Hybrid Transfer Learning and Broad Learning System for Wearing Mask Detection in the COVID-19 Era

Bingshu Wang, Yong Zhao, *Member, IEEE*, and C. L. Philip Chen, *Fellow, IEEE*

***Abstract*—In the era of Corona Virus Disease 2019 (COVID-19), wearing a mask can effectively protect people from infection risk and largely decrease the spread in public places, such as hospitals and airports. This brings a demand for the monitoring instruments that are required to detect people who are wearing masks. However, this is not the objective of existing face detection algorithms. In this article, we propose a two-stage approach to detect wearing masks using hybrid machine learning techniques. The first stage is designed to detect candidate wearing mask regions as many as possible, which is based on the transfer model of Faster_RCNN and InceptionV2 structure, while the second stage is designed to verify the real facial masks using a broad learning system. It is implemented by training a two-class model. Moreover, this article proposes a data set for wearing mask detection (WMD) that includes 7804 realistic images. The data set has 26403 wearing masks and covers multiple scenes, which is available at "https://github.com/BingshuCV/WMD." Experiments conducted on the data set demonstrate that the proposed approach achieves an overall accuracy of 97.32% for simple scene and an overall accuracy of 91.13% for the complex scene, outperforming the compared methods.**

***Index Terms*—Broad learning system (BLS), Corona Virus Disease 2019 (COVID-19), transfer learning, wearing mask detection (WMD).**

I. Introduction

SINCE the first patient infected by Corona Virus Disease 2019 (COVID-19) has been identified in 2019, the virus spread the world very fast. It is quickly declared as a global pandemic by the World Health Organization. By the end of March 4, 2021, more than 115.22 millions of humans were infected by the virus, and more than 2.56 millions of people were dead by the virus or the disease caused by COVID-19 across the globe, with more being added every day, according to the COVID-19 dashboard released by the Johns Hopkins University of Medicine [1].

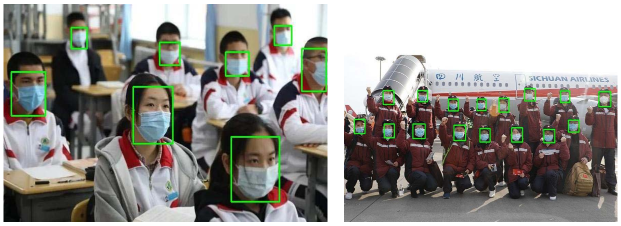

Fig. 1. Some results of WMD by the proposed approach.

In the fight against the pandemic coronavirus, many doctors and epidemiologists hold a view that the transmission of COVID-19 can be effectively restricted if people wear a mask, keep social distance, wash hands, and active quarantine. It has been verified to be very effective that wearing a mask is one of the main precautionary measures for the public [2]. As a result, people are encouraged, even forced by laws and rules, to wear a mask when they need to enter public areas, such as supermarkets, hospitals, and airports [3], [4].

To beat COVID-19, governments need to guide and monitor people in public places, for example, noncontact temperate measurement through monitoring instruments [5]–[8]. However, monitoring a large number of people in many places is a challenging task. It involves the detection of wearing masks. Most of the monitoring instruments lack this function, which can be implemented by the integration between monitoring devices and machine learning techniques.

The objective of this article is to design an approach to detect people who wear a mask, as illustrated in Fig. 1. The wearing mask is the primary focus in this article because wearing a mask can effectively protect one from the infection risks and largely decrease the spread in public places. Given an input image, the wearing mask regions will be labeled in the output image by the developed deep transfer learning model [9], [10] and the broad learning system (BLS) [11], [12].

To realize the objective, some problems should be addressed. The first problem is that facial masks have various

Manuscript received January 20, 2021; revised March 5, 2021; accepted March 16, 2021. Date of publication March 30, 2021; date of current version May 7, 2021. This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant G2020KY05113; in part by the National Key Research and Development Program of China under Grant 2019YFA0706200 and Grant 2019YFB1703600; in part by the National Natural Science Foundation of China under Grant 61702195, Grant 61751202, Grant U1813203, Grant U1801262, and Grant 61751205; in part by the Science and Technology Major Project of Guangzhou under Grant 202007030006; in part by The Science and Technology Development Fund, Macau SAR, under Grant 079/2017/A2 and Grant 0119/2018/A3; and in part by the Multiyear Research Grants of University of Macau. The Associate Editor coordinating the review process was Lihui Peng. *(Corresponding author: C. L. Philip Chen.)*

Bingshu Wang is with the School of Software, Northwestern Polytechnical University, Suzhou 215400, China (e-mail: wangbingshu@nwpu.edu.cn).

Yong Zhao is with the Key Laboratory of Integrated Microsystems, School of Electronic and Computer Engineering, Peking University Shenzhen Graduate School, Shenzhen 518055, China (e-mail: zhaoyong@pkusz.edu.cn).

C. L. Philip Chen is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510641, China, and also with the Faculty of Science and Technology, University of Macau, Macau (e-mail: philip.chen@ieee.org).

Digital Object Identifier 10.1109/TIM.2021.3069844

styles, such as orientations and stochastic noise. It easily results in the lack of facial features and causes the failures of even state-of-the-art face detection algorithms or models [13]–[17]. Second, although many face data sets have been created for face detection [18]–[20], it still lacks data sets for wearing mask detection (WMD) in realistic scenes. All these factors lead to WMD a challenging task.

In this article, we propose a two-stage method to detect masked faces. This can be regarded as rewarding support for special face detection. The main contributions include the following.

1) This article proposes a two-stage method for WMD. It explores the Faster_RCNN framework with InceptionV2 as a predetection stage and uses BLS as a verification stage. It is verified to be effective by the combination of two stages.
2) We create a novel data set for WMD from scenes of struggling against the pandemic. It has 7804 realistic images with 26 403 masked faces, varying from easy to hard. The data set will be available to the public soon.
3) Quantitative and visual experiments on the data set indicate the designed method's effectiveness, with an overall 94.19% accuracy outperforming the compared methods.

II. Related Work

In the past years, facial mask detection is attracting more and more attention. We will give a brief review of these detection techniques from two parts: the facial mask detection methods and the related data sets.

A. Facial Mask Detection Methods

Traditional methods usually used handcrafted features for face detection. One of the most used features is a haar-like feature, which can be trained by the AdaBoost algorithm for face detection [21]. Dewantara and Rhamadhaningrum [22] exploited the AdaBoost algorithm with Haar, LBP, and HOG features to train a cascade classifier for multipose masked face detection. It is reported that using the Haar-like feature achieves a higher accuracy of 86.9%. Petrovic and Kocic [23] introduced an affordable IoT-based system for COVID-19 indoor safety. The mask detection method is based on three libraries in OpenCV: frontal face, mouth, and nose classifiers. It detects face first and then verifies it using the characteristic of mouth and nose.

Deep learning methods based on convolutional neural networks (CNNs) have achieved great success in the field of object detection [9], [10], [24]. Recently, some techniques have been applied to the field of facial mask detection. Ge *et al.* [18] proposed LLE-CNNs for masked face detection. It includes a proposal module to extract candidate facial regions, an embedding module to turn a high-dimensional descriptor into a similarity by using the locally linear embedding (LLE) algorithm, and the verification module to identify candidate facial regions and refine their positions. It is reported that the method outperforms six algorithms by more than 15%. Jiang and Fan [25] designed a face mask detector: RetinaFaceMask. It is comprised of three parts: a feature pyramid network to fuse multiple semantic feature maps, a novel context attention module to concentrate on detecting masked faces, and a cross-class object removal algorithm. The method [26] explored a transfer learning of inception structure to detect mask faces. The approach [27] exploited a deep learning architecture to detect masks and faces and applied it to the CCTV system to help the authority to take necessary actions. It achieves 98.7% accuracy on a test set with 308 images. However, it can only process a fixed size of 64 × 64 images under simple scenes.

Loey *et al.* [28] proposed a hybrid deep transfer learning model and machine learning method for masked face detection. It utilized ResNet50 [29] to extract feature maps and employed decision trees, support vector machine (SVM), and ensemble algorithm for recognition. Finally, SVM is selected as the classifier and achieves 99.49% accuracy on the given data set. Qin and Li [30] combined image super-resolution and classification networks as a new condition identification of face mask-wearing. Experimental results indicate that the addition of image super-resolution can improve the classification accuracy by 1.5% than the deep learning method without a super-resolution module. Militante and Dionisio [31] used a VGG16 [32] structure for the face mask and physical distancing detection, which can send out an alarm and a voice notice if one does not wear a mask or observe the social distance. The method reached a 97% accuracy on fixed size of 224 × 224 images. The approach designed by Loey *et al.* [33] utilized ResNet-50 and YOLOv2 techniques to train a model for medical masked face detection. By introducing mean IoU to estimate the best number of anchor boxes, it achieves an average precision of 81%.

In addition, there are many other techniques [34]–[38] developed for face and WMD. Accurate locations of facial masks can improve the accuracy of face recognition algorithms [39]–[43]. In this article, our main concern is WMD, as shown in Fig. 1.

B. Related Data Sets

Some data sets were created for occluded face or facial mask detection. Ge *et al.* [18] created an occluded face detection data set from the Internet by keywords search "face mask occlusion cover." It consists of 25 876 train images and 4935 test images. Each masked face has multiple property labels: face location, eye location, face direction, occlusion degree, and occlusion type. Wang *et al.* [44] proposed a Real World Masked Face data set. It encompasses 4342 images and these images are divided three groups according to image size: smaller than 256 × 256; a fixed size of 256 × 256 and most of the images are distorted; different sizes of images without distortion. However, the data set does not provide label information.

One simulated masked face data set was created by [45]. It includes 826 masked face images and 825 face images. Each image only has one mask with a large size, which indicates that it is a simple data set. The authors of [46] created a data set by selecting images from MAFA [18] and

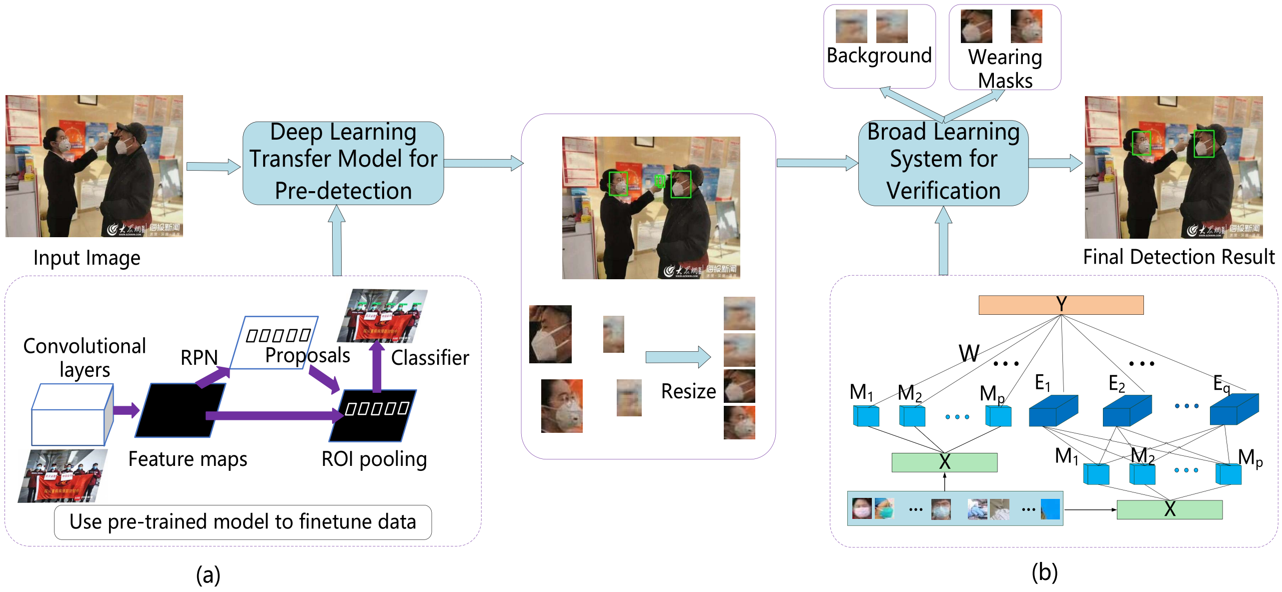

Fig. 2. Framework of the proposed approach. Two stages, including predetection and verification, are designed. In the first stage, a detection model is trained to detect candidate facial regions. In the second stage, a classification model is trained to distinguish realistic facial masks from the background. (a) Train predetection model. (b) Train verification model.

WIDER FACE [47]. They corrected some errors and provided labeled information. Adnane Cabani *et al.* [48] designed a MaskedFace-Net including face region detection, facial landmarks detection, mask-to-face mapping, and manual image filtering to synthesize a total of 137 016 masked face images with a size of 1024 × 1024. It contains about 49% correctly masked faces and 51% incorrectly masked faces, which is the biggest data set for the wearing mask classification task.

In summary, most of the abovementioned data sets were created from simple scenes or synthesis, lacking labels or a sense of reality to some degree. In this article, we create a data set with labels where the original images are from realistic scenes of fighting against COVID-19. Most of the images have a variety of sizes and orientations.

III. Proposed Method

The flowchart of our method is illustrated in Fig. 2. It includes two stages: predetection and verification. The first is to use a trained Faster_RCNN model to detect candidate facial masks, and in the second stage, a classifier trained by BLS is applied to remove background regions.

A. Deep Transfer Learning Model for Predetection

We develop a deep transfer learning model for the predetection of wearing masks. Wearing mask is the region of interest (ROI). Detecting those ROIs requires a model that can propose accurate and effective regions. The region proposal network (RPN) introduced by the Faster_RCNN framework can provide a series of candidate regions [9]. Moreover, the framework offers a powerful new way to generate the regions with their classification scores after a straightforward process. Thus, it is a good choice as a predetection module for our task. The primary principle of this stage is to locate ROIs as many as possible. The predetection covers four steps as follows.

1) *Extract Feature Maps:* A series of convolutional operations followed by relu and pooling layers is designed to extract feature maps. The last layer of the feature map will be used by subsequent RPN and ROI pooling steps.
2) *Generate Proposals:* It is implemented by RPNs, which aims to produce sufficient proposals for selection and is called anchor generator. Each point of the image can be regarded as an anchor. Four scales (0.25, 0.5, 1.0, 2.0) and three aspect ratios (0.5, 1.0, 2.0) are set empirically, which ensures that the network generates enough boxes. RPN includes the box-regression layer and the box-classification layer. The goal of the box-regression layer is to adjust the positions of proposals, while the goal of the box-classification layer is to determine whether a box belongs to an object or background.
3) *Obtain Fixed Dimension of Feature Map:* This step is realized by ROI-pooling. It receives a feature map from convolutional layers (step 1) and the proposals generated by RPN (step 2) and produces a fixed-size feature map from every ROI by max-pooling operation. It solves the problem of fixed feature map requirements for subsequent classification and regression. The fixed dimension of the feature map never relies on input sizes; it merely depends on the layer's parameters.
4) *Object Classification and Location Regression:* This step receives a fixed dimension of the feature map and outputs the probability of classes. Meanwhile, the bounding box regression is carried out to obtain accurate locations of boxes. Predicted objects and their locations are generated finally.

It should be noted that RPN is an effective way to provide sufficient proposals. It helps the detection model to reach a good tradeoff between accuracy and computations. After a straightforward pass of four steps, many candidate regions are generated. The loss function for an image is defined as

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{\text{cls}}}\sum_{i}^{n} L_{\text{cls}}(p_i, p_i^*) + \lambda\frac{1}{N_{\text{reg}}}\sum_{i}^{n} p_i^* L_{\text{reg}}(t_i, t_i^*) \quad (1)$$

where i is the index of an anchor, and p_i is the predicted probability belonging to wearing mask. p_i^* represents the ground truth; it is 1 if the anchor is positive and is 0 if the anchor is negative. t_i is the predicted coordinates of a box, and t_i^* is the ground-truth coordinates of a positive anchor. L_{cls} is the classification loss, and L_{cls} is the regression loss. $p_i^* L_{\text{reg}}$ means that only positive anchors are computed. Classification loss and regression loss are normalized by terms N_{cls} and N_{reg}. λ is denoted as a weighted balancing

Generally, traditional convolution networks used in [9] have higher complexity and computation. When convolution networks reduce dimensions too many, it may cause information loss, which is called a representational bottleneck. To address the issue, the InceptionV2 structure is designed [10], which is enhanced from the original Inception module first proposed by Szegedy *et al.* [49]. It aims to reduce representational bottlenecks and decrease computational complexity.

Fig. 3 elucidates three modules of InceptionV2 structure. Module A is designed by factorizing a 5 × 5 convolution to two 3 × 3 convolutions, which obeys spatial aggregation principle, as said in [10]. It can reduce $(5 \times 5 - (3 \times 3 + 3 \times 3)/5 \times 5) = 28\%$ of computation by the factorization, leading to a boost in performance.

What is more, spatial factorization into asymmetric convolutions is another strategy to reduce complexity. Module B illustrates that a $n \times n$ can be factorized by a combination of $1 \times n$ and $n \times 1$. For example, a 3×3 convolution is replaced by a 1×3 convolution and a 3×1 convolution orderly. This solution of two layers is $(3 \times 3 - (1 \times 3 + 3 \times 1)/3 \times 3) = 33\%$ cheaper than that of one layer.

Especially, filter banks are expanded to avoid representational bottlenecks. It means wider than deeper to promote the high-dimensional representations, which helps process locally within a network. In summary, three modules in Fig. 3 are utilized in our predetection model.

In this article, our predetection model is transferred from a pretrained detection model on the COCO data set [50], [51]. The training data set for mask detection is labeled by a tool named "LabelImg" [52], as shown in Fig. 2. Candidate regions with boxes and scores can be generated in the predetection stage.

B. Broad Learning System for Verification

This stage is to verify the predetection results, whether they are objects or background. Herein, BLS is exploited. It is built up in the form of a flat neural network, which is the main characteristic of BLS [11], [53]. For classification, input images are first converted into random feature nodes in the form of "mapped features"; then, all the mapped features are expanded to feature nodes in the form of "enhanced features." This is regarded as a considerable means to explore essential features from the wide dimension.

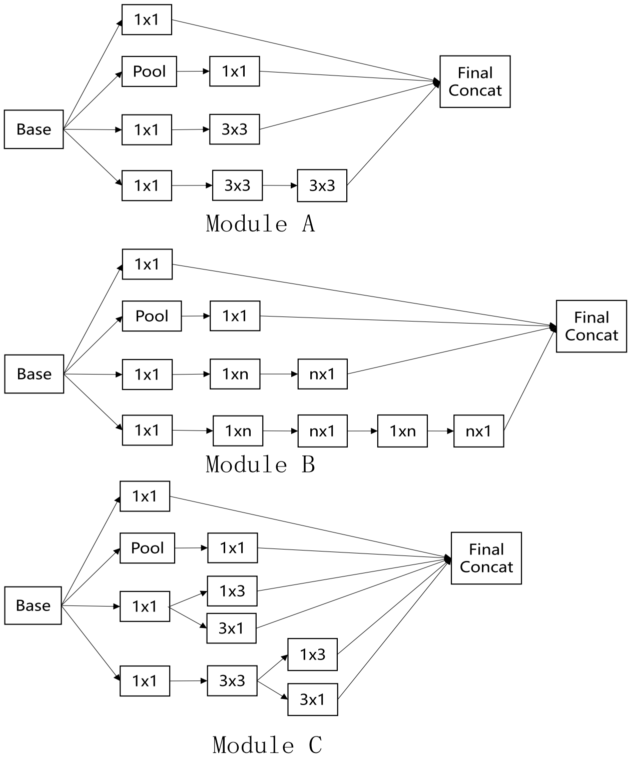

Fig. 3. Three modules of the InceptionV2 structure.

Herein, we define the ith group of mapped features by

$$M_i = \varphi(XW_{m_i} + \beta_{m_i}), \quad i = 1, 2, \ldots, p \quad (2)$$

where W_{m_i} and β_{m_i} are generated weights randomly from specified distribution, and φ is a mapping function. To explore more essential features, the mapped features are fine-tuned by sparse autoencoder [54]. After a series of mapping operation, p groups of mapped features are generated, which can be expressed by a concatenation of $M^p \equiv [M_1, \ldots, M_p]$. Then, all the processed features are expanded to enhanced features

$$E_j = \sigma(M^p W_{e_j} + \beta_{e_j}), \quad j = 1, 2, \ldots, q \quad (3)$$

where σ is a nonlinear activation function, e.g., tansig. The terms W_{e_j} and β_{e_j} are defined as weights generated from given distribution. The first q groups of enhanced nodes are expressed by $E^q \equiv [E_1, \ldots, E_q]$.

All the mapped features and enhanced features are jointly connected to the output layer

$$Y = [M_1, M_2, \ldots, M_p, E_1, E_2, \ldots, E_q]W = [M^p|E^q]W \quad (4)$$

where W is the weights of whole network, and the term Y represents output. In practice, the selections of parameters p and q rely on the complexity of task and requirement of computation cost. The weight W can be derived from $W \triangleq [M^p|E^q]^+ Y$, where $[M^p|E^q]^+$ can be computed by the pseudo inverse of ridge regression approximation.

In particular, when a designed BLS cannot learn a task well, an effective solution is to add mapped feature or

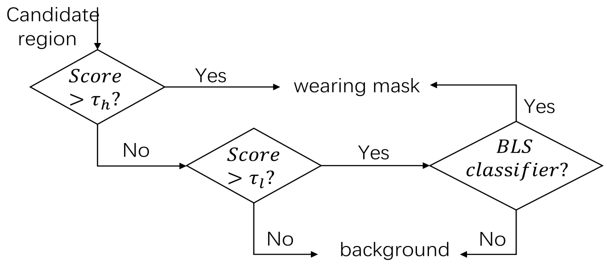

Fig. 4. Verification process by the BLS classifier based on box score.

enhanced feature. This is treated as an incremental learning, which makes BLS structure built up without retraining from the scratch. When adding a mapped feature $M_{p+1} = \varphi(XW_{m_{p+1}} + \beta_{m_{p+1}})$, the concatenation of mapped features become $M^{p+1} \equiv [M_1, \ldots, M_{p+1}]$. As a consequence, the enhanced feature nodes can be updated as $E^{ex_j} \triangleq [\sigma(M^{p+1}W_{ex_1} + \beta_{ex_1}), \ldots, \sigma(M^{p+1}W_{ex_j} + \beta_{ex_j})]$, where W_{ex_j} and β_{ex_j} $j = 1, 2, \ldots, q$ are random weights. If enhanced feature is added, the new enhanced feature node can be expressed by $E_{q+1} = \sigma(M^{p+1}W_{ex_{q+1}} + \beta_{ex_{q+1}})$.

Herein, we denote $A_p^q \triangleq [M^p|E^q]$ and $A_{p+1}^{q+1} \triangleq [M^p|M_{p+1}|E^{ex_q}|E_{q+1}]$. The updated weights can be calculated by

$$(A_{p+1}^{q+1})^+ = \begin{bmatrix} (A_p^q)^+ - DB^T \\ B^T \end{bmatrix} \tag{5}$$

$$W_{p+1}^{q+1} = \begin{bmatrix} W_p^q - DB^TY \\ B^TY \end{bmatrix} \tag{6}$$

where $D = (A_p^q)^+[M_{p+1}|E^{ex_q}|E_{q+1}]$

$$B^T = \begin{cases} (C)^+ & \text{if } C \neq 0 \\ (1 + D^TD)^{-1}D^T(A_p^q)^+ & \text{if } \ C = 0 \end{cases} \tag{7}$$

and $C = [M_{p+1}|E^{\mathrm{ex}_q}|E_{q+1}] - A_p^q D$.

As can be seen from above derivations, this update of weight benefits BLS in a fast speed and ensures training efficiency. This characteristic makes BLS have the flexibility and adaptability to various application scenes. In terms of wearing mask classification, a slight BLS model is adequate for a simple scene, such as indoor conditions. For complex scenes, one needs to train BLS judiciously to meet the application requirements. In this article, the detailed processing in the second stage is presented in Fig. 4. For a candidate region, if its score is larger than τ_h, it will be regarded as a wearing mask confidently. If its score is less than τ_l, it will be regarded as background definitely. Only those regions whose scores are between τ_l and τ_h will be verified by the BLS classifier. The process does not consider those candidate regions with low scores. Thus, it can reduce computation costs and is effective for verification.

IV. Experimental Results

In this section, we present experimental results and detailed analysis for our approach and other methods. The compared methods are all deep learning algorithms: MobileNet [37], a commercial software called PaddlePaddle [55], and two Faster_RCNN models: Faster_RCNN-ResNet50 and Faster_RCNN-InceptionV2 [50], and SSD-InceptionV2 [24]. Details will be illustrated from four parts: self-built wearing mask data set; evaluation metrics and parameter setting; quantitative analysis; and visual results and discussion.

TABLE I

Detailed Description for Our WMD Data Set

WMD Dataset	***Train***	***Val***	***Test***	***Sum***
Image Number	5410	800	1594	7804
Mask Number	17654	1936	6813	26403

TABLE II

Detailed Description for Test Set

Test Set	***DS1***	***DS2***	***DS3***	***Sum***
Image Number	500	500	594	1594
Mask Number	500	1458	4855	6813

A. Wearing Mask Data Set

As illustrated in Fig. 2, a data set of wearing masks is created, which includes two parts: the WMD data set and the wearing mask classification (WMC) data set. The WMD is used to train a detection model. The WMC is used to train a two-class classifier. Some of the wearing mask samples in WMC are from WMD. They will be introduced orderly.

All the images for WMD data set are collected from the Internet with different sizes and styles. Most of them come from the realistic scenes of COVID-19 prevention, for example, the communities, hospitals, sickrooms, railway stations, meeting rooms, construction sites, factories, and so forth. Some samples are shown in Fig. 5. There are three steps in the process of creating data set. First, coarse images are cropped from news reports, videos, and other similar small data sets. Second, some bad samples are removed, and only the samples having facial masks are chosen. Third, a label tool named "LabelImg" is exploited to mark the rectangular positions of wearing masks. By the operation repeatedly, 7804 images with 26 403 labeled masks are generated. The data set is summarize in Table I. It is open to the public: "https://github.com/BingshuCV/WMD."

Especially, the test set is divided into three parts according to task difficulty: DS1, DS2, and DS3. Table II gives the statistical information. For DS1, each image has only one person, i.e., only one wearing mask is included. For DS2, the number of wearing masks for every image is from two to four. Each image in DS3 has five and more wearing masks with small sizes. In summary, multiple scenes are covered in a total number of 1594 images varying from easy to hard.

Moreover, some samples in the WMC data set are shown in Fig. 6. WMC includes two classes: wearing masks and background. Wearing mask samples are extracted from the train set, as shown in Table I. To be realistic, most of the background samples are also extracted from the WMD data set, and some are cropped from the Internet. In total, 19 590 mask samples and 18 555 background samples are obtained for training.

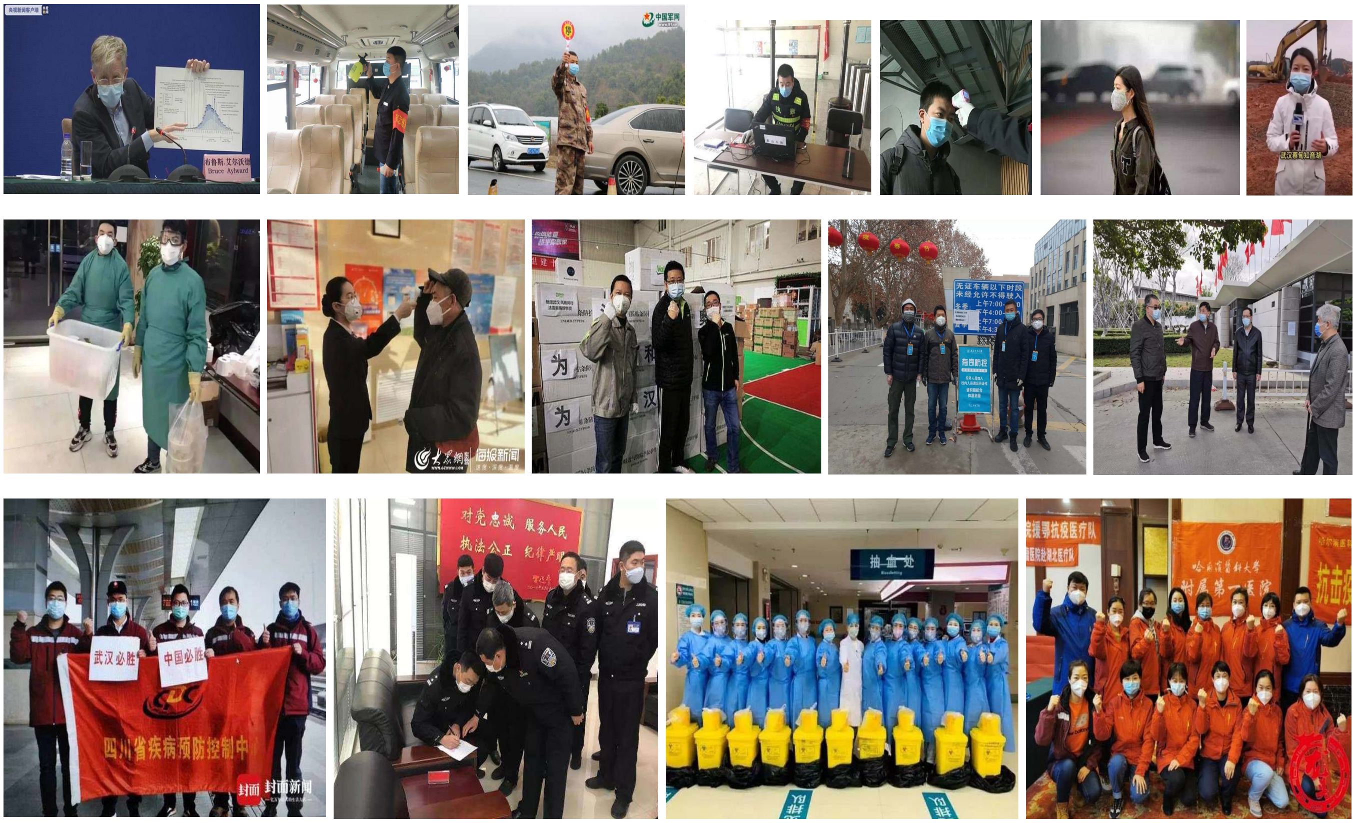

Fig. 5. Some images of our WMD data set. It covers various scenes and crowd density. For the first row, each image has only one wearing mask. For the second row, the number of wearing masks for each image is from two to four. For the third row, each image has five and more wearing masks, and smaller wearing masks are included.

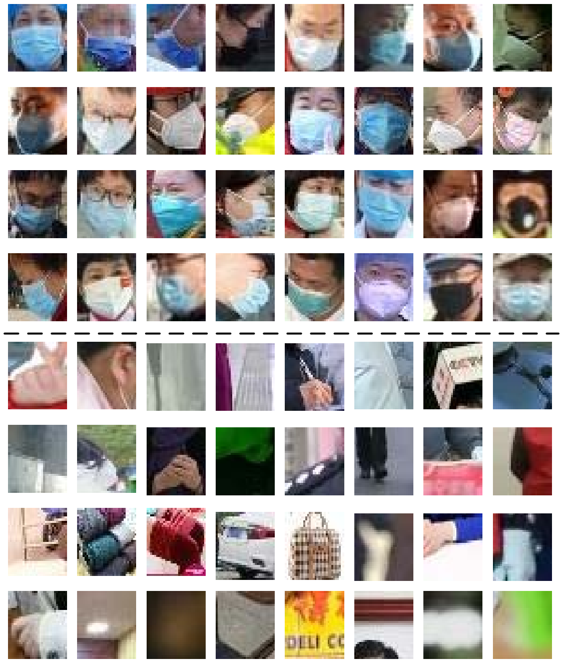

Fig. 6. Some samples in our WMC data set. Top: wearing mask images. Bottom: background images.

B. Evaluation Metrics and Parameter Setting

To measure the performance of different methods, evaluation metrics need to be invested. Intersection over Union (IoU) is always used to compare the predicted boxes with ground-truth boxes [56]

$$\text{IoU} = \frac{|P \bigcap G|}{|P \bigcup G|} \tag{8}$$

where P is defined as a predicted box and G is defined as ground-truth box. $\bigcap$ is the intersection operation, and $\bigcup$ is the union operation. The range of IoU is $0 \leq IoU \leq 1$, which stands for matching confidence. In this article, if it meets $0.45 \leq IoU \leqslant 1$, the predicted box will be seen as a success.

The metric IoU is used for one box comparison. For a data set, there are many boxes in images. Thus, common metrics include Recall, Precision, $F1$, and False Rate that are used for the statistical analysis. The term TP represents the number of positive samples that are classified as wearing masks. The term FN represents the number of positive samples that are classified as background. The term FP represents the number of negative samples that are classified as wearing masks

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{9}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{10}$$

$$F1 = 2 * \frac{\text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \tag{11}$$

$$\text{False Rate} = \frac{\text{FP}}{\text{TP} + \text{FP}}. \tag{12}$$

TABLE III

RESULTS OF GRID SEARCH FOR TRAINING BLS MODEL

N1	*N2*	*N3*	*Train Acc (%)*	*Train Time (s)*	*Test Acc (%)*	*Test Time (s)*
10	10	3800	98.101	13.588	94.964	0.642
10	15	8400	99.271	39.936	95.145	0.926
10	20	4200	98.343	15.816	95.196	0.733
10	25	5600	98.710	23.928	95.222	0.855
15	10	8400	99.291	40.927	94.628	0.971
15	15	7200	99.017	32.951	94.886	0.956
15	20	5200	98.617	22.618	94.835	0.883
15	25	7800	99.145	40.659	95.041	1.136
20	10	6200	98.883	26.861	94.990	0.893
20	15	6000	98.780	28.488	95.041	0.991
20	20	7400	99.110	38.783	95.351	1.191
20	25	6400	98.955	34.059	95.119	1.193
25	10	5000	98.523	22.464	94.628	0.936
25	15	8000	99.189	43.042	95.145	1.324
25	20	7200	99.055	39.079	95.145	1.280
25	25	7600	99.168	45.295	95.041	1.441

Experiments are conducted on a PC with Windows 10 Operating System, Intel Core i7-10700F CPU, Tensorflow 1.5, and NVIDIA Geforce GTX 1660 Super with 6-GB memory. The structures of compared methods keep up with their original settings. The PaddlePaddle method [55] provides a trained model and an API for users to detect wearing masks. As a software, its mask detection function [55] is built upon the algorithm [57]. SSD-MobileNet-V1 [37] and SSD-InceptionV2 [24] are performed under a CPU mode. All the deep learning frameworks are trained on our data set and fine-tuned from [50] except the method [55].

For our method, the parameter settings in the predetection stage are as follows: the maximum of proposals for RPN is 300, the learning rate is 0.0002, the momentum is 0.9, and the training process runs 200k steps. For the verification stage, the parameter settings are $\tau_l = 0.1$ and $\tau_h = 0.8$; parameters of the BLS model are selected by grid search; and some results are given in Table III. We define $N1$ as the number of groups of mapped features, $N2$ as the number of mapped nodes for each group, and $N3$ as the number of enhanced feature nodes. Finally, the parameter setting with the highest test accuracy (95.351%) is employed: the total number of mapped feature nodes is 400, and the number of enhanced nodes is 7400. The outline to BLS is illustrated in Table III, which is generated on a PC with Windows 10, MATLAB R2017a, and Intel Xeon CPU E5-1650 V2.

TABLE IV

QUANTITATIVE COMPARISON (%) OF THE METHODS ON DS1 SET

DS1 TestSet	*Recall* ↑	*Precision* ↑	*F1* ↑	*FalseRate* ↓
PaddlePaddle	91.20	99.56	95.20	0.44
SSD-MobileNet-V1	75.60	97.42	85.14	2.58
SSD-InceptionV2	91.0	97.64	94.20	2.36
Faster_RCNN-ResNet50	94.80	96.54	95.66	3.46
Faster_RCNN-InceptionV2	98.40	94.62	96.47	5.38
Ours	98.20	96.46	97.32	3.54

TABLE V

QUANTITATIVE COMPARISON (%) OF THE METHODS ON DS2 SET

DS2 TestSet	*Recall* ↑	*Precision* ↑	*F1* ↑	*FalseRate* ↓
PaddlePaddle	79.63	99.57	88.49	0.43
SSD-MobileNet-V1	45.95	89.45	60.72	10.55
SSD-InceptionV2	73.53	90.39	81.09	9.61
Faster_RCNN-ResNet50	88.20	96.11	91.99	3.89
Faster_RCNN-InceptionV2	94.10	92.58	93.33	7.42
Ours	94.17	93.85	94.01	6.15

TABLE VI

QUANTITATIVE COMPARISON (%) OF THE METHODS ON DS3 SET

DS3 TestSet	*Recall* ↑	*Precision* ↑	*F1* ↑	*FalseRate* ↓
PaddlePaddle	62.53	98.35	76.45	1.65
SSD-MobileNet-V1	30.07	87.06	44.70	12.94
SSD-InceptionV2	56.17	86.35	68.06	13.65
Faster_RCNN-ResNet50	82.00	96.11	88.50	3.89
Faster_RCNN-InceptionV2	87.91	93.03	90.40	6.97
Ours	88.24	94.22	91.13	5.78

TABLE VII

OVERALL QUANTITATIVE COMPARISON (%) ON THE WHOLE TEST SET

Overall Result	*Recall* ↑	*Precision* ↑	*F1* ↑	*FalseRate* ↓
PaddlePaddle	77.79	99.16	87.18	0.84
SSD-MobileNet-V1	51.92	94.09	64.27	5.91
SSD-InceptionV2	73.56	91.46	81.12	8.54
Faster_RCNN-ResNet50	88.33	96.25	92.12	3.75
Faster_RCNN-InceptionV2	93.47	93.41	93.43	6.59
Ours	93.54	94.84	94.19	5.16

C. Quantitative Analysis

In this part, experiments are conducted on test sets: DS1, DS2, and DS3. Tables IV–VI elaborate on the detailed quantitative results. It can be seen from the tables that the tendency of Recall and $F1$ in the three tables both decrease for all methods. It clearly indicates that the difficulty level is from easy to hard for DS1, DS2, and DS3.

Table IV shows that PaddlePaddle achieves a very high Precision with 99.57%, but its Recall is unsatisfied. One main reason may be derived from the slightness of its model, which is based on Pyramidbox [57]. As far as SSD-MobileNet-V1

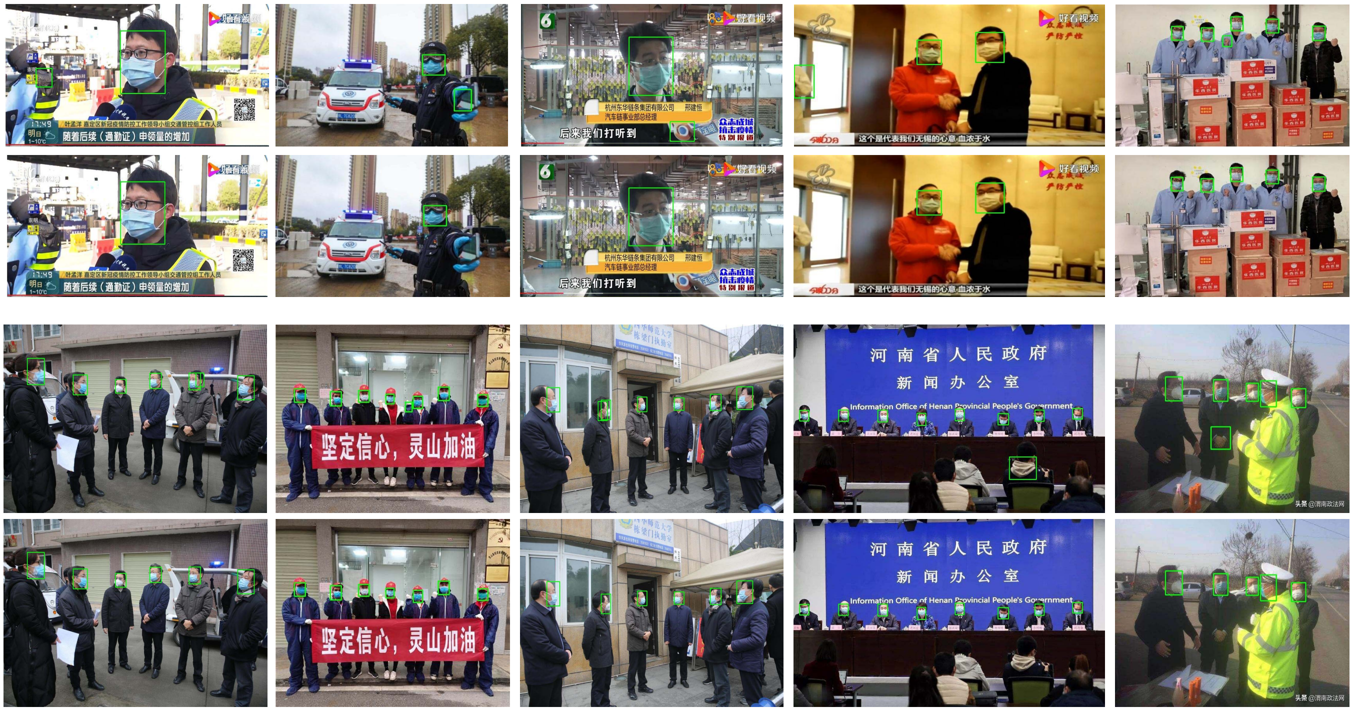

Fig. 7. Visual comparison between Faster_RCNN-InceptionV2 and ours. The first and third rows are generated by Faster_RCNN-InceptionV2, and the second and the fourth rows are generated by our approach.

is concerned, it is inferior to others in the metric of Recall. It is designed for mobile applications; thus, it has a fast running speed. However, it is at the cost of accuracy. The $F1$ value of SSD-InceptionV2 is 9% more than SSD-MobileNet-V1. Faster_RCNN-ResNet50 and Faster_RCNN-InceptionV2 are both built upon the same framework; the difference is the structure of convolutional layers. It can be seen from Table IV that the InceptionV2 structure has advantages over Faster_RCNN-ResNet50 on the metrics of Recall and $F1$. Although the Recall of our method is a bit lower than that of Faster_RCNN-InceptionV2, the proposed method is better than Faster_RCNN-InceptionV2 in the metrics Precision, $F1$, and False Rate.

For the DS2 results in Table V, we experimentally show that the proposed approach is superior to the compared methods in the Recall and $F1$. The detection task for DS2 is harder than DS1 because most of the samples in DS2 have more variations and sizes. This can be concluded from the comparison between Tables IV and V by the metrics of Reacll, Precision, and $F1$ for any method. Especially, the Rrecall of SSD-MobileNet-V1 decreases very largely from 75.60% to 45.95% because shallow layers lead to its weak ability to extract essential features. SSD-InceptionV2 also suffers the obvious decrease of Recall from 91.0% to 73.53%. The methods based on the Faster_RCNN framework tend to obtain more stable and better results than [37], [55]. Our method is no exception.

DS3 is a more challenging set than the previous two sets because more extreme small objects are contained. The changing of Recall sheds light on this point. It can be clearly noted that SSD-MobileNet-V1 and PaddlePaddle are at a low rhythm with an obvious decrease of Recall. SSD-MobileNet-V1 fails to detect wearing masks, with only 30.07%. The $F1$ values of the methods [37], [55] are all below 80%. The results of SSD-InceptionV2 are only better than those of SSD-MobileNet-V1. It has difficulty in detecting small wearing masks. The methods based on the Faster_RCNN framework outperform others obviously. It should be pointed out that our method achieves the highest Recall value with a competitive Precision result in Table VI. Meanwhile, Table VII also demonstrates our approach's effectiveness and advantages over the compared methods. In summary, three test sets represent different scenes from the perspective of size, crowd, and variations in realistic applications. Our approach achieves impressive results.

Moreover, we also offer a comparison of running time. Experiments are performed on size of 640 × 480 pixels' image. The running time for methods is listed: PaddlePaddle (473.5 ms), SSD-MobileNet-V1 (72.8 ms), SSD-InceptionV2 (201.6 ms), Faster_RCNN-ResNet50 (217.7 ms), and Faster_RCNN-InceptionV2 (105.8 ms). Among them, our approach consists of two parts: predetection (Faster_RCNN framework) and verification (BLS model). The verification stage mainly depends on the number of candidate regions within the score range ($\leqslant \tau_h$). It takes about 6.7 ms for our BLS model to process an image with 32 × 32 pixels. If all the scores are higher than τ_h, the BLS model would not be carried out, and the computations are saved.

D. Visual Results and Discussion

Fig. 7 present a visual comparison between Faster_RCNN-InceptionV2 and ours. For the candidate regions with low scores, our method is able to remove background regions and ensure the Precision. For the fourth column, the white protective suit and pale hoodie hat are classified as wearing masks

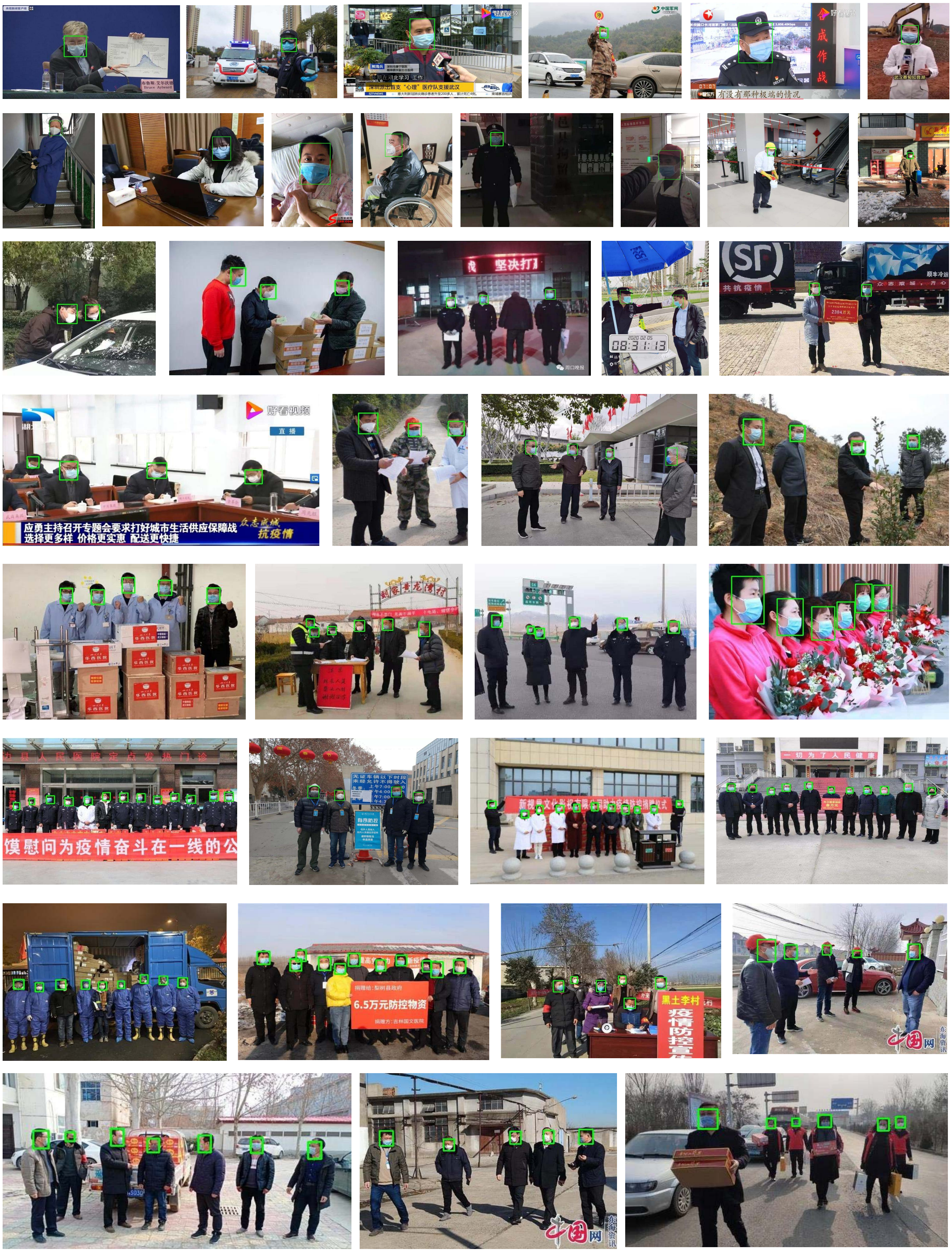

Fig. 8. More visual results of WMD. The first and second rows are the results of DS1. The third and fourth rows are the results of DS2. The remaining rows are the results of DS3.

by mistake because they look like a mask in color and shape. For the fifth columns, some hands are classified as wearing masks with medium scores; these mistakes are inevitable for the Faster_RCNN framework. By the verification of the BLS

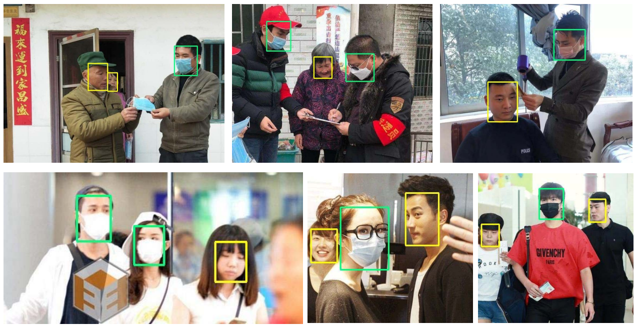

Fig. 9. Apply our approach into face/mask detection. The green boxes represent the wearing masks, and the yellow boxes represent the faces.

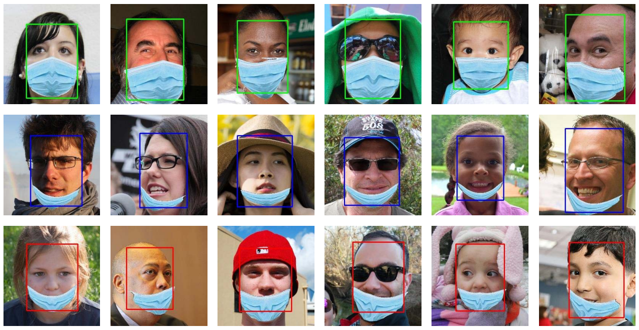

Fig. 10. Extend our approach to the classification of wearing mask. The first row (green boxes) represents the correct class Mask: mask covering nose, mouth, and chin. The second row (blue boxes) represents the incorrect class Mask_Chin: mask only covering chin. The third row (red boxes) represents the incorrect class Mask_Mouth_Chin: mask only covering mouth and chin.

classifier, these mistakes can be corrected effectively. More visual results are given in Fig. 8.

We also conduct an experiment of detecting one who wears a mask or not. If there are faces and wearing masks in images, we detect both of them. To reach the goal, we create a face data set that encompasses more than 16k faces. Then, we combine it with the wearing mask data set together to train our model. The parameters of our model remain unchanged except for adding a face category. Some detection results are shown in Fig. 9. What is more, our approach is expected to combine with infrared thermal imaging temperature measurement technique, protecting the public service professionals and nucleic acid test from the COVID-19 infection risks caused by close contacts. Therefore, our approach is expected to be promising.

In addition, we extend our work with the classification of correct wearing mask and incorrect wearing mask. A method designed by [23] is utilized for comparison. Its implementation depends on OpenCV library classifiers. If a face region is detected, nose detection and mouth detection will be applied to predict whether there is a mask or not and whether wearing a mask is correct or not. The data set used for experiments is proposed by [48]. A total of 13 200 images are selected randomly, including three categories: Mask(correct), Mask_Chin(incorrect), and Mask_Mouth_Chin(incorrect). The used data set covers the train set (7500 images), val set (1200 images), and test set (4500 images). The accuracy results obtained by [23] are correct Mask (66.87%), incorrect Mask_Chin (84.4%), and incorrect Mask_Mouth_Chin (67.73%). The accuracy results obtained by our method are correct Mask (99.87%), incorrect Mask_Chin (99.93%), and incorrect Mask_Mouth_Chin (97.47%). It is clearly noted that our method achieves competitive results, outperforming the method [23] significantly. Some visual results generated by our method are presented in Fig. 10.

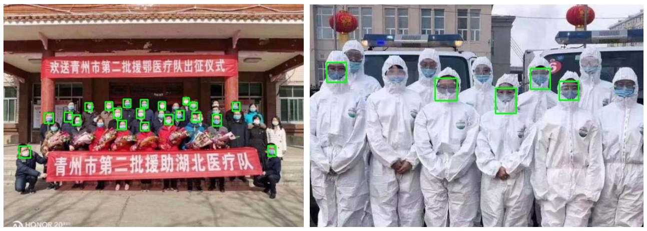
欢送青州市第二批援鄂医疗队出征仪式
青州市第二批援助湖北医疗队

Fig. 11. Some detection failures, including the small objects and facial regions occluded by whole protective clothing.

However, there are still some failures in results, as shown in Fig. 11. It is difficult for our method to deal with small objects and the facial almost protected by protective clothing, mask, and medical goggles. Insufficient features might be the main reason. A possible solution to this problem is to apply image super-resolution with the current approach. In this regard, more research needs to be investigated.

V. Conclusion

In this article, we propose a hybrid deep transfer learning and BLS for facial mask detection. It is designed to contain two stages: predetection and verification. The predetection is implemented by the Faster_RCNN framework through a transfer learning technique. The detection model is fine-tuned from a multiple-class detection model. The verification is implemented by a classifier of BLS. With a low score setting in predetection, more candidate regions are used for verification. This strategy is able to reach a tradeoff between Recall and Precision. Notably, we build a wearing mask data set containing 17 654 train masks, 1936 val masks, and 6813 test masks. The test set encompasses three sets varying from easy to hard. Experimental results shed light on our approach's effectiveness with a Recall of 93.54% and a Precision of 94.84% and advantages over the compared methods. The proposed method is expected to detect wearing masks to help realize the functions, such as noncontact temperature measurement and monitoring crowd in the pandemic era and other situations. Hopefully, our work can provide some help in the fighting against COVID-19.

References

[1] *COVID-19 Dashboard By the Center for Systems Science and Engineering*. Accessed: Mar. 4, 2021. [Online]. Available: https://coronavirus.jhu.edu/map.html

[2] E. P. Fischer, M. C. Fischer, D. Grass, I. Henrion, W. S. Warren, and E. Westman, "Low-cost measurement of face mask efficacy for filtering expelled droplets during speech," *Sci. Adv.*, vol. 6, no. 36, Sep. 2020, Art. no. eabd3083.

[3] W. H. Organization *et al.*, "Advice on the use of masks in the context of COVID-19: Interim guidance, 5 June 2020," World Health Org., Geneva, Switzerland, Tech. Rep. WHO/2019-nCoV/IPC_Masks/2020.4, 2020.

[4] M. Klompas, C. A. Morris, J. Sinclair, M. Pearson, and E. S. Shenoy, "Universal masking in hospitals in the COVID-19 era," *New England J. Med.*, vol. 382, no. 21, p. e63, May 2020.

[5] Z. Jiang *et al.*, "Detection of respiratory infections using rgb-infrared sensors on portable device," *IEEE Sensors J.*, vol. 20, no. 22, pp. 13674–13681, Oct. 2020.

[6] S. Khan *et al.*, "Comparative accuracy testing of non-contact infrared thermometers and temporal artery thermometers in an adult hospital setting," *Amer. J. Infection Control*, vol. 49, no. 5, pp. 597–602, Oct. 2020.

[7] J. Cheng *et al.*, "Remote heart rate measurement from near-infrared videos based on joint blind source separation with delay-coordinate transformation," *IEEE Trans. Instrum. Meas.*, vol. 70, 2021, Art. no. 5005313.

[8] M. Khanafer and S. Shirmohammadi, "Applied AI in instrumentation and measurement: The deep learning revolution," *IEEE Instrum. Meas. Mag.*, vol. 23, no. 6, pp. 10–17, Sep. 2020.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[10] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[11] C. L. P. Chen and Z. Liu, "Broad learning system: An effective and efficient incremental learning system without the need for deep architecture," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 10–24, Jan. 2018.

[12] C. L. P. Chen, Z. L. Liu, and S. Feng, "Universal approximation capability of broad learning system and its structural variations," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 4, pp. 1191–1204, Apr. 2018.

[13] *Libfacedetection*. Accessed: Feb. 28, 2020. [Online]. Available: https://github.com/ShiqiYu/libfacedetection

[14] J. Wang, Y. Yuan, and G. Yu, "Face attention network: An effective face detector for the occluded faces," 2017, *arXiv:1711.07246*. [Online]. Available: http://arxiv.org/abs/1711.07246

[15] M. Omidyeganeh *et al.*, "Yawning detection using embedded smart cameras," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 3, pp. 570–582, Mar. 2016.

[16] Y. Chen, L. Song, Y. Hu, and R. He, "Adversarial occlusion-aware face detection," in *Proc. IEEE 9th Int. Conf. Biometrics Theory, Appl. Syst. (BTAS)*, Oct. 2018, pp. 1–9.

[17] J. Li *et al.*, "DSFD: Dual shot face detector," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5060–5069.

[18] S. Ge, J. Li, Q. Ye, and Z. Luo, "Detecting masked faces in the wild with LLE-CNNs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2682–2690.

[19] E. Learned-Miller, G. B. Huang, A. RoyChowdhury, H. Li, and G. Hua, "Labeled faces in the wild: A survey," in *Advances in Face Detection and Facial Image Analysis*. Berlin, Germany: Springer, 2016, pp. 189–248.

[20] M. Kopaczka, R. Kolk, J. Schock, F. Burkhard, and D. Merhof, "A thermal infrared face database with facial landmarks and emotion labels," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 5, pp. 1389–1401, May 2019.

[21] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, Oct. 2001, pp. I.

[22] B. S. Bayu Dewantara and D. Twinda Rhamadhaningrum, "Detecting multi-pose masked face using adaptive boosting and cascade classifier," in *Proc. Int. Electron. Symp. (IES)*, Sep. 2020, pp. 436–441.

[23] N. Petrovic and D. Kocic, "Iot-based system for COVID-19 indoor safety monitoring," in *Proc. IcETRAN*, 2020, pp. 1–6.

[24] U. Alganci, M. Soydas, and E. Sertel, "Comparative research on deep learning approaches for airplane detection from very high-resolution satellite images," *Remote Sens.*, vol. 12, no. 3, p. 458, Feb. 2020.

[25] M. Jiang, X. Fan, and H. Yan, "RetinaMask: A face mask detector," 2020, *arXiv:2005.03950*. [Online]. Available: http://arxiv.org/abs/2005.03950

[26] G. J. Chowdary, N. S. Punn, S. K. Sonbhadra, and S. Agarwal, "Face mask detection using transfer learning of inceptionv3," in *Proc. Int. Conf. Big Data Anal.* Berlin, Germany: Springer, 2020, pp. 81–90.

[27] M. M. Rahman, M. M. H. Manik, M. M. Islam, S. Mahmud, and J.-H. Kim, "An automated system to limit COVID-19 using facial mask detection in smart city network," in *Proc. IEEE Int. IoT, Electron. Mechatronics Conf.*, Sep. 2020, pp. 1–5.

[28] M. Loey, G. Manogaran, M. H. N. Taha, and N. E. M. Khalifa, "A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic," *Measurement*, vol. 167, Jan. 2020, Art. no. 108288.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[30] B. Qin and D. Li, "Identifying facemask-wearing condition using image super-resolution with classification network to prevent COVID-19," *Sensors*, vol. 20, no. 18, p. 5236, Sep. 2020.

[31] S. V. Militante and N. V. Dionisio, "Real-time facemask recognition with alarm system using deep learning," in *Proc. 11th IEEE Control Syst. Graduate Res. Colloq. (ICSGRC)*, Oct. 2020, pp. 106–110.

[32] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[33] M. Loey, G. Manogaran, M. H. N. Taha, and N. E. M. Khalifa, "Fighting against COVID-19: A novel deep learning model based on YOLO-v2 with ResNet-50 for medical face mask detection," *Sust. Cities Soc.*, vol. 65, Feb. 2020, Art. no. 102600, 2020.

[34] X. Ren and X. Liu, "Mask wearing detection based on YOLOV3," *J. Phys., Conf. Ser.*, vol. 1678, no. 1, 2020, Art. no. 012089.

[35] P. Mohan, A. Jyoti Paul, and A. Chirania, "A tiny CNN architecture for medical face mask detection for resource-constrained endpoints," 2020, *arXiv:2011.14858*. [Online]. Available: http://arxiv.org/abs/2011.14858

[36] A. S. Joshi, S. S. Joshi, G. Kanahasabai, R. Kapil, and S. Gupta, "Deep learning framework to detect face masks from video footage," in *Proc. 12th Int. Conf. Comput. Intell. Commun. Netw. (CICN)*, Sep. 2020, pp. 435–440.

[37] A. G. Howard *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*. [Online]. Available: http://arxiv.org/abs/1704.04861

[38] S. R. Rudraraju, N. K. Suryadevara, and A. Negi, "Face mask detection at the fog computing gateway," in *Proc. Federated Conf. Comput. Sci. Inf. Syst.*, Sep. 2020, pp. 521–524.

[39] W. Hariri, "Efficient masked face recognition method during the COVID-19 pandemic," 2020. [Online]. Available: https://www.researchsquare.com/article/rs-39289/v3, doi: 10.21203/rs.3.rs-39289/v3.

[40] S. Chen, W. Liu, and G. Zhang, "Efficient transfer learning combined skip-connected structure for masked face poses classification," *IEEE Access*, vol. 8, pp. 209688–209698, 2020.

[41] L. Li, X. Mu, S. Li, and H. Peng, "A review of face recognition technology," *IEEE Access*, vol. 8, pp. 139110–139120, 2020.

[42] N. Damer, J. H. Grebe, C. Chen, F. Boutros, F. Kirchbuchner, and A. Kuijper, "The effect of wearing a mask on face recognition performance: An exploratory study," in *Proc. Int. Conf. Biometrics Special Interest Group*, Oct. 2020, pp. 1–6.

[43] B. Yang, J. Wu, and G. Hattori, "Facial expression recognition with the advent of face masks," in *Proc. 19th Int. Conf. Mobile Ubiquitous Multimedia*, Nov. 2020, pp. 335–337.

[44] Z. Wang *et al.*, "Masked face recognition dataset and application," 2020, *arXiv:2003.09093*. [Online]. Available: http://arxiv.org/abs/2003.09093

[45] *Prajnasb's Masked Face Dataset*. Accessed: Apr. 15, 2020. [Online]. Available: https://github.com/prajnasb/observations

[46] *Face Mask Detection Implemented by AIZOOTech*. Accessed: Apr. 3, 2020. [Online]. Available: https://github.com/AIZOOTech/FaceMaskDetection

[47] S. Yang, P. Luo, C. C. Loy, and X. Tang, "WIDER FACE: A face detection benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5525–5533.

[48] A. Cabani, K. Hammoudi, H. Benhabiles, and M. Melkemi, "MaskedFace-net—A dataset of correctly/incorrectly masked face images in the context of COVID-19," *Smart Health*, vol. 19, Mar. 2021, Art. no. 100144.

[49] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[50] *TensorFlow 1 Detection Model Zoo*. Accessed: Dec. 25, 2020. [Online]. Available: https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf1_detection_zoo.md

[51] T.-Y. Lin *et al.*, "Microsoft coco: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2014, pp. 740–755.

[52] *LabelImg*. Accessed: Dec. 25, 2020. [Online]. Available: https://github.com/tzutalin/labelImg/

[53] C. L. P. Chen and J. Z. Wan, "A rapid learning and dynamic stepwise updating algorithm for flat neural networks and the application to time-series prediction," *IEEE Trans. Syst., Man, Cybern. B. Cybern.*, vol. 29, no. 1, pp. 62–72, Oct. 1999.

[54] J. Tang, C. Deng, and G.-B. Huang, "Extreme learning machine for multilayer perceptron," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 4, pp. 809–821, Apr. 2016.

[55] *Pyramidbox_lite_mobile_mask Implemented By Paddlepaddle for Mask Detection*. Accessed: Feb. 24, 2020. [Online]. Available: https://www.paddlepaddle.org.cn/hub/scene/maskdetect

[56] C. L. P. Chen and B. Wang, "Random-positioned license plate recognition using hybrid broad learning system and convolutional networks," *IEEE Trans. Intell. Transp. Syst.*, early access, Aug. 4, 2020, doi: 10.1109/TITS.2020.3011937.

[57] X. Tang, D. K. Du, Z. He, and J. Liu, "Pyramidbox: A context-assisted single shot face detector," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 797–813.

Bingshu Wang received the B.S. degree in computer science and technology from Guizhou University, Guiyang, China, in 2013, the M.S. degree in electronic science and technology (integrated circuit system) from Peking University, Beijing, China, in 2016, and the Ph.D. degree in computer science from the University of Macau, Macau, in 2020.

He is currently an Associate Professor with the School of Software, Northwestern Polytechnical University, Suzhou, China. His current research interests include computer vision, intelligent video analysis, and machine learning.

Dr. Wang is also a member of the Chinese Association of Automation (CAA).

Yong Zhao (Member, IEEE) received the Ph.D. degree in automatic control and applications from Southeast University, Nanjing, China, 1991.

He then joined Zhejiang University, Hangzhou, China, as an Assistant Researcher. In 1997, he went to Concordia University, Montreal, QC, Canada, as a Post-Doctoral Fellow. He was a Senior Audio/Video Compression Engineer with Honeywell Corporation, Mississauga, ON, Canada, in May 2000. In 2004, he became an Associate Professor at the Peking University Shenzhen Graduate School, Shenzhen, China, where he is currently the Header of the lab Mobile Video Networking Technologies. He is working on computer vision, machine learning, video analytics, and video compression with a special focus on applications of these new theories and technologies to various industries. His team has developed many innovative products and projects that have been successful in the market.

C. L. Philip Chen (Fellow, IEEE) graduated from the University of Michigan, Ann Arbor, MI, USA, in 1985.

He is currently a Chair Professor and the Dean of the College of Computer Science and Engineering, South China University of Technology, Guangzhou, China. Being a Program Evaluator of the Accreditation Board of Engineering and Technology Education (ABET) in the US, for computer engineering, electrical engineering, and software engineering programs, he successfully architects the Engineering and Computer Science Programs of the University of Macau, Macau, receiving accreditations from Washington/Seoul Accord through The Hong Kong Institution of Engineers (HKIE), Hong Kong, which is considered as his utmost contribution in engineering/computer science education for Macau as the former Dean of the Faculty of Science and Technology. His current research interests include cybernetics, systems, and computational intelligence.

Dr. Chen is also a fellow of the American Association for the Advancement of Science (AAAS), the International Association of Pattern Recognition (IAPR), the Chinese Association of Automation (CAA), and HKIE and a member of the Academia Europaea (AE), the European Academy of Sciences and Arts (EASA), and the International Academy of Systems and Cybernetics Science (IASCYS). He received the IEEE Norbert Wiener Award in 2018 for his contribution to systems and cybernetics, and machine learning. He received the Best Transactions Paper Awards from the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS for his papers in 2014 and 2018. He was a recipient of the 2016 Outstanding Electrical and Computer Engineers Award from his alma mater, Purdue University, in 1988, after he graduated from the University of Michigan. He was the Chair of TC 9.1 Economic and Business Systems of the International Federation of Automatic Control from 2015 to 2017. He is also a Highly Cited Researcher by Clarivate Analytics from 2018 to 2020. He was the IEEE Systems, Man, and Cybernetics Society President from 2012 to 2013 and the Editor-in-Chief of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS from 2014 to 2019. He is also the Editor-in-Chief of the IEEE TRANSACTIONS ON CYBERNETICS and an Associate Editor of the IEEE TRANSACTIONS ON ARTIFICIAL INTELLIGENCE and IEEE TRANSACTIONS ON FUZZY SYSTEMS.