EAI Endorsed Transactions
on Pervasive Health and Technology
Research Article EAI.EU

An Efficient Face Mask Detector with PyTorch and Deep Learning

CMAK. Zeelan Basha[1*], B.N. Lakshmi Pravallika[2] and E. Bharani Shankar[2]

[1]Assistant Professor, Koneru Lakshmaiah Educational Foundation, Guntur, Andhra Pradesh, India.
[2]Student, Koneru Lakshmaiah Educational Foundation, Guntur, Andhra Pradesh, India.

Abstract

INTRODUCTION: The outbreak of a coronavirus disease in 2019 (COVID-19) has created a global health epidemic that has had a major effect on the way we view our environment and our daily lives. The Covid-19 affected numbers are rising at a tremendous pace. Because of that, many countries face an economic catastrophe, recession, and much more. One thing we should do is to separate ourselves from society, remain at home, and detach ourselves from the outside world. But that's no longer a choice, people need to earn to survive, and nobody can remain indefinitely within their homes. As a precaution, people should wear masks while keeping social distance, but some ignore such things and walk around.
OBJECTIVES: To develop a Face Mask Detector with OpenCV, PyTorch, and Deep Learning that helps to detect whether or not a person wears a mask.
METHODS: A Neural Network model called ResNet is trained on the dataset. Furthermore, this work makes use of the inbuilt Face Detector after training. Finally, we predict whether or not a person is wearing a mask along with the percentage of the face covered or uncovered.
RESULTS: The validation results have been proposed to be 97% accurate when compared to applying different algorithms.
CONCLUSION: This Face Mask Detection System was found to be apt for detecting whether or not people wear masks in public places which contribute to their health and also to the health of their contacts in this COVID-19 pandemic.

Keywords: COVID-19, face mask, resnet, pytorch, RMFRD.

Received on 11 September 2020, accepted on 15 December 2020, published on 08 January 2021

doi: 10.4108/eai.8-1-2021.167843

1. Introduction

COVID-19 has restructured life as we know it. Many of us remain at home, avoiding people on the streets and modifying our everyday routines, like going to school or work, in ways we have never imagined. The World Health Organization (WHO) announced coronavirus disease to be a pandemic [1] in 2019 (COVID-19). Situation Study 96 estimated that coronavirus has infected more than 2.7 million people worldwide and caused more than 180,000 deaths. Besides, there are a variety of identical broad Serious respiratory disorders, such as severe acute respiratory syndrome (SARS)[2] and the Middle East respiratory syndrome (MERS)[3], which have arisen in recent years, COVID-19 has had a higher prevalence than SARS. As a result, more people are worried about their well-being, and public health is considered a top priority for governments. Although we are modifying old habits, we need to follow new behaviors. First and foremost, it's the practice of wearing a mask or a face that covers every time we're in public space. As reported cases of COVID-19 continue to grow, the CDC (Centres for Disease Control and Prevention) recommends that everyone wears a cloth mask when going out in public. Experts claim that

[*]Corresponding author. Email: Cmak.zeelan@gmail.com

masks for community use do not stop anyone from getting infected, but they do help prevent the spread of the disease by those with the virus. Face mask detection has, therefore, become a crucial computer vision challenge to benefit global society, but studies related to face mask detection are minimal.

With the explosive development of machine learning techniques, the issue of facial recognition seems to be well solved. Beyond the impressive success of the current works, there is a concern that the implementation of better face detectors is becoming increasingly difficult. In particular, the identification of masked faces, which can be very helpful for applications such as video surveillance and event analysis, remains a major challenge for many current models. In this paper, we propose a face mask detector capable of detecting face masks and contributing to public health. The proposed method in this paper fine-tunes the Residual Neural Network popularly called ResNet and constructs the model. Later, we train the model and perform face detection to extract the Region of Interest (ROI) from each image present in the dataset. Finally, we pass the image through the constructed model to determine if the face has a mask or not. ResNet uses multiple layers and applies the Data Augmentation technique to enhance the effectiveness of the prediction. All the architectures proposed till now like AlexNet, VGGNet, GoogleNet, Inception, etc., take a lot of time and memory for training with many computations with reduced accuracy. This problem of time and memory consumption can be reduced using the methodology proposed in this paper.

This paper is organized as follows: In section 2, the related work done previously in this area is described, Section 3 explains the methodology i.e., various stages involved in the development of the proposed system. Section 4 describes the results and Section 5 concludes the paper.

2. Literature Survey

Prior, the research in the area has focused on the edge and grey value of face image being based on pattern recognition combined with the knowledge on the face model. Adaboost[4] was a fantastic training classifier. The facial detection technology has made a breakthrough with the iconic Viola Jones Detector[5], which has greatly enhanced the real-time facial detection. Viola Jones Detector refined the features of Haar[6] but failed to solve real-world issues and was affected by different factors such as facial brightness and orientation. Whereas Viola Jones could be used in well-lit frames, it did not fit well in dark environments and non-frontal images. These problems have led independent researchers to work on creating new models of deep-learning face detection to produce better results for various facial conditions. Instead of using hand-carved features, deep learning-based detectors have recently demonstrated exceptional performance due to their robustness and high extraction ability. There are two main categories: one-stage object, and two-stage object detectors. In the latter case, the two-stage detector produces a conceptual framework in the first stage, and then fine-tunes those proposals in the second phase. Moreover, the two-stage detector can provide a high detection output, however, at a low speed.

The R-CNN seminal work is proposed by R. Girshick et al.[7]. R-CNN uses selective search to suggest some of the feature vectors that may contain objects. Subsequently, proposals are transmitted into the CNN model to extract the features, and a support vector machine (SVM) is used to identify classes of objects. That being said, the second stage of R-CNN[8] is prohibitively costly, as the network must detect proposals on a yet another-by-one basis and use of a distinct SVM for the final classification task. Quintessential architectures such as AlexNet[9] and VGGNet[10] contain stacked convolutionary layers. AlexNet has won the ImageNet LSVRC-2012 competition with 5 convolutional layers and 3 fully connected layers, while VGGNet is an improvement over AlexNet as it replaces large kernels with 3x3 multiple kernels in a row. The winning GoogleNet[11] architecture of ILSVRC-2014 uses parallel convolution kernels and concatenates function maps together. It has been used for 1×1, 3×3 and 5×5 convolutions and 3×3 max-pooling. Tiny convolutions extract feature maps, whereas larger convolutions extract high-level features. We used ResNet to create skip connections that allow deep neural networks to avoid exhaustion in training accuracy. These architectures are also used for the initial extraction of features in face detection networks. Face Recognition is also done using Adaptive K-Nearest Neighbour, adaptive weighted average, reverse weighted average, and exponentially weighted average[21].

3. Methodology

Our proposed technique of detecting Face Mask starts with pre-processing followed by other methodologies as shown in the architecture of the proposed system “Fig.1”. We have used the RMFRD which stands for Real-World Masked Face Recognition Dataset available on the internet for free[22]. “Fig.2” and “Fig.3” are the images from the RMFRD representing with and without mask respectively. RMFRD is currently the largest masked face dataset within the real world. These datasets are openly accessible to academia and industry on the grounds of which different applications on masked faces can be built. The dataset contains 5,000 portraits of 525 individuals wearing masks and 90,000 pictures of the same 525 subjects with no masks. The whole project was implemented in Python using Deep Learning Libraries like PyTorch[12], Caffe[13], and Computer Vision libraries like OpenCV.

3.1. Pre-processing

Once the most suitable raw input data has been chosen, it must be pre-processed otherwise the neural network would not generate reliable predictions. The decisions taken at this stage of growth are vital to the success of the network. Transformation and Normalization [14] are two commonly used methods of pre-processing. Transformation requires the change of raw data inputs to create a new input to the network, while normalization is a transformation performed on new data input to disperse the data equally and scale it to an appropriate range for the network. Awareness of the domain is critical in choosing pre-processing methods to highlight the intrinsic features of the data, which can improve the capability of the network to learn how to connect inputs and outputs.

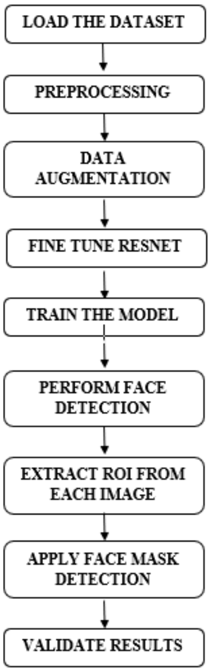

Figure 1. Proposed Model for the Face Mask Detection system

Figure 2. A Masked image from RMFRD

In our case, we have performed random resized cropping and random vertical flip to each of the images in the dataset. Most of the deep neural networks including ResNet require square images as input. Their usual pixel size is 224x224.

Figure 3. An unmasked image from the RMFRD

3.2 Data Augmentation

Data augmentation [15] is a method that can be used to arbitrarily enlarge the size of a training dataset by generating updated versions of images in a dataset. Training deep-learning neural network models on more data will result in more robust models, and enhancement techniques will generate image variations that can boost the ability of fit models to generalize what they have learned from new images. We have rendered several changes, including a variety of image manipulation operations, such as shifts, flips, zooms, mean subtractions, and far more on our dataset. “Fig. 4” illustrates an example of Data Augmentation.

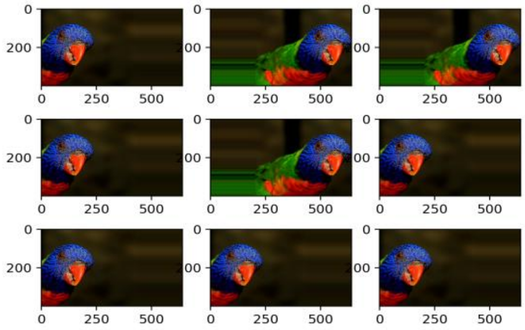

Figure 4. Data Augmentation

3.3 Fine-Tune ResNet

The CNN model that we used to create is ResNet. A residual neural network (ResNet) is an artificial neural network (ANN) of a type, in which constructs are identified from pyramidal cells in the cerebral cortex. Residual neural networks do this by using skip links, or shortcuts to leap over certain layers as illustrated in “Fig.5”. ResNet, i.e., Residual Networks, is a traditional neural network used as a basis for a variety of computer vision challenges. This model won the ImageNet Challenge in 2015. The foundational breakthrough with ResNet has allowed us to successfully train extremely deep neural networks with 150 + layers. Immediately prior to ResNet training, very deep neural networks were difficult due to the issue of gradients dropping. Our work includes implementations of the following ResNet versions and a comparison of the results accordingly.

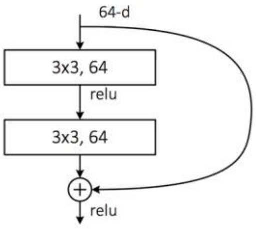

Figure 5. An Original Residual Module

The ResNet9 model consists of four CNN layers, two Residual blocks, and one Linear layer. The image passes through the CNN layer, the Residual node, and is flattened to a 512-size vector after max pooling.
The ResNet15 model consists of six CNN layers, three Residual blocks, and three Linear layers. The image passes through the CNN sheet, the Residual node, and is flattened to a 1028-size vector after max pooling.

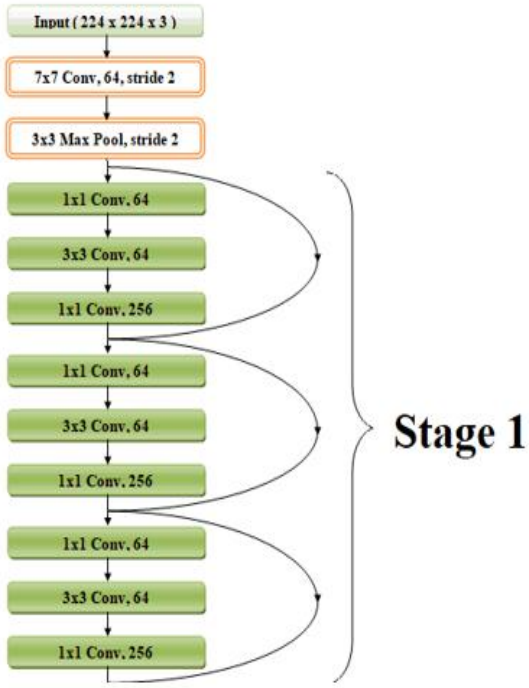

Figure 6. Stage 1 Architecture of ResNet50

The ResNet50 architecture has four stages, as shown in "Fig.6", “Fig.7”, “Fig.8” and “Fig.9”. The network will accept an input image with such a height, width as multiples of 32, and 3 as a channel width. We perceived the input size to be 224 x 224 x 3. Each ResNet architecture uses 7×7 and 3×3 kernel sizes for preliminary convolution and max pooling. Thereafter, Stage 1 of the network begins and has 3 Residual blocks containing 3 layers each. The size of the kernels used to perform the convolution process in all three layers of the stage 1 block is 64, 64, and 128, respectively.
The Curved arrows refer to the identity connection. The dashed linked arrow shows that the convolution process in the Residual Block is done with stride 2, so the input size will be minimized to half in height and width, but the channel width would be multiplied. As we move from one stage to another, the width of the channel is doubled, and the size of the input is whittled down to half.

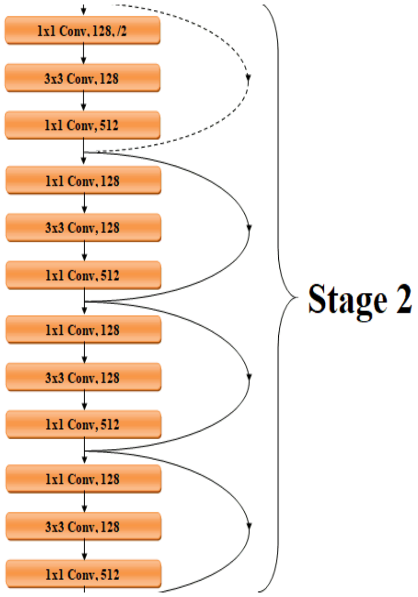

Figure 7. Stage 2 Architecture of ResNet50

Fine-tuning is a transfer learning technique. Knowledge gained during training in one form of the problem is used to learn in another similar task or domain. Fine-tuning ResNet is a 3-step process.

- Pre train the ResNet with ImageNet[16] weights leaving off the fully connected head network. The fully connected layer (FC) acts on a compressed input where all neurons are connected to each input. If present, FC layers are typically located near the end of CNN architectures and can be used to optimize goals such as class scores.
- Create a new, fully connected head and mount it to the base instead of the old head. Now, the architecture is suitable for fine-tuning.

- Freeze the ResNet base layers. The weights of these layers will not be modified during the backpropagation process. Then, the weights of the head layer are tuned.

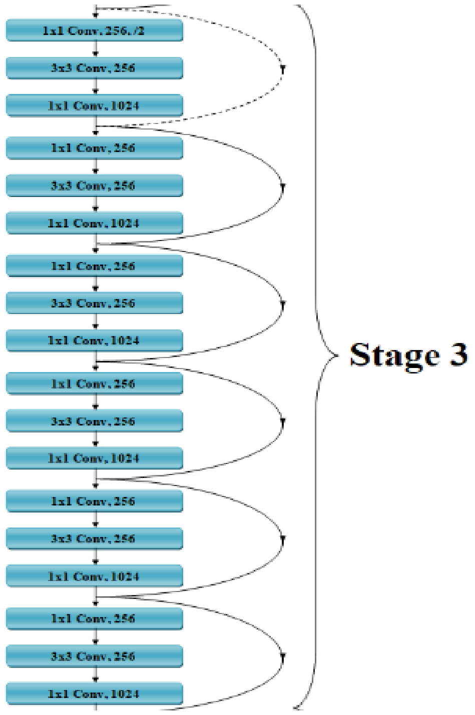

Figure 8. Stage 3 Architecture of ResNet50

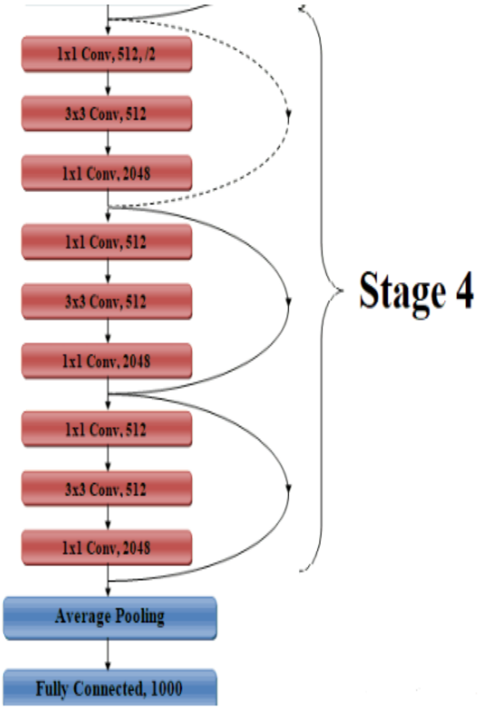

Figure 9. Stage 4 Architecture of ResNet50

3.4 Train the model

Training data will be represented by 75% of all the available data and the remaining data will be marked for testing. Initially, we compile the above model with the learning rate decay and Adam Optimizer using the binary cross-entropy since this is a two-class problem. Now, the model is trained and validated using our training and testing sets.

3.5 Perform Face Detection

Now that our model is well trained, we perform mask detection. But initially, we need to detect the face in the image in order to perform mask detection. For this, we have used Caffe-based face detector [17], which is available in the face detector subdirectory of Deep Neural Network samples. We set a parameter called confidence which is a selectable probability threshold that can be fixed to override 50 % for filtering weak face detections. Once we have predicted where a face is in the image, we try to meet the threshold value before extracting the Region of Interest from the face.

3.6 Extract ROI from each Image

The face ROI[18] was indeed a rectangular shape mounted automatically to cover the face, hair, and neck of the models, while the ROI for each model was allocated to the eye and mouth coordinates within the rectangular areas. Then, we pre-processed the ROI again just like we have done during the training.

3.7 Apply Face Mask Detection

We passed Fig.3 through our constructed model to detect whether that face had a mask. We evaluated the class of the image based on the probabilities returned by the detector and add associated colors for annotation. We draw a bounding box using OpenCV including the class label and the predicted probability. “Fig.10” and “Fig.11” depicts the output of the designed system. Below is the proposed Pseudo Code for the implementation.

```
def face_mask_detection(datadir, valid_size = .2){
 //Resize and Crop images into 224 pixels and Flip them(Data Augmentation)
train_transforms = Random Resize Crop images into 224 px and Random vertical Flip
//Calculate split ratio using length of training data and valid_size
split = length of training set * valid_size
//Split the data into training and testing sets
train_idx, test_idx = drive indices from split
 train_sampler = Sample data from train_idx
 test_sampler = Sample data from test_idx
trainloader = Load training data with batch_size 32
testloader = Load training data with batch_size 32
//Construct ResNet model using resNet50
model = models.resnet50(pretrained=True)
Compile the model using Adam Optimizer
Train the dataset with 20 epoches
epochs = 20
steps = 0
running_loss = 0
print_every = 10
train_losses, test_losses = [], []
for(epoch=1 to epochs){
        for(inputs,labels in train_loader){
          steps = steps+1
                    if (steps % print_every = 0)
          test_loss = 0
          accuracy = 0
                             for(inputs,labels in train_loader){
        calculate batch loss
        calculate test loss from batch loss
        calculate train loss
        }
        }

        }
        Save the model
        Plot the traning and testing loss
        //Detection of face mask
        Load the serialized face detector model
        Load the input image from path
        image = cv2.imread(args["path_of_img"])
        Construct a binary large object from the loaded input image
        Pass this blob to the network and compute face detections and loop over
        detectionsObtained = net.forward()
        for(i=0, detectionsObtained.shape[2]){
        Calculate confidence/Probability associated with that particular detection
        Extract Face ROI
        Pass the face through the above constructed resNet50 model to predict whether it has a mask or not
        Determine class label i.e., Mask or No Mask
        Include the probability
        Display the bounding box along with label and color selected with each label. For Mask, green and for No Mask, red
        Display the final output image.
        }
}
```

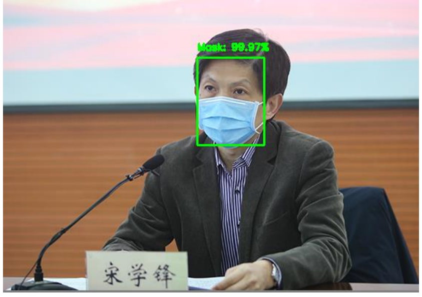

Figure 10. Detection of face mask with 99.97% probability

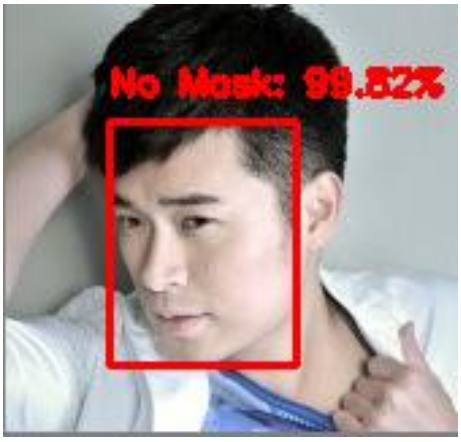

Figure 11. Detection of no mask with 99.82% probability

4. Results and Discussions

As stated in the preceding section, RMFRD is collected that includes masked and unmasked images of people. The type of our work is experimental and it is implemented using real world datasets in Python and this section describes experimental results. The experiments of the proposed smart face mask detection schemes were implemented using deep learning libraries like pytorch, caffe and computer vision libraries like OpenCV in python3, which is necessary and suitable for better accuracy in the process of designing a deep neural network like ResNet. “Fig. 12”, “Fig.13”, “Fig. 14” illustrates the training and validation loss graphs of ResNet9, ResNet15, and ResNet50 respectively. The accuracies achieved using ResNet9, ResNet15, and ResNet50 are 83%, 89%, and 97% respectively when trained our model with 20 epochs and with a batch size of 32. We plot a ROC (Receiver Operating Characteristic) [19] Curve “Fig.15” which illustrates the prediction capability. The ROC curve is obtained by plotting the true positive rate (TPR), often called sensitivity, against the false positive rate (FPR)[20].

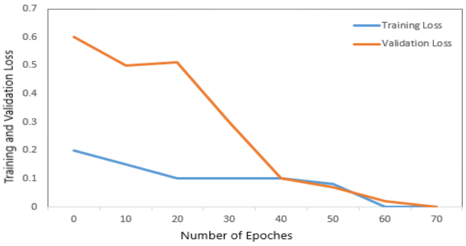

Figure 12. Loss in ResNet9

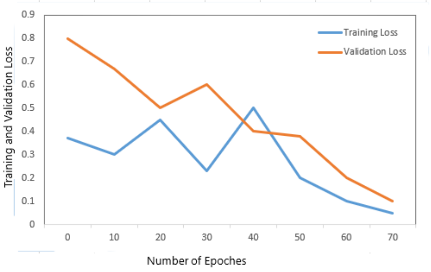

Figure 13. Loss in ResNet15

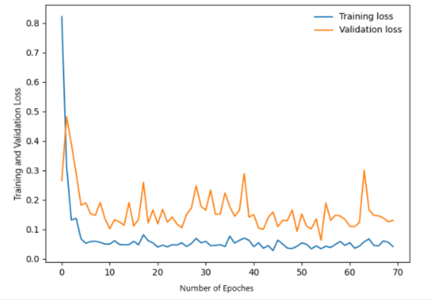

Figure 14. Loss in ResNet50

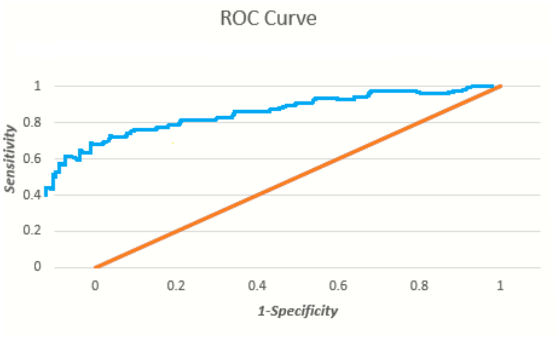

Figure 15. ROC Curve

Table 1 depicts the methodologies used in the proposed system and are listed below based on their detection accuracies. It gives sensitivity, specificity, and accuracy figures of ResNet9, ResNet15, and ResNet50 respectively.

Table 1. Different Neural Networks with Accuracies

Classifier	Training - Testing	Sensitivity	Specificity	Accuracy
ResNet9	80-20 70-30 50-50	86.5 88.4 89.3	80.5 83.2 89.6	83.2 85.9 80
ResNet15	80-20 70-30 50-50	95.7 93.2 96.4	91.3 94.5 89.9	93.5 84.1 89.5
ResNet50	80-20 70-30 50-50	91.3 89.7 93.2	88.9 96.2 94.5	97.8 96.2 97

5. Conclusion

This Face Mask Detection System was found to be apt for detecting whether people wear masks in public places, which contribute to their own health and also to the health of fellow people in this COVID-19 pandemic. This can help assist the government authorities in the process of detection by taking the CC camera footage as input at public places. Our detection system has done very well, when trained on the world’s largest face mask dataset RMFRD and we also presume that it would do better when trained on even larger datasets than RMFRD in the future. We managed to accomplish an overall efficiency of 97%.

Recommendations

As face masks have become a very common part of our lives, it is a mandatory thing with respect to the current pandemic to ensure the safety of ourselves and others as well. Hence, this system can be used by the government to take strict precautions and ensure all of its citizens wear a mask. It can be used in CC TV camera footage at Traffic signals, crowded streets to detect people who don’t wear a mask and impose some sort of action.

Future Research Work

This work can be further extended to detect face shield by changing the Region of Interest used in the current work.

References

[1] Wibisono, T., Aleman, D. M., & Schwartz, B. (2008). A non-homogeneous approach to simulating the spread of disease in a pandemic outbreak. 2008 Winter Simulation.

[2] Chang, Y.-F., Huang, J. C., Su, L.-C., Chen, Y.-M. A., Chen, C.-C., & Chou, C. (2009). Localized surface plasmon coupled fluorescence fibre-optic biosensor for severe acute respiratory syndrome coronavirus nucleocapsid protein detection. 2009 14th Optoelectronics and Communications Conference.

[3] Kim, D., Hong, S., Choi, S., & Yoon, T. (2016). Analysis of transmission route of MERS coronavirus using decision tree and Apriori algorithm. 2016 18th International Conference on Advanced Communication Technology (ICACT).

[4] T.-H. Kim, D.-C. Park, D.-M. Woot. Jeong, and S.-Y. Min, “Multi-class classifier-based adaboost algorithm,” in Proceedings of the Second Sinoforeign-interchange Conference on Intelligent Science and Intelligent Data Engineering, ser. IScIDE’11. Berlin, Heidelberg: Springer-Verlag,2012, pp. 122–127

[5] P. Viola and M. J. Jones, “Robust real-time face detection,” Int. J.Comput. Vision, vol. 57, no. 2, pp. 137–154, May 2004.

[6] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001,vol. 1, Dec 2001, pp. I–I.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014,pp. 580–587.

[8] R. Girshick, “Fast r-cnn,” in Proceedings of the IEEE international conference on computer vision, 2015, pp.1440–1448.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in Neural Information Processing Systems 25, F. Pereira, C. J. C. Burges, L. Bottou, and. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[10] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” CoRR, vol. abs/1409.1556, 2014.

[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” 2015.

[12] A. Jain, A. A. Awan, Q. Anthony, H. Subramoni and D. K. D. Panda, "Performance Characterization of DNN Training using TensorFlow and PyTorch on Modern Clusters," 2019 IEEE International Conference on Cluster Computing (CLUSTER), Albuquerque, NM, USA, 2019, pp. 1-11, doi: 10.1109/CLUSTER.2019.8891042.

[13] M. Komar, P. Yakobchuk, V. Golovko, V. Dorosh and A. Sachenko, "Deep Neural Network for Image Recognition Based on the Caffe Framework," 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP), Lviv, 2018, pp. 102-106, doi: 10.1109/DSMP.2018.8478621.

[14] M. Komar, P. Yakobchuk, V. Golovko, V. Dorosh and A. Sachenko, "Deep Neural Network for Image Recognition Based on the Caffe Framework," 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP), Lviv, 2018, pp. 102-106, doi: 10.1109/DSMP.2018.8478621.

[15] A. Fawzi, H. Samulowitz, D. Turaga and P. Frossard, "Adaptive data augmentation for image classification," 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, 2016, pp. 3688-3692, doi: 10.1109/ICIP.2016.7533048.

[16] M. Ebrahim, M. Al-Ayyoub and M. A. Alsmirat, "Will Transfer Learning Enhance ImageNet Classification Accuracy Using ImageNet-Pretrained Models?," 2019 10th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 2019, pp. 211-216, doi: 10.1109/IACS.2019.8809114.

[17] K. Yan, S. Huang, Y. Song, W. Liu and N. Fan, "Face recognition based on convolution neural network," 2017 36th Chinese Control Conference (CCC), Dalian, 2017, pp. 4077-4081, doi: 10.23919/ChiCC.2017.8027997.

[18] Biswas, A., Khara, S., Bhowmick, P., & Bhattacharya, B. B. (2008). Extraction of regions of interest from face images using cellular analysis. Proceedings of the 1st Bangalore Annual Compute Conference on - Compute ’08. doi:10.1145/1341771.1341787

[19] C. Z. Basha, B. Lakshmi Pravallika, D. Vineela and S. L. Prathyusha, "An Effective and Robust Cancer Detection in the Lungs with BPNN and Watershed Segmentation," 2020 International Conference for Emerging Technology (INCET), Belgaum, India, 2020, pp. 1-6, doi: 10.1109/INCET49848.2020.9154186

[20] C. Z. Basha, M. R. K. Reddy, K. H. S. Nikhil, P. S. M. Venkatesh and A. V. Asish, "Enhanced Computer Aided Bone Fracture Detection Employing X-Ray Images by Harris Corner technique," 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2020, pp. 991-995.

[21] M. A. Turk and A. P. Pentland, “Face recognition using eigenfaces,” in Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Comput. Sco. Press, 1991,pp.586–591.[Online]. Available: http://dx.doi.org/10.1109/CVPR.1991.139758

[22] https://github.com/X-zhangyang/Real-World-Masked-Face-Dataset