Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Elsevier hereby grants permission to make all its COVID-19-related research that is available on the COVID-19 resource centre - including this research content - immediately available in PubMed Central and other publicly funded repositories, such as the WHO COVID database with rights for unrestricted research re-use and analyses in any form or by any means with acknowledgement of the original source. These permissions are granted for free by Elsevier for as long as the COVID-19 resource centre remains active.
FMD-Yolo: An efficient face mask detection method for COVID-19 prevention and control in public

Peishu Wu, Han Li, Nianyin Zeng, Fengping Li

1. Introduction

Coronavirus Disease 2019 (COVID-19) is a world-wide epidemic and efficient prevention and control of this disease has become the focus of global scientific communities. In this paper, a novel face mask detection framework FMD-Yolo is proposed to monitor whether people wear masks in a right way in public, which is an effective way to block the virus transmission. In particular, the feature extractor employs Im-Res2Net-101 which combines Res2Net module and deep residual network, where utilization of hierarchical convolutional structure, deformable convolution and non-local mechanisms enables thorough information extraction from the input. Afterwards, an enhanced path aggregation network En-PAN is applied for feature fusion, where high-level semantic information and low-level details are sufficiently merged so that the model robustness and generalization ability can be enhanced. Moreover, localization loss is designed and adopted in model training phase, and Matrix NMS method is used in the inference stage to improve the detection efficiency and accuracy. Benchmark evaluation is performed on two public databases with the results compared with other eight state-of-the-art detection algorithms. At IoU = 0.5 level, proposed FMD-Yolo has achieved the best precision AP50 of 92.0% and 88.4% on the two datasets, and AP75 at IoU = 0.75 has improved 5.5% and 3.5% respectively compared with the second one, which demonstrates the superiority of FMD-Yolo in face mask detection with both theoretical values and practical significance.

This work was supported in part by the Natural Science Foundation of China under Grant 62073271, in part by the UK-China Industry Academia Partnership Programme under Grant UK-CIAPP-276, in part by the International Science and Technology Cooperation Project of Fujian Province of China under Grant 2019I0003, in part by the Fundamental Research Funds for the Central Universities under Grant 20720190009, in part by the Open Fund of Engineering Research Center of Big Data Application in Private Health Medicine under Grant KF2020002, and in part by the Fujian Key Laboratory of Automotive Electronics and Electric Drive (Fujian University of Technology) under Grant KF-X190002.

Corresponding author.
E-mail address: zny@xmu.edu.cn (N. Zeng).
millisecond face detection speed and is suitable for deployment on mobile devices. [15] proposes a deep regression-based user image detector (DRUID), which makes segmentation-based deep convolution algorithms available in the mobile domain and also has great advantages of saving time. A face detection model that integrates Squeeze-Excitation Network and ResNet is designed in [44], where a regression network is applied to track and localize faces in video sequences. As for mask detection, [32] applies YOLOv3 and Faster RCNN to detect whether a mask is worn or not, and [24] proposes a transfer learning method based on ResNet50 and YOLOv2 for the detection of medical masks. In addition, RCNN series algorithms are adopted to realize a multi-task problem in [16], including social distance calculation and mask detection.

According to above discussions, it is found that few of current methods have highlighted the unique characteristics of face mask detection, and they generally handle this problem as a conventional object detection task. However, in practice not only detection of both mask and face is necessary, but localization of the mask in face is also of vital significance, which can estimate whether the mask is worn in the right or wrong way. In order to meet the special demand, in this paper, a face mask wearing oriented detection scheme is proposed along with the corresponding algorithm, as shown in Fig. 1. In particular, three scenarios have been defined, including not wearing masks and wearing mask in the right/wrong way. Before splitting the data, preprocessing operations are performed to eliminate dirty data and adjust image size. Afterwards, anchor clustering is performed to select suitable anchor scale of the training data, which benefits model training by introducing in specific designed prior knowledge for different tasks; and then other data augmentation operations are applied to enrich the samples. By means of sufficient training epochs, the best model and corresponding weight parameters will be saved and evaluated. Finally, once the precision reaches a predefined criterion, the model will be deployed on the terminal device to realize practical application. It is remarkable that when the video streams or pictures are fed into terminal devices, in-time detection results can be outputted and visualized through the designed algorithm. Consequently, intelligent monitoring of mask wearing can be accomplished by deploying relevant devices in places with high population flow-rate such as airports and supermarkets, which can effectively improve the efficiency of COVID-19 epidemic prevention and control. The major contributions of this paper are summarized as follows:

1) A general face mask detection framework is proposed, which is mask wearing oriented and can be easily extended for secondary development and improvement.

2) A novel face mask detection algorithm FMD-Yolo is proposed, where the feature extractor, feature fusion operation and post-processing techniques are all specifically designed.

3) Experimental results on two public datasets have fully demonstrated the superiority of FMD-Yolo, which outperforms several state-of-the-art methods. Moreover, the high generalization ability enables application in other situations.

The remainder of this paper is organized as follows. Some preliminaries of related work are provided in Section 2 and in Section 3, and detailed implementations of FMD-Yolo are elaborated, including data preprocessing, analytic methods, overall structure and post-processing techniques. Benchmark evaluation results and parallel comparisons with comprehensive discussions are presented in Section 4, and in Section 5, conclusions are drawn.

2. Related work

For a comprehensive and in-depth understanding of proposed face mask detection algorithm, some background information are presented in this section, including the feature extractor, feature fusion methods and YoloV3 framework.

2.1. Feature extractor based on residual structure

In the field of computer vision (CV), one important research is to effectively extract features from multi-scale information. With the rapid development of deep learning techniques, feature extraction has gradually transformed from manual design to automatic construct by algorithms [14]. In object detection tasks, corresponding algorithm is effectively transformed from manual design to automatic construct by algorithms [14]. In object detection tasks, corresponding algorithm is known as the backbone, and up to now plenty of networks have emerged such as VGG [30], ResNet [8], DenseNet [9], DLA (Deep Layer Aggregation) [40]. One practical strategy applied in those methods is to enrich connections among feature maps, like stacking layers and setting skip connections.

It should be pointed out that ResNet is one of the most widely applied backbone which designs a residual block to overcome the degradation problem associated with training a deep network. Res2Net [5] constructs a hierarchical progressive residual connection, which is conducing to improving the multi-scale feature expression ability of the network at a finer level. Major innovation of Res2Net is to replace 3 × 3 convolution in the bottleneck of ResNet with hierarchical cascaded feature group convolution, and this modification is shown in Fig. 2. The design of hierarchical convolution facilitates enlarging the receptive

Fig. 1. General flowchart of face mask detection algorithm (FMDA) framework.
field within the bottleneck block, where more refined multi-branches can achieve better capture of the objects. To be specific, Res2Net integrates $s$ groups of convolution kernel and has $s \times w$ channels in total, where $w$ represents the width of network. With larger value of $s$, more abundant receptive fields will be allowed to learn multi-scale features. In this paper, $s$ and $w$ are set to 4 and 26 respectively.

As shown in Fig. 2, bottleneck in Res2Net is divided into $s$ subsets after the $1 \times 1$ convolution, which are defined as $X_i, i \in \{1, 2, \cdots, s\}$. Each subset has the same size as the input, but the number of channels becomes $\frac{1}{4}$. Except for $X_1$, each sub-feature is performed a $3 \times 3$ convolution operation $C_i(\cdot), i \in \{2, 3, \cdots, s\}$ and lateral connections are adopted for enhancing interaction. The hierarchical residual connection enables the changes of receptive fields at a finer grained level to capture details and global characteristics. Notice the $3 \times 3$ convolution in each group can receive information from the left branch, thus the final output can contain multi-scope features, which is denoted as $Y_i, i \in \{1, 2, \cdots, s\}$ and can be obtained as:

$$ Y_i = \begin{cases} X_i & i = 1; \\ C_i(X_i) & i = 2; \\ C_i(X_i + Y_{i-1}) & 2 < i \leq s. \end{cases} \quad (1) $$

### 2.2 Feature pyramid network

In early object detection algorithms, output of the backbone network will be directly fed into the subsequent component, which is the so-called detection head. However, these output feature maps usually cannot make full use of information to characterize objects at multiple scales, and simultaneously the resolution is too small compared with the original input. To overcome such limitation and enhance detection accuracy and reliability, feature fusion methods come to another research hot-spot, which can merge multi-scale features at different levels.

In [21], the authors propose a single shot detector (SSD), which directly uses feature maps of different stages to achieve multi-scale detection, but there is no interconnection among individual feature maps. Feature pyramid network (FPN) [19] applies lateral concatenation and up-sampling operation to effectively merge high-level semantic features in top layers and high-resolution information in bottom layers. It is remarkable that many other feature fusion methods are designed based on FPN, like path aggregation network (PANet) [23], which adds an extra bottom-up path. This simple bidirectional fusion is proven helpful as information in bottom layer can be delivered upwards, and afterwards, more complicated bidirectional fusion methods have emerged. For example, the bi-directional feature pyramid network (BiFPN) [33] adopts cross-scale connection to the PANet. By means of multiple repetitions, information of each stage is able to be sufficiently integrated, even the fused feature maps can be reused.

To sum up, the feature map fusion approaches have developed from simple manner like top-down and bidirectional fusion to more complex ones, as is shown in Fig. 3. $C_i(i \in \mathbb{N})$ denotes the feature maps generated at each stage of the backbone network and $P_i(i \in \mathbb{N})$ denotes obtained output after multi-scale feature fusion. Define the fusion operation as fusion($\cdot$), and the relationship between output and input can be depicted as:

$$ P_i, P_{i+1}, \cdots, P_{i+k} = \text{fusion}(C_i, C_{i+1}, \cdots, C_{i+k}), (i, k \in \mathbb{N}) \quad (2) $$

### 2.3 You only look once (YOLO)

YOLO (You only look once) series algorithms have already experienced four versions [27–29,3], which are deemed as magnified masterpieces in the field of object detection and have been widely applied into various scenarios, including road marking detection [39], license plate recognition [10] and agricultural production [37], etc. In particular, YoloV3 [29] is recognized as an important breakthrough, which makes a trade-off between detection precision and speed. Structure of YoloV3 is illustrated in Fig. 4, where backbone network adopts the DarkNet53 with fully connected layer removed, and primary components include convolution blocks and stacked residual units (Res Unit), which are denoted as CBR (Conv BN Relu) and ResN respectively. Notice each CBR block contains convolution, batch normalization and leaky relu activation; $N$ in ResN is the number of stacked residual units.

As shown in Fig. 4, another unique characteristic of YoloV3 is the implementation of multi-scale prediction, i.e., detection at $32 \times, 16 \times$ and $8 \times$ down-sampling for the input images, which generates feature maps with size of $13 \times 13, 26 \times 26$ and $52 \times 52$ for detection of large, medium and small size objects, respectively. It is also remarkable that feature fusion manner in YoloV3 is similar to the idea of FPN, which takes full advantages of high-level semantic features and low-level geometric information so that the overall framework can have strong feature presentation ability and consequently, more accurate detection will be achieved.
3. Detailed implementations of proposed FMD-Yolo

In this section, the overall framework of the proposed FMD-Yolo method is elaborated, including data preprocessing, model implementation and post-processing method.

3.1. Data pre-processing and analysis methods

Fig. 1 presents the procedure of applied data preprocessing, which contains sample and batch transformations. At first, Mixup [43] method is applied along with other traditional data augmentation operations like random color distort, expansion and flip, etc. Traditional methods enhance the robustness of model training by introducing noises and perturbations in a single image, but Mixup helps to enhance the generalization ability of the model and improve the stability of training process by linearly interpolating two random samples and corresponding labels in training set to generate new virtual samples, which can be depicted as:

\[
\tilde{x} = \lambda x_i + (1-\lambda)x_j \quad \tilde{y} = \lambda y_i + (1-\lambda)y_j
\]

where \(x_i\) and \(x_j\) represent two different types of the input, \(y_i\) and \(y_j\) are corresponding one-hot labels, and \(\lambda\) is mixing coefficient calculated by beta distribution \(\lambda \sim \text{Beta}(\alpha, \alpha), \alpha \in (0, \infty)\) (\(\alpha\) and \(\beta\) both take 1.5 in this paper).

Secondly, input images are resized to \(640 \times 640\) and then normalized by Normalize operation. To further match training data with FDM-Yolo model, k-means clustering and mutation operation in genetic algorithm are used to select the best combination of prior anchor sizes, i.e., the “Anchor Cluster” in Fig. 1 [4]. At last, the size of anchors is further optimized and determined by mutation operation in genetic algorithm. It is noteworthy that an appropriate anchor size setting benefits the model convergence and useful prior knowledge can be learned, which can speed up the model training process and obtain more matching results with the actual values, and detailed implementation flowchart of anchor cluster is shown in Fig. 5.

As shown in Fig. 5, firstly \(K\) initial clustering centers are selected where the \(K\) is set to 9 in this paper. Then each sample point is assigned to the nearest neighbor clusters, which is measured by the IoU(Intersection over Union)-based distance between clustering center and the sample point, as provided in Eq. (4). Until the cluster centers are fixed and no longer changed, k-means clustering is accomplished and genetic algorithm will be applied to evolve the anchor settings. To be specific, after parameter initialization (fitness function \(f\), mutation probability \(mp\), variation coefficient \(c\) and generations \(gen\)), mutation operation is carried out to generate a new generation, which is then multiplied with previously obtained clustering centers to calculate the \(f\). Original clustering center will be updated only when a larger fitness value is obtained, and after 1000 iterations the update is terminated with final anchor size setting saved.

\[
D(X_i) = 1 - \text{IOU}(X_i, C_i)
\] (4)

3.2. Details of FMD-Yolo network structure

Fig. 6 illustrates the framework and implementation details of proposed face mask detection model FMD-Yolo. There are three major components in designed deep network, which are Im-Res2Net-101, En-PAN and Classification + Regression. In particular, Im-Res2Net-101...
(101 denotes the number of layers) is a modified Res2Net structure serving as the backbone of FMD-Yolo, which is responsible for extracting features from the inputs and having three-stage outputs denoted as $C_3$, $C_4$, and $C_5$. En-PAN is a bidirectional fusion approach based on PANet to enhance the detection performance of FMD-Yolo network by further refining and fusing outputted information from backbone.

Meanwhile, the loss function of FMD-Yolo is a modification on that of YoloV3, which will be introduced in the next subsection. For a more comprehensive understanding, Fig. 7 and Fig. 8 present the designed Im-Res2Net-101 and En-PAN architecture, respectively.

Im-Res2Net-101 contains a total of five stages, and FMD-Yolo utilizes the output feature maps of the last three stages and passes them...
into the subsequent En-PAN feature fusion module. In stage 1, by using three $3 \times 3$ convolutions, the learning ability is guaranteed with only a small number of parameters. Then in stage 2, in order to retain as much valid information in the feature map as possible, a $3 \times 3$ max pooling operation with a stride size of 2 is performed after which the output will enter an Im-Res2Net-a module for 3 times. From stage 3 to 5, Im-Res2Net-b module is used and repeated for 4, 23 and 3 times respectively. Nonlocal operation is introduced in stage 3 and 4, and deformable convolution [45] is applied in stage 4 and 5. Since the size and shape of human faces and masks in real scenes are changeable, a fixed receptive field will lead to performance degradation. Nevertheless, deformable convolution allows modeling capability of geometric transformation by adding a two-dimensional offset values to conventional convolution operation, which enables sampling region to be deformed freely. The output feature maps of regular and deformable convolution at each position $c_0$ are denoted as $RC$ and $DC$ respectively, as described in Eq. (5) and Eq. (6).

$$RC(c_0) = \sum_{c \in S} w(c_0) \cdot x(c_0 + c_n)$$  \hspace{1cm} (5) 

$$DC(c_0) = \sum_{c \in S} w(c_0) \cdot x(c_0 + c_n + \Delta c_n)$$  \hspace{1cm} (6)

where $S$ refers to the sampling grid area, $c_n$ represents each position traversed on $S$, and $w(\cdot)$ is the weight function; $\Delta$ is the offset value that displaces each point on area $S$ by using offset $\{\Delta c_n | n = 1, 2, \ldots, |S|\}$.

In addition, to reduce loss of information after convolution, the right branch of Im-Res2Net-b applies a $2 \times 2$ average pooling with stride of 2 and a $1 \times 1$ convolution operation with stride of 1, rather than a $1 \times 1$ convolution with stride of 2. The Im-Res2Net-b module is followed by a Spacetime Non-local block, which is essentially based on a self-attention mechanism that makes the processed information not limited to a local region [35], and thus captures the remote dependencies directly by computing the interaction between any two positions. “Spacetime” represents space-based and time-based information, that is, the attention mechanism can also play its role in video sequences. Non-local operation directly fuses the global information instead of simply stacking multiple convolutional layers and can generate richer semantic information, which is equivalent to constructing a convolution kernel with the same size of corresponding input feature map. Following equation describes the work principle of Non-local operation.

$$y_i = \frac{1}{C(x)} \sum_{y \in Y} f(x_i, x_j)g(x_j)$$  \hspace{1cm} (7)

where $x$ and $y$ refer to the input and output feature maps, respectively; $i$ and $j$ represent the current and global position, which are both indices of space and time, $f(\cdot)$ is the calculation of similarity between $i$ and $j$, where the Embedded Gaussian function is used; $g(\cdot)$ calculates the linear transformation representation of the feature map at $j$ position, which can be realized by $1 \times 1$ convolution; $C(\cdot)$ is a response factor, and the final output $y$ is obtained after standardization.

Fig. 8 illustrates the structure and implementation of En-PAN, which adopts the bi-directional fusion of PAnet and makes full advantages of multi-scale information based on the Yolo Residual module to enhance feature presentation ability. Except for a top-down path, a bottom-up path is added and connected horizontally to fully integrate the overall target information of both the high-level feature maps and the low-level texture information. The top-down and bottom-up paths go
through a total of five Yolo Residual modules to thoroughly improve the robustness, and to make those feature maps match in size. Upsample block and $3 \times 3$ convolution operation with stride of 2 have been employed in the top-down and bottom-up path, respectively. Based on aforementioned three-stage outputs of $C_5$, $C_4$ and $C_5$ of Im-Res2Net-101, output of En-PAN can be obtained as:

$$
\begin{align*}
    P_3 &= YR(F_3), F_3 = C_5 \oplus Up(F_4); \\
    P_4 &= YR(Co(P_3) \oplus (F_3)), F_4 = YR(C_4 \oplus Up(F_3)); \\
    P_5 &= YR(Co(P_4) \oplus (F_3)), F_5 = YR(C_5)
\end{align*}
$$

(8)

where $YR$ and $Up$ denote the operations associated with the Yolo Residual block and the upSample block respectively, and $Co$ represents a $3 \times 3$ convolution with stride $= 2$.

Yolo Residual Block also has two branches, where the left side includes one $1 \times 1$ convolution and three Conv Blocks, while the right side has one $1 \times 1$ convolution and SE (Squeeze and Excitation) Layer [11]. Eventually output of both branches will be concatenated and after a $1 \times 1$ convolution, final output is obtained. It should be pointed out that applied Conv Block introduces Coordinate Convolution (Coord Conv) [22], Spatial Pyramid Pooling (SPP) [7] and DropBlock [6] optimization strategies. In particular, Coord Conv adds two channels to the input feature maps to represent the horizontal and vertical coordinates, which makes the convolution spatially aware. Since the face mask detection task in this paper is to output the bounding boxes by observing the image pixels in Cartesian coordinate space, the application of Coord Conv can solve the defects caused by ordinary convolution during spatial transformation so as to improve the space perception ability. DropBlock is a regularization method that can be deemed as a variant of Dropout, which not only drops one unit in feature map, but also neighbor regions together to prevent over-fitting problem and guarantees the efficiency of regularization work, so it is more suitable for the face mask detection task of FMD-Yolo. SPP further aggregates the features obtained from the previous convolutional layers and can adapt to changes in image target size and aspect ratio, which is proven a multi-resolution strategy and makes the deformation more robust against the target object. In this paper, SPP module is constructed by using three parallel max-pooling layers with sizes of 5, 9 and 13 respectively. Then inputs and outputs of the three layers are concatenated. Moreover, SE Layer of Yolo Residual Block enables adaptive calibration of feature channels by explicitly modeling the interdependence therein through squeeze and excitation operations, allowing the network to selectively enhance beneficial feature channels and suppress useless ones by using global information, which is realized by one average pooling and two fully-connected layers.

3.3. Loss function and post-processing methods

On the basis of cross-entropy loss function of original YoloV3 network, proposed FMD-Yolo algorithm further introduces in a localization loss, including lIoU loss and IoU-aware loss. lIoU measures the differences between predicted bounding box and corresponding ground-truth, which reflects the localization accuracy and is helpful for keeping model performance in complex environments such as overlapping multiple targets. In this paper it is calculated as:

$$
L_{\text{lIoU}} = (1 - \text{iou}^2) \cdot w_i
$$

(9)

where $w_i$ is an introduced weight and takes value of 2.5.

IoU-aware loss further adopts an additional IoU-aware prediction branch for better measure the localization accuracy [38], which can be obtained by:

$$
L_{\text{lIoU-o}} = -\text{iou} \cdot \log(\sigma(o)) - (1-\text{iou}) \cdot \log(1-\sigma(o))
$$

(10)

where $\text{iou}$ denotes the intersection over union between the predicted boxes and ground-truth, $o$ is the output of IoU-aware branch, and $\sigma$ is the sigmoid activation function. So far, a localization loss is defined as $L_{\text{loc}} = L_{\text{lIoU}} + L_{\text{lIoU-o}}$ and final loss function $L_{\text{total}}$ applied in FMD-Yolo consists of coordinates loss $L_{\text{coord}}$, objectness loss $L_{\text{obj}}$, classification loss $L_{\text{cls}}$ and localization loss $L_{\text{loc}}$:

$$
L_{\text{total}} = L_{\text{coord}} + L_{\text{obj}} + L_{\text{cls}} + L_{\text{loc}}
$$

(11)

Additionally, it is remarkable that in object detection area, non-maximum suppression (NMS) technique is able to remove redundant boxes by extracting target detection boxes with high confidence and suppressing others to finally obtain the true optimal prediction results. Applying a suitable NMS method not only facilitates improving the efficiency, but also enhances the prediction accuracy. Hence, in the inference phase of FMD-Yolo, Matrix NMS [36] is selected as the post-processing method, which uses the sorted upper triangular IoU matrix for parallelization operation to improve the detection accuracy, and can effectively prevent overlapping similar object prediction boxes from conflicting with each other.

4. Experiments and discussions

In this section, proposed FMD-Yolo is evaluated on two public human face mask detection datasets, and the experimental results are reported and discussed in detail. At first, a brief introduction of our experimental preparation is given, including information of applied databases, evaluation metrics and parameterization. Then, FMD-Yolo algorithm is comprehensively analyzed and compared with other state-of-the-art deep learning methods.

4.1. Experiment environment and settings

4.1.1. Applied datasets

In this paper, two face mask datasets have been selected to verify the effectiveness of FMD-Yolo, which are publicly available at website https://www.kaggle.com/andrewmvd/face-mask-detection and https://aistudio.baidu.com/aistudio/datasetdetail/24982. For convenience, they are denoted as MD-2 and MD-3, respectively. In particular, MD-2 contains two categories face (a1) and face _ mask (a2) referring to faces and faces with masks, respectively; whereas MD-3 includes three categories, which are without _ mask (b1), with _ mask (b2) and mask _ weared _ incorrect (b3), where the last one indicates the mask is worn incorrectly. Table 1 provides the category distribution of each database and the partition for network training.

4.1.2. Evaluation metrics

As face mask detection is essentially a classification and localization task, typical indicators have been used for evaluation including true positive (TP), true negative (TN), false positive (FP) and false negative (FN), based on which Precision and Recall are defined as follows:

$$
\text{Precision} = \frac{TP}{TP + FP}
$$

(12)

| Table 1 | Datasets information. |
|---------|-----------------------|
| Numbers | MD-2 | MD-3 |
| Training set | 6362 | 683 |
| Validation set | 1590 | 170 |
| Category a1 / b1 of training set | 9937 | 567 |
| Category a2 / b2 of training set | 3212 | 2388 |
| Category b3 of training set | - | 107 |
| Mean boxes per image | 2.067 | 4.490 |
4.2.1. Overall results and discussions

Table 2 presents the experimental settings of proposed FMD-Yolo algorithm on both MD-2 and MD-3 databases. In order to further demonstrate the superiority of FMD-Yolo, it has been compared with other state-of-the-art object detection algorithms including Faster RCNN baseline [26], Faster RCNN with FPN [19], YoloV3 [29], YoloV4 [3], RetinaNet [20], FCOS [34], EfficientDet [33] and HRNet [31]. For a fair comparison, training and validation datasets maintain the same and the best convergent model of each algorithm are selected, and all experiments are run under the deep learning framework PaddlePaddle 2.1.2 with a Tesla V100 GPU (32GB). In other words, the eight comparison algorithms and FMD Yolo have the same hardware configuration and software environment.

4.2.2. Category-wise results and analysis

In this subsection, more in-depth category-wise discussions are presented. Table 5 provides the evaluation results on MD-2 database, which ranks second on each indicator, FMD-Yolo has made an improvement of 4.6%, 1.8%, 5.5%, 10.0% and 2.3% respectively, which demonstrates the superiority of proposed FMD-Yolo in this two-category detection problem.

According to Eq. (12) and Eq. (13), precision measures the accuracy of model prediction results and recall reflects the ability of identifying TP instances in all positive samples. Notice that at a larger IoU value, improvement brought by FMD-Yolo on AP75 is even 3.7% higher than that on AP50, which indicates that FMD-Yolo satisfies the higher localization accuracy with a more significant performance advantage. It is mainly because IoU and IoU-aware loss have been adopted in the training process of FMD-Yolo that prediction results are more precise. In terms of mAR which exceeds the second one by 10%, it can be inferred that pre-processing phase of FMD-Yolo model have contributed the most, including various data augmentation methods and the Anchor Cluster algorithm combining k-means and genetic algorithms, which matches the best size settings of anchors. In addition, comparison with classical YoloV3 and YoloV4 models has demonstrated the effectiveness of designed feature extractor and fusion structure in FMD-Yolo.

In Table 4, experimental results on MD-3 database have been reported. Notice that compared with MD-2, MD-3 has an extra category which leads to more difficulty in recognition. Therefore, some algorithms such as EfficientDet have presented performance degradation while working on MD-3 than that on MD-2, as can be seen from Table 3 and Table 4. However, proposed FMD-Yolo still performs best, which improves mAP, AP50, AP75, mAR and AR50 by 3.9%, 3.8%, 3.9%, 7.1%, and 0.2% respectively, compared with corresponding model that ranks second. It can be concluded that FMD-Yolo algorithm has strong robustness that it can present overwhelming advantages in both two-category and three-category face mask detection tasks, which is suitable for application in more complex and variable real-world scenarios.

In order to further intuitively analyze and compare the performance of each algorithm in face mask detection task, Fig. 9 shows the overall P-R curves for each algorithm on MD-2 and MD-3 datasets where IoU = 0.5 is selected as the threshold to divide positive and negative samples. For any two models, if the P-R curve of one is completely wrapped by that of another, the latter is deemed better. In addition, the superiority of individual models can be estimated based on the area enclosed by P-R curve and coordinate axis or based on the break even point (BEP). The so-called BEP is the coordinate value of the P-R curve of each model when the precision and recall are equal, which is the intersection of P-R curve and the precision = recall line. As illustrated in Fig. 9, both area under P-R curve and BEP of FMD-Yolo method are significantly better than those of other algorithms regardless of the database, which further indicates that FMD-Yolo does have outstanding performance in face mask detection applications.

| Methods                  | Evaluation metrics |
|--------------------------|--------------------|
|                           | mAP   | AP50 | AP75 | mAR  | AR50 |
| Faster RCNN baseline     | 0.565 | 0.869| 0.668| 0.623| 0.895|
| Faster RCNN with FPN     | 0.597 | 0.886| 0.713| 0.655| 0.900|
| Yolo V3                  | 0.574 | 0.888| 0.659| 0.670| 0.929|
| Yolo V4                  | 0.599 | 0.899| 0.718| 0.661| 0.920|
| RetinaNet                | 0.616 | 0.887| 0.726| 0.664| 0.909|
| FCOS                     | 0.605 | 0.894| 0.710| 0.674| 0.932|
| EfficientDet             | 0.613 | 0.870| 0.726| 0.665| 0.910|
| HRNet                    | 0.618 | 0.902| 0.745| 0.671| 0.916|
| FMD-Yolo (ours)          | 0.664 | 0.920| 0.800| 0.774| 0.955|

| Methods                  | Evaluation metrics |
|--------------------------|--------------------|
|                           | mAP   | AP50 | AP75 | mAR  | AR50 |
| Faster RCNN baseline     | 0.521 | 0.821| 0.604| 0.601| 0.889|
| Faster RCNN with FPN     | 0.536 | 0.846| 0.551| 0.615| 0.907|
| Yolo V3                  | 0.507 | 0.824| 0.585| 0.609| 0.927|
| Yolo V4                  | 0.525 | 0.843| 0.591| 0.587| 0.897|
| RetinaNet                | 0.489 | 0.792| 0.536| 0.579| 0.875|
| FCOS                     | 0.521 | 0.796| 0.558| 0.628| 0.916|
| EfficientDet             | 0.454 | 0.699| 0.518| 0.569| 0.841|
| HRNet                    | 0.512 | 0.802| 0.556| 0.571| 0.845|
| FMD-Yolo (ours)          | 0.575 | 0.884| 0.643| 0.699| 0.929|
where precision $cAP_{50}$ and average recall $cAR_{50}$ at $IoU = 0.5$ of categories $a1$ and $a2$ are reported. From the results, it is found that all methods have better performance on detecting $a2$ category. In particular, FMD-Yolo presents a 10.8% higher $cAP_{50}$ and 8.7% better $cAR_{50}$ value on $a2$ than those of $a1$. But even regarding to the $a1$ class, FMD-Yolo also performs the best results, which outperforms the second one by 3.2% and 4.1% on $cAP_{50}$ and $cAR_{50}$, respectively.

Similarly, Table 6 reports the category-wise evaluation results on MD-3 dataset. Obviously, the category $b3$ is the most difficult one for accurate detection. FMD-Yolo takes the first place in terms of precision on each category, specifically on $b3$; the $cAP_{50}$ value is even 3.3% higher than the second one, which demonstrates the superiority of FMD-Yolo in complicated detection work. As for $cAR_{50}$, FMD-Yolo ranks fourth on category $b3$ and best on the others. Slightly loss of $cAR_{50}$ value on $b3$ may be resulted from smaller number of this category in the validation set, and the fact that FMD-Yolo model pays more attention to higher quality detection with larger IoU. But still, FMD-Yolo presents a satisfactory performance on this three-category detection problem.

Furthermore, as shown in Fig. 10, the P-R curves of FMD-Yolo are illustrated on categories $a1$, $a2$, $b1$, $b2$, $b3$ in MD-3 dataset. Notice that the values of BEP (break even point) and the area under the curves are satisfied with $a2 > a1$ and $b2 > b1 > b3$, which verifies the reasonability of above discussions. It is also worth noting that both category $a2$ and $b2$ belong to the face wearing mask type, where FMD-Yolo performs well with $cAP_{50}$ of 97.4% and 93.9% respectively, and this indicates that in real-world application FMD-Yolo is capable of accurately identifying people who have worn a mask.

### Table 5
Performance comparison on each category of the MD-2 dataset.

| Methods                        | $a1$ | $a2$ | $b1$ | $b2$ | $b3$ |
|-------------------------------|------|------|------|------|------|
| $cAP_{50}$                    | 0.776| 0.811| 0.962| 0.979|      |
| $cAR_{50}$                    | 0.811| 0.824| 0.961| 0.975|      |
| Faster RCNN baseline          | 0.804| 0.863| 0.971| 0.994|      |
| Faster RCNN with FPN          | 0.829| 0.855| 0.969| 0.985|      |
| Yolo V3                       | 0.796| 0.825| 0.978| 0.993|      |
| Yolo V4                       | 0.819| 0.870| 0.969| 0.994|      |
| RetinaNet                     | 0.762| 0.826| 0.979| 0.993|      |
| EfficientDet                  | 0.834| 0.852| 0.971| 0.980|      |
| HRNet                         | 0.866| 0.911| 0.974| 0.998|      |
| FMD-Yolo (ours)               |      |      |      |      |      |

### Table 6
Performance comparison on each category of the MD-3 dataset.

| Methods                        | $b1$ / $b2$ / $b3$ |
|-------------------------------|------------------|
| $cAP_{50}$                    | 0.817 / 0.817 / 0.817 |
| $cAR_{50}$                    | 0.730 / 0.730 / 0.730 |
| Faster RCNN baseline          | 0.838 / 0.838 / 0.838 |
| Faster RCNN with FPN          | 0.864 / 0.864 / 0.864 |
| Yolo V3                       | 0.840 / 0.840 / 0.840 |
| Yolo V4                       | 0.828 / 0.828 / 0.828 |
| RetinaNet                     | 0.774 / 0.774 / 0.774 |
| FCOS                          | 0.842 / 0.842 / 0.842 |
| EfficientDet                  | 0.696 / 0.696 / 0.696 |
| HRNet                         | 0.825 / 0.825 / 0.825 |
| FMD-Yolo (ours)               | 0.892 / 0.939 / 0.920 |

5. Conclusion

In this paper, an efficient automatic face mask recognition and detection framework FMD-Yolo and corresponding algorithm are proposed. In particular, a modified Res2Net structure Im-Res2Net-101 serves as the backbone to extract features with rich receptive fields from the input. Subsequently there follows a feature fusion component En-PAN, which is a novel path aggregation network and primarily consists of
Yolo Residual Block, SPP, Coord Conv, SE Layer blocks. In En-PAN, high-level semantic information and low-level details can be sufficiently merged so that highly distinctive feature with strong robustness can be constructed. In addition, the localization loss function including IoU and IoU-aware loss is applied for training FMD-Yolo, and parallel computing method Matrix NMS is applied in the post-processing stage, which greatly enhances the model inference efficiency. Benchmark evaluations on two publicly available datasets and comparison with other state-of-the-art object detection algorithms have demonstrated the effectiveness and superiority of FMD-Yolo. In future work, we will further enhance the generalization ability of FMD-Yolo so as to enable applications in various real-world detection scenarios, which can be achieved by designing a more efficient feature fusion model with channel and spatial self-attention mechanism.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] A. Basri, B.M. Henry, L. Pighi, L. Leone, G. Lippi, Evaluation of indoor hospital acclimatization of body temperature before COVID-19 fever screening, J. Hosp. Infect. 112 (2021) 127–128.
[2] V. Bazarevsky, V. Karytnikov, A. Vakunov, K. Ravendran, M. Grundmann, Blazeface: Sub-Millisecond Neural Face Detection on Mobile Gpus, 2019 arXiv preprint, arXiv:1907.05047.
[3] A. Bochkovskiy, C. Wang, H. Liao, YOLOv4: optimal speed and accuracy of object detection, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020.
[4] C. Chiuangrueng, C. Sequin, Optimal adaptive k-means algorithm with dynamic adjustment of learning rate, IJCNN-91-Seattle Int. Jt. Conf. Neural Netw. 1 (1991) 835–862.
[5] S. Gao, M. Cheng, K. Zhao, X. Zhang, M. Yang, P. Torr, Res2Net: a new multi-scale backbone architecture, IEEE Trans. Pattern Anal. Mach. Intell. 43 (2019) 652–662.
[6] G. Ghiasi, T. Lin, Q. Le, DropBlock: a regularization method for convolutional networks, Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS) 2018, pp. 10750–10760.
[7] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, IEEE Trans. Pattern Anal. Mach. Intell. 37 (9) (2015) 1904–1916.
[8] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4, 2016, pp. 770–778.
[9] H. Huang, Z. Liu, L. van der Maaten, K. Weinberger, Deeply connected convolutional networks, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017, pp. 2261–2269.
[10] K. Hendry, Chen, automatic license plate recognition via sliding-window darknet-YOLO deep learning, Image Vis. Comput. 87 (2019) 47–56.
[11] J. Hu, L. Shen, S. Albanie, G. Sun, E. Wu, Squeeze-and-excitation networks, IEEE Trans. Pattern Anal. Mach. Intell. 42 (8) (2020) 2011–2021.
[12] D. Jain, Z. Zhang, K. Huang, Random walk-based feature learning for micro-expression recognition, Pattern Recogn. Lett. 115 (2018) 92–100.
[13] D. Jain, Z. Zhang, K. Huang, Multi-angle optimal pattern-based deep learning for automatic facial expression recognition, Pattern Recogn. Lett. 139 (2020) 157–165.
[14] A. Krizhevsky, I. Sutskever, G. Hinton, ImageNet classification with deep convolutional neural networks, Neural Inf. Process. Syst. (2012) 1097–1105.
[15] U. Mahlib, S. Sarkar, R. Chellappa, Partial face detection in the mobile domain, Image Vis. Comput. 82 (2019) 1–17.
[16] S. Meisel, K. Devi, S. Maheswari, J. Menaka, Real time data analysis of face mask detection and social distance measurement using matlab, Mater. Today Proc. (2021).
[17] H. Law, J. Deng, CornerNet: detecting objects as paired keypoints, Proceedings of the European Conference on Computer Vision (ECCV) 2018, pp. 734–750.
[18] M. Liao, H. Liu, X. Wang, X. Hu, Y. Huang, X. Liu, K. Brenan, J. Mecha, M. Nirmalan, J. Lu, A technical review of face mask wearing in preventing respiratory COVID-19 transmission, Curr. Opin. Colloid Interface Sci. 52 (2021).
[19] T. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017, pp. 936–944.
[20] T. Lin, P. Goyal, R. Girshick, K. He, P. Dollar, Focal loss for dense object detection, IEEE Trans. Pattern Anal. Mach. Intell. 42 (2) (2020) 318–327.
[21] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, A. Berg, SSD: single shot multibox detector, Proceedings of the European Conference on Computer Vision (ECCV) 2016, pp. 21–37.
[22] R. Liu, J. Lehman, P. Molino, F. Such, E. Frank, A. Sergeev, J. Yosinski, An intriguing failing of convolutional neural networks and the coordconv solution, Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS) 2018, pp. 9626–9635.
[23] M. Loey, G. Manogaran, M. Taha, N. Khalifad, Fighting against COVID-19: a novel deep learning model based on YOLO-v2 with ResNet-50 for medical face mask detection, Sustain. Citi Soc. 65 (2021).
[24] A. Rahmani, S. Mirmahaleh, Coronavirus disease (COVID-19) prevention and treatment methods and effective parameters: a systematic literature review, Sustain. Citi Soc. 64 (2020).
[25] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. 39 (6) (2017) 1137–1149.
[26] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: unified, real-time object detection, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016, pp. 779–788.
[27] J. Redmon, A. Farhadi, YOLO9000: better, faster, stronger, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017, pp. 6517–6525.
[28] J. Redmon, A. Farhadi, YOLOv3: An Incremental Improvement, 2018 arXiv: 1804.02767, [online] Available: https://arxiv.org/abs/1804.02767.
[29] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, Int. Conf. Learn. Represent. 2015 (2015) 1–14.
[30] K. Sun, B. Xiao, D. Liu, J. Wang, Deep high-resolution representation learning for human pose estimation, 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2019, pp. 5603–5610.
[31] S. Singh, U. Ahuja, M. Kumar, K. Kumar, M. Sachdeva, Face mask detection using YOLOv3 and Faster R-CNN models: COVID-19 environment, Multimedia Tools Appl. 80 (2021) 19753–19768.
[32] M. Tan, R. Pang, Q. Le, EfficientDet: scalable and efficient object detection, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2020, pp. 10781–10790.
[33] Z. Tian, C. Shen, H. Chen, T. He, FCOS: fully convolutional one-stage object detection, 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 9626–9635.
[34] X. Wang, R. Girshick, A. Gupta, K. He, Non-local neural networks, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition 2018, pp. 7794–7803.
[35] X. Wang, R. Zhang, K. Tao, Lei. Li, C. Shen, SOLOv2: dynamic and fast instance segmentation, Adv. Neural Inf. Process. Syst. 33 (2020) 17721–17732.
[36] D. Wu, S. Lv, M. Jiang, H. Song, Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments, Comput. Electron. Agric. 178 (2020).
[37] S. Wu, X. Li, X. Wang, IoU-aware single-stage object detector for accurate localization, Image Vis. Comput. 97 (2020).
[38] X. Ye, D. Hong, C. Chen, P. Hisio, L. Fu, A two-stage real-time YOLOv2-based road marking detector with lightweight spatial transformation-invariant classification, Image Vis. Comput. 102 (2020).
[39] F. Yu, D. Wang, E. Shelhamer, T. Darrell, Deep layer aggregation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018, pp. 2403–2412.
[40] N. Zeng, H. Zhang, B. Song, W. Liu, Y. Li, A.M. Dobaie, Facial expression recognition via learning deep sparse autoencoders, Neurocomputing 273 (2018) 643–649.
[41] N. Zeng, H. Li, Y. Peng, A New Deep Belief Network-based Multi-task Learning for Diagnosis of Alzheimers’ Disease, Neural Computing and Applications, 2021 2021, pp. 9300–9308.