, a model named YOLOv4-P6-FaceMask and could contribute to preventing the COVID-19 from pervasion for protect against other infectious diseases; which can be prevalent by such things as speaking at close range, coughing, sneezing. The proposed model achieves state-of-the-art results on face mask datasets, with an accuracy of 93%, a mean average recall of 92%, a real-time speed of 35 fps with input 1280 × 1280, and average loss equal to 1.8%.

In future work, we aspire to:

- Increase the accuracy of our YOLOv4-P6-FaceMask
- Build applications using the YOLOv4-P6-FaceMask model (social distance applications, android applications …)
- Combined YOLOv4-P6-FaceMask with another recent tracker
- Using the images extracted from the model to find out if the person is sick or shows symptoms of the disease from his facial features
- Integrate the proposed approach in embedded system (Raspberry Pi, Drone…)

Data availability The datasets used during the current study are available in the repositories (https://www.shuoyang1213.me/WIDERFACE/.https://www.kaggle.com/andrewmvd/face-mask-detection) The results of this work are available in the repository. (https://www.drive.google.com/drive/folders/1GkZk5WaVcz6vHbNvGQQjb4WcyRrE9m0H?usp=sharing)

Declarations

Conflict of interests The authors declare that they have no conflict of interest.

References

1. "FMD", Kaggle, (2020). [Online]. Available. https://www.kaggle.com/andrewmvd/face-mask-detection
2. “MMD” Kaggle, (2020). [Online]. Available., https://www.kaggle.com/vtech6/medical-masks-dataset
3. “SMFD” Kaggle, (2020) [Online]. Available, Accessed 25 May 2020 https://github.com/prajnasb/observations
4. Ameur B, Belahcene M, Masmoudi S, Hamida AB (2019) Efficient hybrid descriptor for face verification in the wild using the deep learning approach. https://doi.org/10.3103/S1060992X19030020
5. Belahcene M (2013) Biometric identification and authentification. Phd Thesis. Mohamed Khider University, Biskra
6. Belahcene M (2013) Biometric identification and authentification. Phd Thesis. Mohamed Khider University, Biskra http://thesis.univ-biskra.dz/id/eprint/944
7. Bewley A, Ge Z, et al (2016) simple online and realtime tracking. In 2016 IEEE international conference on image processing, ICIP, pp. 3464-3468. IEEE
8. Bochkovskiy A, Wang C Y, Liao H Y M (2020) YOLOv4: optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934
9. Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2017) Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Trans Pattern Anal Mach Intell 40(4):834–848
10. Chen J, Mai H, Luo L, Chen X, Wu K (2021) Effective feature fusion network in BIFPN for small object detection. In 2021 IEEE international conference on image processing (ICIP) (pp. 699-703). IEEE
11. Chowdary GJ, Punn NS et al (2020) Face mask detection using transfer learning of inceptionv3. In: International conference on big data analytics. Springer, Cham, pp 81–90. https://doi.org/10.1007/978-3-030-66665-1_6
12. Chu P, Fan H, Tan CC, Ling H (2019) Online multi-object tracking with instance-aware tracker and dynamic model refreshment. In 2019 IEEE winter conference on applications of computer vision (WACV) (pp. 161-170). IEEE
13. Ciaparrone G, Sánchez FL, Tabik S, Troiano L, Tagliaferri R, Herrera F (2020) Deep learning in video multi-object tracking: a survey. Neurocomputing 381:61–88. https://doi.org/10.1016/j.neucom.2019.11.023
14. Ding M, Xiao B, Codella N, Luo P, Wang J, Yuan L (2022) DaViT: dual attention vision transformers. arXiv preprint arXiv:2204.03645
15. Ejaz MS, Islam MR, Sifatullah M, Sarker A, (2019) Implementation of principal component analysis on masked and non-masked face recognition. In 2019 1st international conference on advances in science, engineering and robotics technology, ICASERT, pp. 1-5. IEEE

16. Elaggoune H, Belahcene M, Bourennane S (2022) Hybrid descriptor and optimized CNN with transfer learning for face recognition. Multimed Tools Appl 81(7):9403–9427. https://doi.org/10.1007/s11042-021-11849-1
17. Everingham M, Van Gool L, Williams CK, Winn J, Zisserman A (2010) The pascal visual object classes (voc) challenge. Int J Comput Vis 88(2):303–338
18. Fan X, Jiang M (2021) RetinaFaceMask: a single stage face mask detector for assisting control of the COVID-19 pandemic. In 2021 IEEE international conference on systems, man, and cybernetics (SMC) (pp. 832-837). IEEE
19. Girshick R (2015) Fast R-CNN. In proceedings of the IEEE international conference on computer vision. Pp 1440-1448
20. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587)
21. Girshick R, Donahue J, Darrell T, Malik J (2015) Region-based convolutional networks for accurate object detection and segmentation. IEEE Trans Pattern Anal Mach Intell 38(1):142–158. https://doi.org/10.1109/TPAMI.2015.2437384
22. Goyal H, Sidana K, Singh C, Jain A, Jindal S (2022) A real time face mask detection system using convolutional neural network. Multimed Tools Appl 81:1–17
23. He K, Zhang X, Ren S, Sun J (2015) Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans Pattern Anal Mach Intell 37(9):1904–1916
24. Jiang M, Fan X, Yan H, (2020) Retinamask: a face mask detector. arXiv preprint arXiv:2005.03950
25. Jiang X, Gao T, Zhu Z, Zhao Y (2021) Real-time face mask detection method based on YOLOv3. Electronics 10(7):837. https://doi.org/10.3390/electronics10070837
26. Kim M, Alletto S, Rigazio L (2016) Similarity mapping with enhanced siamese network for multi-object tracking. arXiv preprint arXiv:1609.09156
27. Kumar A, Kalia A, Verma K, Sharma A, Kaushal M (2021) Scaling up face masks detection with YOLO on a novel dataset. Optik 239:166744. https://doi.org/10.1016/j.ijleo.2021.166744
28. Li X, Yang Z, Wu H (2020) Face detection based on receptive field enhanced multi-task cascaded convolutional neural networks. IEEE Access 8:174922–174930
29. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Zitnick CL (2014) Microsoft coco: common objects in context. In: European conference on computer vision. Springer, Cham, pp 740–755. https://doi.org/10.1007/978-3-319-10602-1_48
30. Lin T Y, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. In proceedings of the IEEE international conference on computer vision. Pp 2980-2988. arXiv:1708.02002
31. Liu S, Huang D (2018) Receptive field block net for accurate and fast object detection. In proceedings of the European conference on computer vision (ECCV) (pp. 385-400)
32. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016) Ssd: single shot multibox detector. In: European conference on computer vision. Springer, Cham, pp 21–37. https://doi.org/10.1007/978-3-319-46448-0_2
33. Liu Z, Mao H, Wu CY, Feichtenhofer C, Darrell T, Xie S (2022) A ConvNet for the 2020s. arXiv preprint arXiv:2201.03545
34. Loey M, Manogaran G, Taha MHN, Khalifa NEM (2021) Fighting against COVID-19: a novel deep learning model based on YOLO-v2 with ResNet-50 for medical facemask detection. Sustain Cities Soc 65:102600. https://doi.org/10.1016/j.scs.2020.102600
35. Luo S, Li X et al (2019) SFA: small faces attention face detector. IEEE Access 7:171609–171620
36. Milan A, Leal-Taixé L, et al (2016) MOT16: A benchmark for multi-object tracking. arXiv preprint arXiv:1603.00831
37. Misra D (2019) Mish: a self regularized non-monotonic activation function. arXiv preprint arXiv:1908.08681
38. Nagrath P, Jain R, Madan A, Arora R, Kataria P, Hemanth J (2021) SSDMNV2: a real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2. Sustain Cities Soc 66:102692
39. Prasad P, Chawla A (2022) Facemask detection to prevent COVID-19 using YOLOv4 deep learning model. In 2022 second international conference on artificial intelligence and smart energy (ICAIS) (pp. 382-388). IEEE
40. Qin B, Li D (2020) Identifying facemask-wearing condition using image super-resolution with classification network to prevent COVID-19. Sensors 20(18):5236. https://doi.org/10.3390/s20185236

41. Redmon, J, Ali F (2017) YOLO9000: better, faster, stronger. Proceedings of the IEEE conference on computer vision and pattern recognition. p 7263–7271
42. Redmon J, Ali F (2018) YOLOv3: An incremental improvement. arXiv preprint arXiv:1804.02767
43. Redmon J, Divvala S, Girshick R, Farhadi A (*2016*) You only look once: unified, real-time object detection. In proceedings of the IEEE conference on computer vision and pattern recognition. Pp 779-788
44. Ren S, He K, Girshick R, Sun J (2016) Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 39(6):1137–1149
45. Sadeghian A, Alahi A, Savarese S (2017) Tracking the untrackable: learning to track multiple cues with long-term dependencies. In proceedings of the IEEE international conference on computer vision, pp. 300-311
46. Sanchez-Matilla R, Poiesi F, Cavallaro A (2016) Online multi-target tracking with strong and weak detections. In: European conference on computer vision. Springer, Cham, pp 84–99
47. Tan M, Le Q (2021) Efficientnetv2: smaller models and faster training. In international conference on machine learning (pp. 10096-10106). PMLR
48. Tan M, Pang R, Le Q V (2020) Efficientdet: scalable and efficient object detection. In proceedings of the IEEE/CVF conference on computer vision and pattern recognition. Pp 10781-10790. arXiv:1911.09070
49. Tripathi S, Augustin AI, Sukumaran R, Dheer S, Kim E (2022) HematoNet: expert level classification of bone marrow cytology morphology in hematological malignancy with deep learning. medRxiv
50. Wang CY, Liao HYM, et al (2020) CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, pp. 390–391
51. Wang CY, Liao HYM, Wu YH, Chen PY, Hsieh JW, Yeh IH (2020) CSPNet: a new backbone that can enhance learning capability of CNN. In proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops (pp. 390-391)
52. Wang Z, Wang G, Huang B, Xiong Z, et al (2020) Masked face recognition dataset and application. arXiv preprint arXiv:2003.09093
53. Wang, ChY, Alexey B, Hong Y, Mark L (2021) Scaled-yolov4: Scaling cross stage partial network." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition
54. Wojke N, Bewley A, Paulus D (2017) Simple online and realtime tracking with a deep association metric. In 2017 IEEE international conference on image processing, ICIP, pp. 3645-3649. IEEE
55. Wu X, Xue Q (2021) An improved CornerNet-lite method for pedestrian detection of unmanned aerial vehicle images. In 2021 China automation congress (CAC) (pp. 2322-2327). IEEE
56. Wu P, Li H, Zeng N, Li F (2022) FMD-Yolo: an efficient face mask detection method for COVID-19 prevention and control in public. Image Vis Comput 117:104341
57. Yang B, *Yan J*, et al (2015) Convolutional channel features. In Proceedings of the IEEE international conference on computer vision, pp. 82–90
58. Yang S, Luo P, et al (2015) From facial parts responses to face detection: A deep learning approach. In Proceedings of the IEEE international conference on computer vision, pp. 3676–3684
59. Yang S, Luo P, Loy CC, Tang X (2016) Wider face: a face detection benchmark. In proceedings of the IEEE conference on computer vision and pattern recognition, pp. 5525-5533
60. Zhao ZQ, Zheng P, Xu ST, Wu X (2019) Object detection with deep learning: a review. IEEE Trans Neural Netw Learn Syst 30(11):3212–3232
61. Zhu C, Zheng Y, Luu K, Savvides M (2017) Cms-rcnn: contextual multi-scale region-based cnn for unconstrained face detection. In: Deep learning for biometrics. Springer, Cham, pp 57–79. https://doi.org/10.1007/978-3-319-61657-5_3
62. Zou Z, Shi Z, Guo Y, Ye J (2019) Object detection in 20 years: a survey. arXiv preprint arXiv:1905.05055

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.