1 Introduction

Some samples of wearing masks in Fig. 1 indicate that wearing a mask is an effective means to fight against COVID-19.

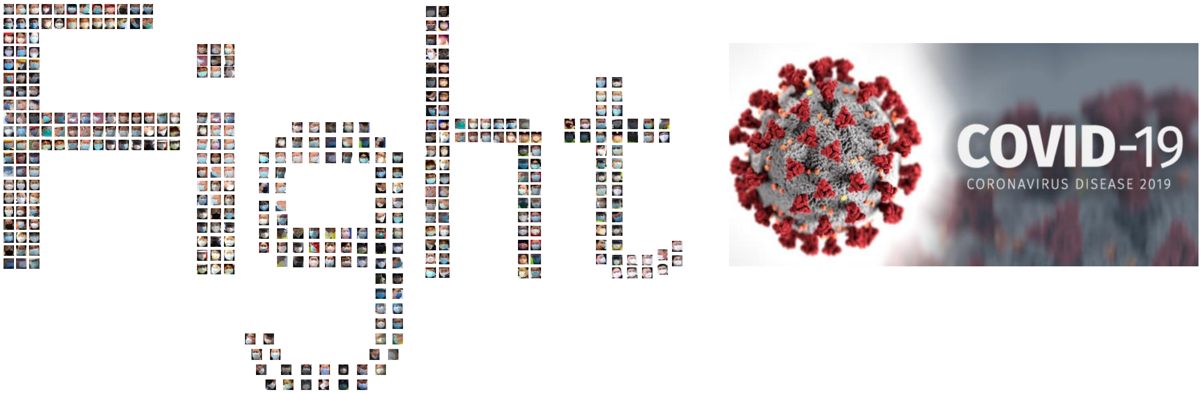
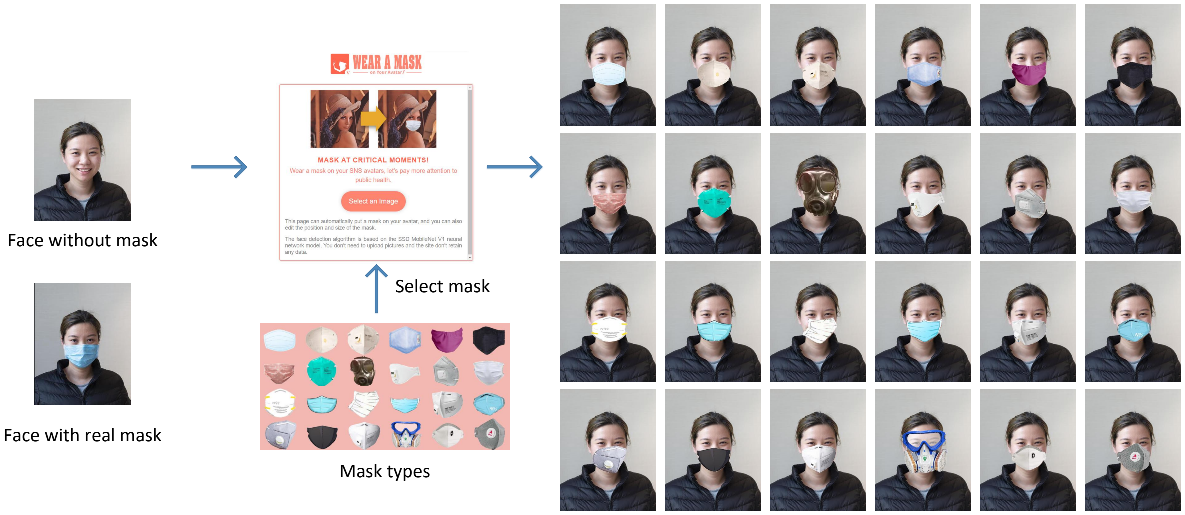


Figure 1: Some small images of people wearing masks to fight against COVID-19. The virus image is from "https://fscluster.org/coronavirus".

2 Masked facial Detection Datasets

An example of stimulating mask-wearing is presented in Fig. 2.



Wear a mask on your SNS avatars, let's pay more attention to public health.

This page can automatically put a mask on your avatar, and you can also edit the position and size of the mask.

The face detection algorithm is based on the SSD MobileNet V1 neural network model. You don't need to upload pictures and the site don't retain any data.

Figure 2: An example of simulating masks. Face with real mask is also provided in comparison with simulated samples. The input image is captured from a website [1]. The simulated method [2] provides 24 types of masks and its demo link is available [3].

3 Masked Facial Detection Methods

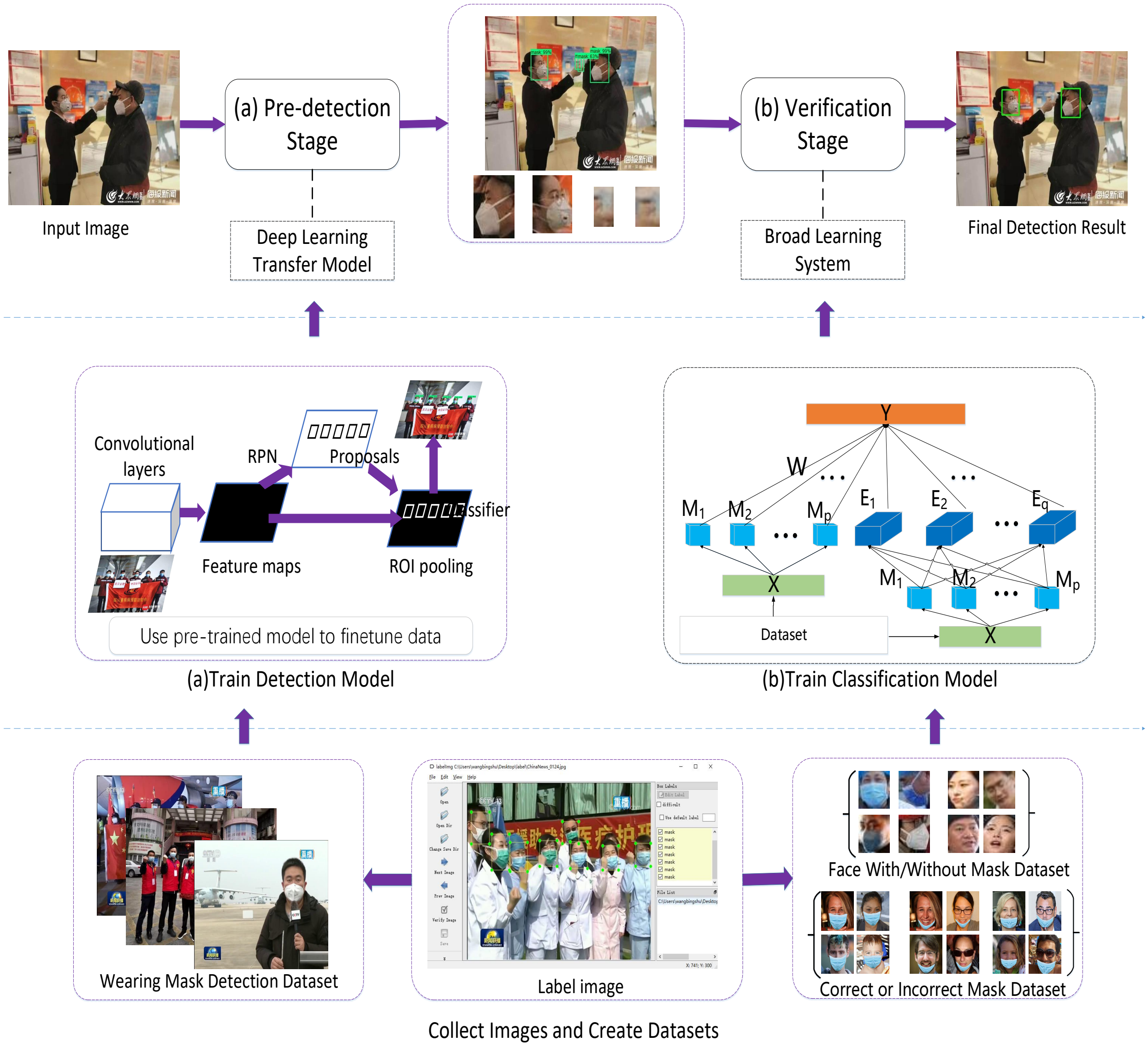

Input Image
(a) Pre-detection Stage
Deep Learning Transfer Model
(b) Verification Stage
Broad Learning System
Final Detection Result
Convolutional layers
RPN
Proposals
ssifier
Feature maps
ROI pooling
Use pre-trained model to finetune data
(a)Train Detection Model
Y
W
M1
M2
Mp
E1
E2
Eq
X
Dataset
(b)Train Classification Model
Wearing Mask Detection Dataset
Label image
Face With/Without Mask Dataset
Correct or Incorrect Mask Dataset
Collect Images and Create Datasets

Figure 3: An example of two-stage method using deep learning transfer model for face pre-detection and broad learning system for verification.

Neural Network + Neural Network:

Fig. 3 outlines a representative [4] using two-stage strategy to realize masked face detection task. The first stage is designed by a deep learning transfer model: Faster R-CNN [5, 6] and the second stage is designed by broad learning system (BLS) [7]. Input image is sent to the pre-detection stage. Then many candidate regions are generated and they are further classified by trained BLS model which can remove false positives and keep masked faces. Finally, detected results are generated with labels. To train pre-detection model, annotated dataset is required, which is created using a tool called "LabelImg" [8]. The extracted faces and masks can be used to create classification datasets that are problem-dependent, for example, with/without mask, correct/incorrect mask.

The pre-detection in Faster R-CNN structure mainly includes four steps: extract feature maps, generate proposals by Region Proposal Networks (RPN), obtain fixed dimension of feature map, and object classification and location regression. Faster R-CNN has advantages over SSD and YOLO in accuracy [9]. Its loss function is denoted by

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

where i represents the index of an anchor. p_i and p_i^* are defined as the predicted probability of a box and real value of ground-truth anchor, respectively. For p_i^*, its value is 1 for positive anchor and 0 for negative anchor. t_i and t_i^* are the predicted coordinates of a box and ground-truth, respectively. Classification loss is expressed by L_{cls} and regression loss is L_{reg}. The $p_i^* L_{reg}$ means that only positive anchors are considered. Terms N_{cls} and N_{reg} are used to normalize classification loss and regression loss. λ is defined as a weighted balance.

The verification stage employs BLS in Fig.3 . BLS is a flat neural network structure with a very high training efficiency [7] and many variants have been proposed [10, 11, 12]. Herein, we give the basic description about the basic BLS. Its main idea is to convert input images into random feature nodes as "mapped features", and expand all the mapped features to enhanced nodes as "enhanced features". All the features including mapped and enhance nodes are connected to output. The weight can be computed by the pseudo inverse of ridge regression approximation. Details are presented as follows.

The "mapped features" are expressed by

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

where W_{m_i} and β_{m_i} are generated weights randomly from given distribution, φ is a mapping function. Then, Sparse auto-encoder is used to explore more essential features from all the mapped features. After p groups of mapping operations, the mapped features can be expressed by a concatenation of $M^p \equiv [M_1, ..., M_p]$, which is used to expand enhanced features.

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

where σ is a nonlinear activation function like *tansig*. The terms W_{e_j} and β_{e_j} are generated weights from given distribution. After q groups of expanding operations, enhanced features are expressed by $E^q \equiv [E_1, ..., E_q]$.

The $M^p$and E^q are both connected to the output.

$$\begin{aligned} Y &= [M_1, M_2, ..., M_p, E_1, E_2, ..., E_q]W \\ &= [M^p | E^q]W \end{aligned} \tag{4}$$

where Y is the output. The weights of whole network W can be computed from $W \triangleq [M^p|E^q]^+ Y$. $[M^p|E^q]^+$ can be computed by the pseudo inverse of ridge regression approximation. It should be noted that the parameter setting of p and q depends on the task complexity.

In practice, when a BLS model can not learn a task well, one effective way is to add feature nodes that is called incremental learning. This ensures efficiency in training phase. It does not need to retrain from the scratch [4]. The combination of Faster R-CNN and BLS are verified to be effective on WMD dataset [4]. It achieves 97.32% accuracy for simple scene and 91.13% for complex scene. BLS can be as a good selection for classification when training efficiency and small size of model are required in applications.

References

[1] http://big5.xinhuanet.com/gate/big5/www.xinhuanet.com/photo/2020-03/22/c_1125750098.htm.

[2] https://github.com/zamhown/wear-a-mask.

[3] https://zamhown.gitee.io/wear-a-mask/.

[4] B. Wang, Y. Zhao, and C. P. Chen, "Hybrid transfer learning and broad learning system for wearing mask detection in the covid-19 era," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, no. 5009612, 2021.

[5] 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, 2016.

[6] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[7] C. P. Chen and Z. Liu, "Broad learning system: An effective and efficient incremental learning system without the need for deep architecture," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 1, pp. 10–24, 2018.

[8] https://github.com/tzutalin/labelImg/.

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

[10] C. P. Chen, Z. Liu, and S. Feng, "Universal approximation capability of broad learning system and its structural variations," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 4, pp. 1191–1204, 2018.

[11] Z. Liu, C. P. Chen, S. Feng, Q. Feng, and T. Zhang, "Stacked broad learning system: From incremental flatted structure to deep model," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020.

[12] C. P. Chen and B. Wang, "Random-positioned license plate recognition using hybrid broad learning system and convolutional networks," *IEEE Transactions on Intelligent Transportation Systems*, 2020.