Computer Modeling in Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2021.014478

ARTICLE

A Real-Time Integrated Face Mask Detector to Curtail Spread of Coronavirus

Shilpa Sethi[1], Mamta Kathuria[1,*] and Trilok Kaushik[2]

[1]Department of Computer Applications, J. C. Bose University of Science and Technology, Faridabad, 121102, India

[2]R&D Department, Samsung India Pvt., Ltd., Noida, 210301, India

[*]Corresponding Author: Mamta Kathuria. Email: mamtakathuria@jcboseust.ac.in

Received: 30 September 2020 Accepted: 26 February 2021

ABSTRACT

Effective strategies to control COVID-19 pandemic need high attention to mitigate negatively impacted communal health and global economy, with the brim-full horizon yet to unfold. In the absence of effective antiviral and limited medical resources, many measures are recommended by WHO to control the infection rate and avoid exhausting the limited medical resources. Wearing mask is among the non-pharmaceutical intervention measures that can be used as barrier to primary route of SARS-CoV2 droplets expelled by presymptomatic or asymptomatic individuals. Regardless of discourse on medical resources and diversities in masks, all countries are mandating coverings over nose and mouth in public areas. Towards contribution of public health, the aim of the paper is to devise a real-time technique that can efficiently detect non mask faces in public and thus enforce to wear mask. The proposed technique is ensemble of one stage and two stage detectors to achieve low inference time and high accuracy. We took ResNet50 as a baseline model and applied the concept of transfer learning to fuse high level semantic information in multiple feature maps. In addition, we also propose a bounding box transformation to improve localization performance during mask detection. The experiments are conducted with three popular baseline models namely ResNet50, AlexNet and MobileNet. We explored the possibility of these models to plug-in with the proposed model, so that highly accurate results can be achieved in less inference time. It is observed that the proposed technique can achieve high accuracy (98.2%) when implemented with ResNet50. Besides, the proposed model can generate 11.07% and 6.44% higher precision and recall respectively in mask detection when compared to RetinaFaceMask detector.

KEYWORDS

Face mask detection; transfer learning; COVID-19; object recognition; image classification

1 Introduction

The 209th report of world health organization (WHO) published on August 16, 2020 reported that coronavirus disease (COVID-19) caused by acute respiratory syndrome (SARS-CoV2) has globally infected more than 6 million people and caused over 379,941 deaths worldwide [1]. According to Carissa F. Etienne, Director, Pan American Health Organization (PAHO), the key to control COVID-19 pandemic is to maintain social distancing, improving surveillance and strengthening health systems [2]. Recently, a study on understanding measures to tackle COVID-19

pandemic carried by researchers at University of Edinburgh reveals that wearing face mask or other covering over nose and mouth cuts risk of Coronavirus spread by avoiding forward distance travelled by person's exhaled breath by more than 90% [3]. Steffen et al. also carried an exhaustive study to compute the community-wide impact of mask use in general public, a portion of which may be asymptomatically infectious in New York and Washington. Their works show that near universal adoption (80%) of even weak mask (20% effective) could prevent 17%–45% of projected deaths over two months in New Work and reduces the peak daily death rate by 34%–58% [4,5]. Their results strongly recommend use of face mask in general public to curtail spread of Coronavirus. Further, with reopen of countries from COVID-19 lockdown, Government and Public health agencies are recommending face mask as essential measures to help keep us safe when venture into public. To mandate the use of facemask, it becomes essential to devise some techniques that enforce individual to apply mask before exposure to public places.

The face mask detection principally refers to detect whether a person is wearing a mask or not. The preliminary stage for analyzing the mask wearing includes face detection. Many Machine Learning algorithms have been proposed in past to analyze face for the purpose of security, authentication and surveillance; the mask wearing at large scale can make ineffective such systems. In fact, the early efforts in face detection have dated back in 1973 [6]. Using the design of handcraft features and machine learning algorithms to train effective classifiers for detection and recognition was proposed by Nanni et al. [7]. The problems encountered with this approach include high complexity in feature design and low detection accuracy. A systematic review on face detection system is presented in [8]. In recent years, face detection methods based on deep convolutional neural network (CNN) have been widely developed [9–12] to improve detection performance.

Although numerous researchers have committed efforts in designing efficient algorithms for face detection and recognition but there exists essential difference between 'detection of face under mask' and 'detection of mask over face.' As per available literature, a very little body of research is attempted to detect mask over face. Thus, our work aims to develop technique that can accurately detect mask over face in public areas (such as airports. railways stations, crowded markets, bus stops) to curtail spread of Coronavirus and thereby contributing to public healthcare.

Further, it is not easy to detect faces with/without mask in public as dataset available for detecting mask on human faces is relatively small leading to hard training of model. So, concept of transfer learning is used here to transfer the learned kernels from networks trained for a similar face detection task on an extensive dataset. The dataset covers various faces images including faces with masks, faces without masks, faces with and without masks in one image and confusing images without masks. With extensive dataset containing 45,000 images, our technique achieves outstanding accuracy of 98.2%. The major contribution of proposed work is given below:

1. A novel object detection module that combines one stage and two stage detectors based on image complexity is capable of accurately detecting the object in real-time from video streams.
2. Improved affine image wrapping technique is developed to crop the facial areas from uncontrolled real-time images having differences in face size, orientation and background.
3. Creation of unbiased facemask dataset with imbalance ratio equals to nearly one.
4. The proposed model requires less memory, making it easily deployable for embedded devices used for surveillance purpose.

The rest of this paper is organized in Sections as follows. Section 2 covers prevalent literature in the field of object recognition. The proposed methodology is presented in Section 3. Section 4 evaluates performance of proposed technique with various pre-trained models over different parameters of speed and accuracy. Finally, Section 5 concludes the work with possible future directions.

2 Related Work

Recently, embedded systems equipped with CCTV camera and computer vision have gained popularity in wide range of applications involving facial recognition, traffic control and monitoring, intrusion detection, additional analytics applications such as smoke detection, unattended baggage, queue management, etc. These applications require open deployment environment that are capable of parsing different scenes, locating objects and taking real-time actions. These surveillance applications correspond to two popular research areas of computer vision namely object recognition and image classification. Further, automatic scene parsing through variety of surveillance platforms brings many new challenges in the area of object recognition [13]. Three main challenges are: i) how to handle various object appearances caused by different orientation, illumination, shadow and size ii) how to efficiently deploy object detection models on surveillance platform with limited computational power and memory iii) how to perform surveillance action in real-time without loss of accuracy.

One viable approach to deal with these challenges is object recognition using deep learning methods. The object recognition using deep learning requires generation of region proposals in a scene followed by classification of each proposal into related class as shown in Fig. 1. Over the years, there has been much advancement proposed by researchers in region proposals techniques to suit variety of applications. We review the recent developments in region proposal techniques using single stage and two stage detectors and application of these techniques for face and mask detection task.

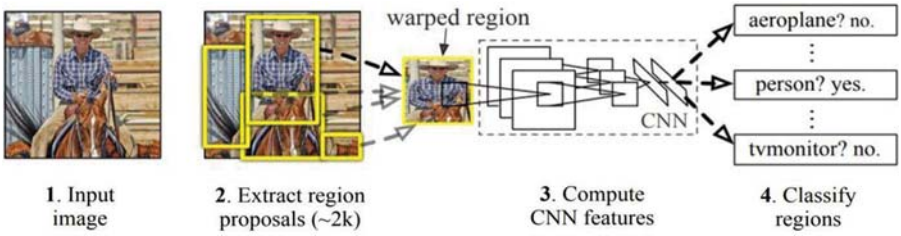

Figure 1: General pipeline for object recognition using deep learning

Single stage detectors: The single-stage detectors treat detection of region proposals as a simple regression problem by taking the input image and learning the class probabilities and bounding box coordinates. OverFeat [14] and DeepMultiBox [15] were early examples. YOLO (You Only Look Once) popularized single stage approach by demonstrating real-time predictions and achieving remarkable detection speed but suffered from low localization accuracy; especially when small objects are taken into consideration [16]. Chun et al. [17] analyzed the performance of YOLOV3 for face detection in complex environment using WIDER FACE. The experimental analysis reveals that the confidence set in YOLOV3 is relatively large due to being trained on COCO dataset. Since WIDER FACE contains images of small and medium sized faces mainly so high confidence setting in YOLOV3 missed detection of many small faces. However, multiple

clustering adjustments on priori box had improved the accuracy in face detection from 57.9% to 80.5%. Alex Escola builds a face mask detector using YOLOV5 on Facemask Detection dataset available on Kaggle [18]. Since the dataset contains only 853 images so, data set is enriched with images from COCO dataset. But the model struggles to detect facemask under certain conditions such as detection of small faces with masks and confusing masks, for example persons long beard etc. Addagarla et al. [19] compared the performance YOLOV3 with NASNetMobile for facemask detection and found that NASNetMobile achieves better accuracy of 91.7% as compared to YOLOV3. Further, Single-Shot Detector (SSD) can predict region proposals and class probabilities simultaneously thereby improving detection accuracy as compared to YOLO [20]. The major difference between SSD and YOLO lies in dealing with objects of variant sizes. YOLO detects objects of different scales in separate layers of network, while SSD runs detection on only top layer irrespective of object size. The work in [21] involves SSD Multibox detector trained on ImageNet and PascalVoc for high quality facemask classification. Andrian Rosebrock proposed a face mask detection system using SSD over a synthetic dataset [22]. Here, SSD is used to create list of bounding boxes around each detected face and MobileNetV2 is used to classify the face into two classes, namely with or without mask. Although, the lighter version of YOLO and SSD series such as YOLO-Lite, YOLO-tiny and tiny SSD are available, detection accuracy of these networks is low. Therefore, how to deploy small detection model without notably decreasing the accuracy on embedded devices for surveillance application such as facemask detection for controlling spread of Coronavirus needs an urgent attention.

Two stage detectors: In contrast to single stage detectors, two stage detectors follow a long line of reasoning in computer vision for prediction and classification of region proposals. They first predict proposals in an image and then apply a classifier to these regions to classify potential detection. Various two stage region proposal models have been proposed in past by researchers. The region-based convolutional neural network also abbreviated as R-CNN described in 2014 by Ross Girshick et al. [23]. Basically, R-CNN applies a selective search algorithm to extract a set of object proposals at the initial stage and applies SVM (Support Vector Machine) classifier for predicting objects and related classes on later stage. Wu et al. [24] proposed a face detection method based on R-CNN and resolved small scale face detection using multi-scale training, feature concatenation and hard negative mining. Roy et al. [13] compared the performance of YOLOV3, YOLOV3-tiny and R-CNN using Moxa3K benchmark dataset for monitoring of people wearing masks. In his work, Inception V2 is taken as the backbone with R-CNN using Tensor flow. The mAP@50 (Mean Average Precision) score obtained by R-CNN with inception V2 was 63.99%, YOLOV3 configured in Darknet was 60.5% and YOLOV3-tiny with Darknet was only 56.57%. Although R-CNN yields high performance for facemask detection but limited by low detection speed. Spatial pyramid pooling SPPNet [25] (modifies R-CNN with an SPP layer) collects features from various region proposals and fed into fully connected layer for classification. The capability of SPNN to compute feature map of entire image in single shot resulted in significant improvement in object detection speed by magnitude of nearly 20 folds greater than R-CNN.

Next, Fast R-CNN [26] is an extension over R-CNN and SPPNet. It introduces a new layer named Region of Interest (RoI) pooling layer between shared convolutional layers to fine-tune the model. Moreover, it allows to simultaneously train a detector and regressor without altering the network configurations. Although Fast-R-CNN effectively integrates the benefits of R-CNN and SPPNet but still lacks in detection speed compared to single stage detectors [27]. Further, Faster R-CNN is amalgam of fast R-CNN and Region Proposal Network (RPN). It enables nearly

cost-free region proposals by gradually integrating individual blocks (e.g., proposal detection, feature extraction and bounding box regression) of object detection system in single step [28,29].

Although this integration leads to accomplishment of break-through for the speed bottleneck of Fast R-CNN but there exists a computation redundancy at subsequent detection stage. Jiang et al. [28] trained Faster R-CNN using WIDERFACE dataset for face detection and verified the performance over FDDB dataset and IJB-A benchmarks. The Region-based Fully Convolutional Network (R-FCN) is the only model that allows complete back-propagation for training and inference [30,31]. Feature Pyramid Networks (FPN) can detect non-uniform objects but least used by researchers due to high computation cost and more memory usage [32]. Furthermore, Mask R-CNN strengthens Faster R-CNN by including the prediction of segmented masks on each RoI [33]. Face detection and segmentation based on improved Mask R-CNN was proposed in [34]. The method integrates segmentation stages with bounding box localization to crop the face from its background in single go. Qin et al. [35] proposed SRCNet a two-stage detector for identifying correctly wear facemask. The first stage involves upscaling low resolution or blurred images using SR network to produce high quality feature maps whereas second stage involves facemask wearing condition using MobileNetV2. Being the lightweight CNN, MobileNetV2 has residual blocks and depth-wise separable convolutions. The residual blocks contribute to the training of deep neural network using CelebA database and depth of network help in distinction of incorrectly wear facemask (IWF) from correctly wear facemask (CWF).

Techniques for improving detectors: Several techniques for improving performance of single stage and two stage detectors have been proposed in past [36]. Easiest among all is cleaning the training data for faster convergence and moderate accuracy. Hard negative sampling technique is often used to provide negative samples for achieving high final accuracy [37]. Modification in context information is another approach used to improve detection accuracy or speed. MS-CNN [38], DSSD [39] and TDN [40] strengthen the feature representation by enriching context of coarser features by including an addition layer in top-down for better object detection. BlitzNet improved SSD by adding semantic segmentation layer to achieve high detection accuracy [41]. A hybrid deep learning-based approach using R-CNN for feature extraction and decision tree, SVM and ensemble are proposed in [42] over LFW dataset. The model achieves reasonably good accuracy on small dataset but requires more memory.

The object detection architectures discussed so far have several open-source models which are pre-trained on large datasets like ImageNet [43], COCO [44] and ILSVRC [45]. These open-source models have largely benefitted in the area of computer vision and can be adopted with minor extensions to solve specific object recognition problem thereby avoiding everything from scratch. Fig. 2 summarizes various pre-trained models based on CNN architectures released from 2012 to 2019.

These models vary in terms of baseline architecture, number of layers, inference speed, memory consumption and detection accuracy. The achievement of each model is mentioned in Fig. 2. In order to enforce mask over faces in public areas to curtail community spread of Coronavirus, a deep learning approach based on available pre-trained model is highly recommended for welfare of the society. The concept of transfer learning for facemask detection using InceptionV3 is presented in [46]. The reason for adopting transfer learning was limited availability of facemask wearing dataset which might cause overfitting problem. The model is trained and tested on a simulated Masked Face Dataset (SMFD). The work in [47] applied transfer learning and content attention using pre-trained MobileNetV2 model over MAFA dataset and achieve a precision of 82.3%. These pre-trained models are required to be finely tuned with benchmark datasets. The number

of datasets with diverse feature pertaining to human faces with and without mask are given in Tab. 1.

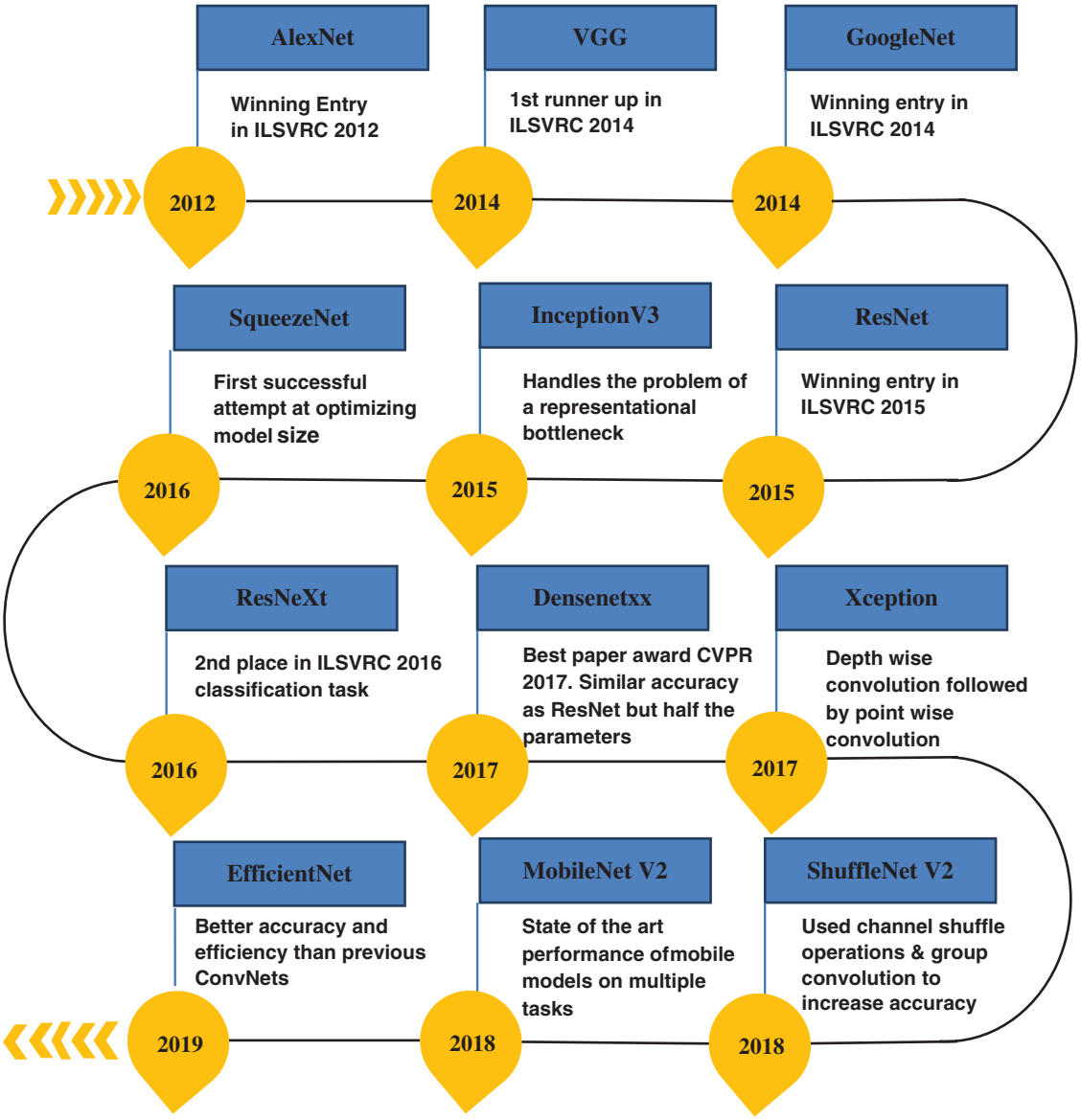

Figure 2: Various pre-trained models based on CNN architecture

An extensive study conducted on available face related datasets reveals that there exist principally two kinds of datasets. These are: i) face recognition dataset and ii) face mask datasets. The face recognition datasets can be used for face identification and authentication tasks whereas face mask datasets are purposely designed for identification of mask over face.

Face mask datasets are mainly populated with images in which the face is obscured by some objects or mask, e.g., the MAsked FAces dataset (MAFA), the Real-World Masked Face Dataset (RMFD), Masked Face Detection Dataset (MFDD), Real-world Masked Face

Recognition Dataset (RMFRD), Simulated Masked Face Recognition Dataset (SMFRD) and MaskedFace-Net.

Tab. 1 summarizes these two kinds of prevalent datasets.

Table 1: Different categories of datasets

Type of datasets	Dataset	Scale	#Faces	#Masked face images
Face recognition	FDDB [48]	2845	5171	–
	MALF [49]	5250	11931	–
	CelebA [50]	200000	202599	–
	WIDERFACE [9]	32203	194000	–
Face masked	MAFA [51]	30811	37824	35806
	RMFRD [52]	95000	9200	5000
	SMFRD [52]	85000	5000	5000
	MFDD [52]	500000	500000	24771
	MaskedFace-Net [53]	139646	–	137016

The following shortcomings are identified after critically observing the available literature,

1. Although there exist several open-source models that are pre-trained on benchmark datasets but only a few models [47] are currently capable of handling COVID related face masked datasets.
2. The available face masked datasets are scarce and need to strengthen with varying degree of occlusions and semantics around different kinds of masks.
3. Although there exist two major types of state of art object detectors: single stage detectors and two stage detectors. Both methods have certain benefits and limitations over each other. Single stage detectors are fast but limited by low accuracy whereas two stage detectors produce accurate results even for complex inputs but at the cost of computational time. So, an optimal technique needs to be devised that can be easily deployed for object detection on a surveillance platform with less memory consumption and perform surveillance in real-time without a notable reduction in accuracy.

To solve these problems, a deep learning model based on transfer learning trained on a highly tuned customized face mask dataset is being proposed and discussed in detail in the next Section.

3 Proposed Architecture

The proposed model is based on object recognition benchmark in [53]. According to this benchmark, all the tasks related to an object recognition problem can be ensemble under three main components: Backbone, Neck and Head as depicted in Fig. 3. Here, the backbone corresponds to baseline convolutional neural network capable of extracting features from images. Since training a convolutional neural network is expensive in terms of computational power and time; transfer learning is applied here. Transfer learning allows to transfer the trained knowledge of the pre-trained neural network in terms of parametric weights to the new model. In order to obtain best results for facemask detection, the experiment is setup with three popular pre-trained models namely ResNet50, MobileNet and AlexNet separately. Although emerging in 2012, AlexNet was considered in our experiments to show the progress in term of performances by comparison with more recent architectures It is experimentally observed that ResNet50 is optimal choice in terms

of accuracy, inference time and memory as compared to AlexNet and MobileNet for our targeted problem (refer Section 4.2).

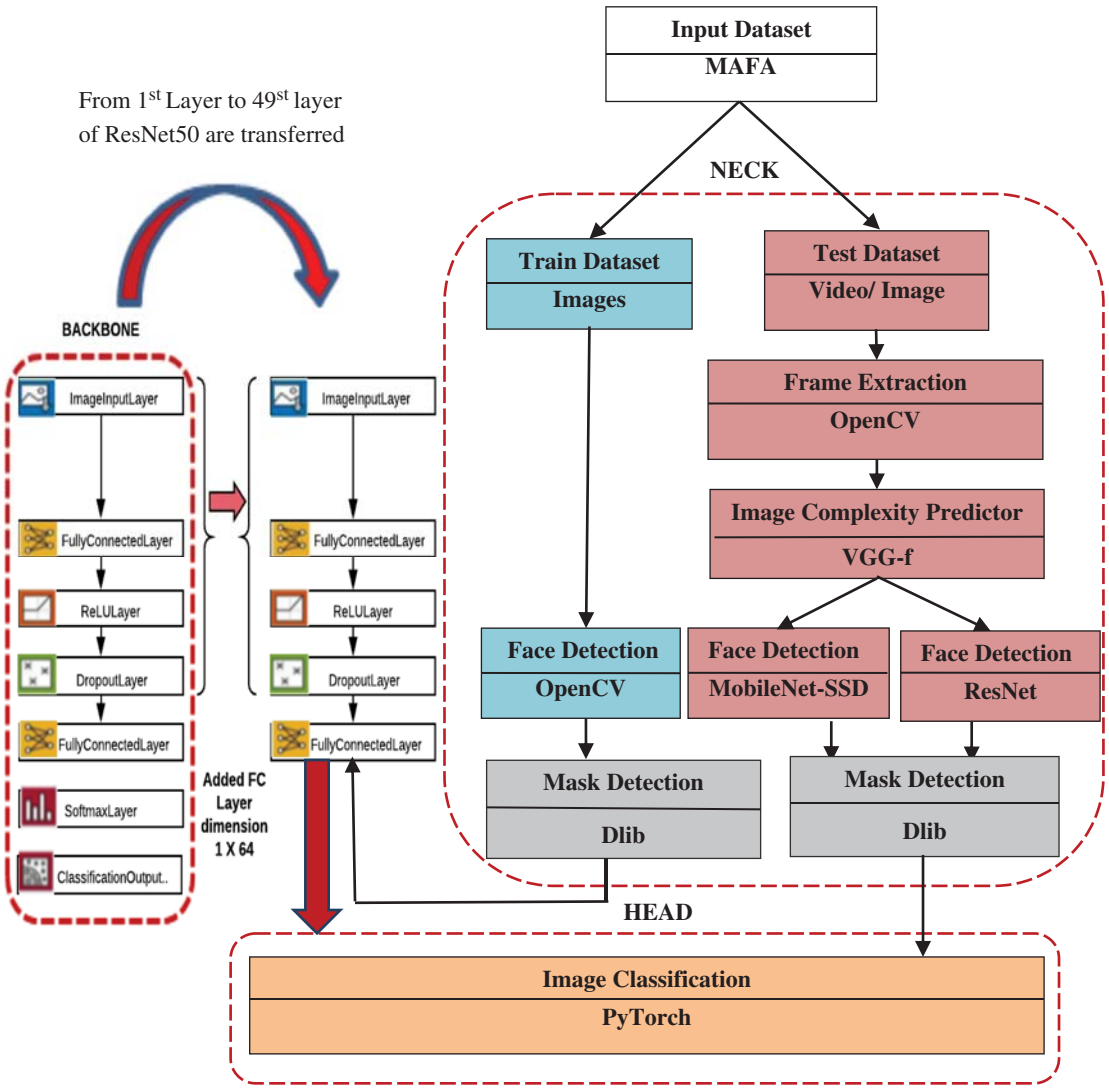

Figure 3: Proposed architecture

The novelty of our work is being proposed in the Neck Component. The intermediate component, Neck contains all those pre-processing tasks that are needed prior to the actual classification of images. Since the number and kind of occlusions can impact the performance of face and mask detection so, an image complexity predictor is being proposed here. The complex images are accurately processed using two stage ResNet50 detector whereas simple images are quickly processed through one stage, MobileNet-SSD detector (refer to Section 3.3). Further, the affine image wrapping technique is applied to crop the detected region into a fixed size anchor bounding box (refer to Section 3.4). Following the above steps, the proposed model is able to achieve a gain of nearly 11% in precision and nearly 6% in recall as compared to existing facemask

detection model under similar experimental setup (refer to Section 4.4). The final precision of proposed model for face detection is 99.2% and for mask detection is 98.92% over MAFA Dataset containing 30,811 images. The final recall of proposed model for face detection is 99% and for mask detection is 98.24%.

Furthermore, OpenCV with Dlib is used to detect a variety of face masks in cropped images. Finally, high level semantically extracted information obtained after mask detection is used to fine-tune the backbone. The last component, Head stands for image classifier, detector or predictor that can achieve the desired objective of deep learning neural network. In the proposed architecture, PyTorch is applied to categorize multiple faces with different types of masks with non-uniform sizes contained in each input image in just a single shot thereby leading to a fast detection system. Further, the output of the head is through a fully convolutional network rather than a fully connected network resulting in reduced parameter cost.

3.1 Construction of Face Masked Dataset

A facemask centric dataset, MAFA [51] with total 25,876 images categorized over two classes with 23,858 masked and 2018 non-masked images is considered for training the proposed mask recognition model. It is observed that MAFA is put up with an extrinsic class imbalanced problem that may introduce a bias towards the majority class. So, an ablation study is conducted to analyses the performance of image classifier once with original MAFA set (imbalanced) and then with proposed dataset (balanced).

3.1.1 Supervised Pre Training

We discriminatively pre-trained the CNN on the original MAFA dataset. Pre-training was performed using the open-source Caffe Python library [54]. In short, our CNN model nearly matches the performance of Jia et al. [55], obtaining a Top_1 error rate 1.8 percentage higher on MAFA validation set. This discrepancy may cause due to a simplified training approach.

3.1.2 Supervised Pre Training with Domain-Specific Fine-Tuning

The other approach is to first remove the inherent bias present in the available dataset and then execute supervised learning over a domain-specific balanced dataset. The bias is alleviated by applying random over-sampling (ROS) with data augmentation. The technique reduces the imbalance ratio $\rho = 11.82$ (original) to $\rho = 1.07$. The formula used for computing the imbalance ratio is given by Eq. (1).

$$\rho = \frac{\text{Count}\,(\text{majority}\,(D_i))}{\text{Count}\,(\text{minority}\,(D_i))} \tag{1}$$

Here, D refers to image Dataset, majority (Di) and minority (Di) return the majority and minority class of D. Count(X) returns number of images in any arbitrary class x. After data balancing, stochastic gradient descent (SGD) training of CNN parameters with learning rate of 0.003 is set over wrapped region proposals. The low learning rate allows fine tuning of model without clobbering the initialization. We added 2025 negative windows with 50 background windows to increase non mask dataset ≈ 22 KB. The balancing leads to reduction in Top_1 error rate of 3.7%.

3.2 Transfer Learning

Due to scarcity of large facemask-centric datasets and to employ the strengths of influential deep convolutional neural networks, the proposed model is built over ResNet50 and able to

transfer functionality, features and weights learnt from first 49 layers to newly added fully connected convolutional layer with dimension 1×64. Transfer learning leads to achieve good results even with optimal dataset, since basic face features have already been learnt by ResNet50 from a much larger dataset like ImageNet. In proposed model, only the parameters of backbone and head are initialized using ResNet50 and neck is customized to perform face and mask detection tasks.

For this work, the last layer of ResNet50 is replaced by adding four more layers. These are: a pooling layer of pool size $= 5 \times 5$, a flattering layer, a dense layer of 128 neurons and finally a decisive layer with softmax activation function are added to classify a face into mask/non-mask.

3.3 Image Complexity Predictor for Face Detection

To address problem 3 identified in Section 2, regarding the low accuracy of single stage object detector and high computational time involved in two stage object detectors, an optimal technique needs to be devised. For this purpose, various face images are analyzed in terms of processing complexity and it is observed that the dataset we consider primarily contains two major classes, i.e., mask and non-mask class but the mask class further contains inherent variety of occlusions other than surgical/cloth facemask e.g., occlusion of ROI by other objects like a person, hand, hair or some food items as depicted in Fig. 4.

These occlusions are found to impact the performance of face and mask detection. Thus, obtaining an optimal trade-off between accuracy and computational time for mask detection is not a trivial task. So, a non-negative complexity predictor (P) linked to each image is being proposed here. Its purpose is to split data into soft verses hard images at initial level followed by mask and non-mask classification at later level as shown in Fig. 4 above. The question that arises here is how to determine whether an image is hard or soft. The answer to this question is provided by a recent work proposed by Ionescu et al. [56] in which an image hardness is estimated based on the difficulty of visual search inherent in an image. The features affecting the performance of visual search include image resolution, image density and object proportion. Here, image density refers to number of objects present in the image and object proportion refers to relative dimension of reference object as a part or whole. Based on these features, a non-negative integer is assigned to each image. If the integer is less than the threshold, the image is put in soft category else it is a hard image. After separation, a soft image is processed through a fast single stage detector whereas a hard image is accurately processed by two stage detector. We employ MobileNet-SSD model for predicting the class of soft images and faster R-CNN based on ResNet50 for making predictions for hard images.

The algorithm for optimal face detection is given in Fig. 5. Further, image complexity predictor is built using pre-trained VGG-f with ν-support vector regression. The last layer of VCG-f is replaced by a fully connected layer. The 4096 features extracted from this layer are then normalized using L2-norm. The obtained normalized feature vectors are further used to regress the ground truth complexity score as proposed in [57]. Tab. 2 summarizes mAP score and Computation time for various combinations of MobileNet and ResNet50 over MAFA test dataset.

The various combinations are made by splitting the test dataset into different proportion of images processed by each detector starting from pure MobileNet (100%–0%) to three intermediate splits (75%–25%, 50%–50%, 25%–75%) to pure ResNet50 (0%–00%).

Here, the test data is partitioned based on random split or soft versus hard split given by Image Complexity Predictor. In order to reduce bias, the average mAp over 5 runs is recorded for the random split. The elapsed time is measured on Inter I7, 2.5 GHZ CPU with 16 GB RAM.

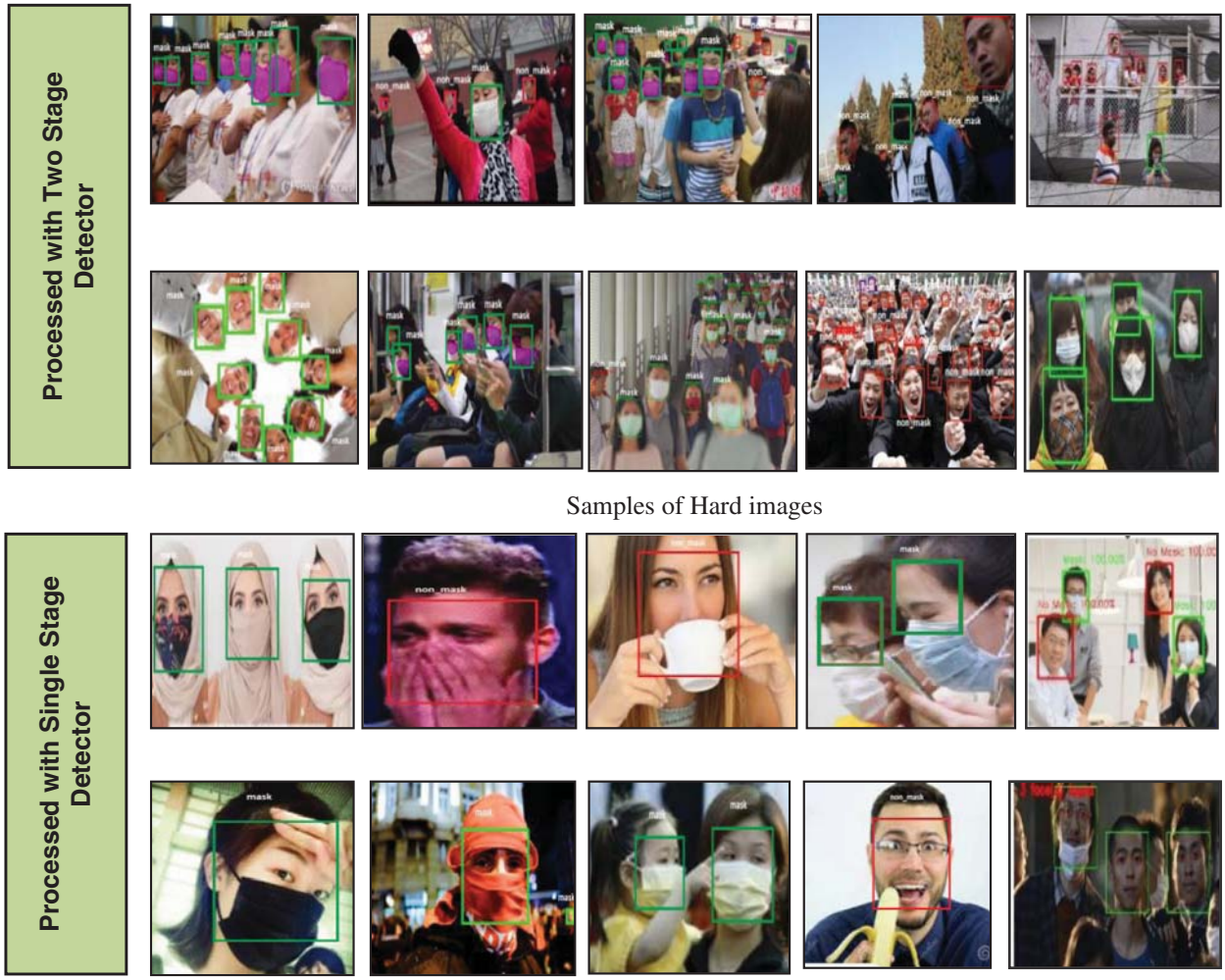

Figure 4: Variety of occlusions present in proposed dataset

Algorithm: OptimalFaceDetector ()
Input:
Image← input image
D_{fast} ← single stage detector
D_{slow} ← two stage detector
CP← Image complexity predictor
τ ←hardness threshold
Computation:
if (*CP*(*Image*) > τ)
$R \leftarrow D_{slow}(Image)$
else
$R \leftarrow D_{fast}(Image)$
Output:
R ← set of region proposals

Figure 5: Algorithm for optimal face detection

Table 2: Comparison between mobilenet-ssd and resnet50 and their various combination based on random *vs*. hard/soft complexity of test data

Comparison parameters	MobileNet-SSD to ResNet50 (left to right)				
	100%–0%	75%–25%	50%–50%	25%–75%	0%–100%
Random_split (mAP)	0.8868	0.9095	0.9331	0.9650	0.9899
Soft/hard_split (mAP)	0.8868	0.9224	0.9631	0.9892	0.9899
Image complexity prediction time (ms)	–	0.05	0.05	0.05	–
Mask detection time (ms)	0.05	1.92	3.08	5.07	6.02
Total computation time (ms)	0.05	1.97	3.13	5.12	6.02

3.4 Bounding Box Regression

After detecting a face in the search proposal, we apply affine transformation to obtain a bounding box of fixed size irrespective of the aspect ratio of the candidate region. The purpose of applying bounding box is to improve localization performance during mask detection. The technique is similar to deformable part models described in [12]. The primary difference between the two methods is that the proposed model regress from features computed by CNN rather than just applying geometric features taken from DPM part locations.

Let each region proposal (face) be represented by a pair (R, G), where $\mathrm{R} = (R_x, R_y, R_w, Rh)$ represents the pixel coordinates of the center of proposals along with width and height. Each ground truth bounding box is also represented in the same way, i.e., $G = (G_x, G_y, G_w, G_h)$. So, the goal is to learn a transformation that can map region proposal (R) to ground-truth bounding box (G) without loss of information. We propose to apply a scale-invariant transformation on pixels coordinates of R and log space transformation on width and height of R. The corresponding four transformation functions are represented as $T_x(R)$, $T_y(R)$, $Tw(R)$ and $T_h(R)$. So, coordinates of the ground truth box can be obtained by Eqs. (2)–(5).

$$G_x = T_x(R_x) + R_x \tag{2}$$

$$G_y = T_y\left(R_y\right) + R_y \tag{3}$$

$$G_w = T_w(R_w) + R_w \tag{4}$$

$$G_h = T_h(R_h) + R_h \tag{5}$$

Here, each T_i (where i denotes one of x, y, w, h) is applied as a linear function of the Pool 6 feature of R denoted by $f6\,(R)$. Here, the dependence of $f6\,(R)$ on R is implicitly assumed. Thus, $T_i(R)$ can be obtained by Eq. (6).

$$T_i(R) = w_i f_6(R) \tag{6}$$

where W_i denotes the weight learned by optimizing the regularized least square objective of ridge regression and is computed by Eq. (7). Ridge regression is used here, to penalize the variables with minor contribution to the outcome; have their coefficient close to zero. One popular penalty is to panelize the model based on sum of squared coefficient values; this is called L2 penalty ($\widehat{w}_i$). The $\widehat{w}_i$ minimize the size of all coefficients and preventing the coefficient from being removed from the model. Further, a hyperparameter called tuning parameter or penalty parameter (λ) is used to control the weighing of penalty to the loss function. A default value of $\lambda = 1.0$ will fully

weight the penalty whereas $\lambda = 0$ excludes the penalty. The scikit-learn library in Python is used to automatically finds good value for λ via RidgeCV class. For our model. λ is set to 0.51.

$$w_i = \sum_{n \in R} (t_i^n - \widehat{w} f_6(R^n))^2 + \lambda |\widehat{w}_i|^2 \tag{7}$$

The regression target (t_i) related to coordinates, width and height of region proposal pair (R, G) are defined by Eqs. (8)–(11), respectively.

$$t_x = (G_x - R_x)/R_w \tag{8}$$

$$t_y = \left(G_y - R_y\right)/R_h \tag{9}$$

$$t_w = \log(G_w/R_w) \tag{10}$$

$$t_h = \log(G_h/R_h) \tag{11}$$

3.5 Loss Function and Optimization

Defining the loss function for the classification problem is among the most important part of the convolutional neural network design. In classification theory, a loss function or objective function is defined as a function that maps estimated distribution onto true distribution. It is desirable for an optimization algorithm to minimize the output of this function. The stochastic gradient descent optimization algorithm is applied to update the model parameters with a learning rate of 0.03. Further, there exist numerous loss functions in PyTorch but one which is most suitable with balance data is a cross-entropy loss. Furthermore, an activation function is required at the output layer to transform the output in such a way that would be easier to interpret for the loss function.

Since the formula for cross-entropy loss given in Eq. (12) takes two distributions, t(x), the true distribution and e(x), the estimated distribution defined over discrete variables x [58], thus activation functions that are not interpretable as probabilities (i.e., negative or greater than 1 or sum of output not equals to 1) should not be selected. Since Softmax guarantees to generate well behaved probabilities distribution over categorial variable so it is chosen in our proposed model.

$$\text{Loss} = \sum_{\forall_x} t\,(x) \log\,(e\,(x)) \tag{12}$$

Further, the loss function over N images (also known as cost function over the complete system) in binary classification can be formulated as given in Eq. (13).

$$\text{Loss} = \frac{1}{N} \sum_{x} \sum_{n=1}^{N} t_n\,(x) \log(e_n\,(x) \tag{13}$$

4 Performance Evaluation

To evaluate the performance of the proposed model, the experiment is conducted to answer the following research questions:

RQ1: Which model will best fit as a backbone for detecting facemask using transfer learning?

RQ2: How does our model perform compared to existing face mask detection model in terms of accuracy and computational speed?

RQ3: What measures should be considered to avoid overfitting?

4.1 Experimental Setup

The experiment is setup by loading different pre-trained models using Torch Vision package (https://github.com/PyTorch/vision). These models are trained on a fine-tuned MAFA dataset using the open-source Caffe Python library. We choose MAFA dataset with 25,876 images and label 37,826 faces with a high degree of variability in occlusion as depicted in Fig. 4. Int-Scenario training/testing strategy is adopted as employed in [9]. The fine-tuned MAFA dataset is split into training, testing and validation sets with 64:20:16, respectively. The algorithms are implemented using Python 3.7 and face detection is achieved through open source library (OpenCV) during the training phase. .dib is used for detecting masks with learning rate = 0.003, momentum = 0.9 and batch size = 64. PyTorch is used for image classification.

4.2 Effectiveness of Different Pre-Trained Models

As discussed in Section 3.2, that we can apply transfer learning on pre-trained models for feature engineering but one question that yet to answer is how we can decide which model is effective for our task. In this Section, we will compare three efficient models namely ResNet50, AlexNet and MobileNet on based on the following criteria:

1. **Top_1 Error:** This type of error occurs when the class predicted with the highest confidence is not the same as the true class.
2. **Inference Time on CPU:** Inference time is the time taken by the model to predict the class of input image, i.e., starting from reading the image, performing all intermediate transformations and finally generating the high confidence class to which the image belongs.
3. **Model size:** It refers to physical space occupied by the **.pth** file of pre-trained models supplied by PyTorch.

A model with **minimum Top_1 error,** less **inference time** on CPU and **less model size** will be considered as a good model for our work. The confusion matrix results for different models are summarized in Tab. 3. The accuracy comparison of various models based on Top_1 error is presented graphically in Fig. 6a. It may be noted from the graph that the error rate is high in AlexNet and almost equal in MobileNet and ResNet50. Next, we compared the models based on inference time. All test images are supplied to each model and inference time for all iterations is averaged out.

Table 3: Confusion matrix obtained using different models

Results	(Predicted)					
	AlexNet		MobileNet		ResNet50	
	Mask	Non mask	Mask	Non mask	Mask	Non mask
Mask (actual)	TP: 4351	FP: 103	TP: 4669	FP: 48	TP: 4657	FP: 51
Non mask (actual)	FN: 227	TN: 4518	FN: 104	TN: 4378	FN: 83	TN: 4403

The process was repeated once for CPU and then for GPU on Google Colab. It may be observed from Fig. 6b that MobileNet takes more time to inference images whereas ResNet and AlexNet take almost equal time for inferring the images.

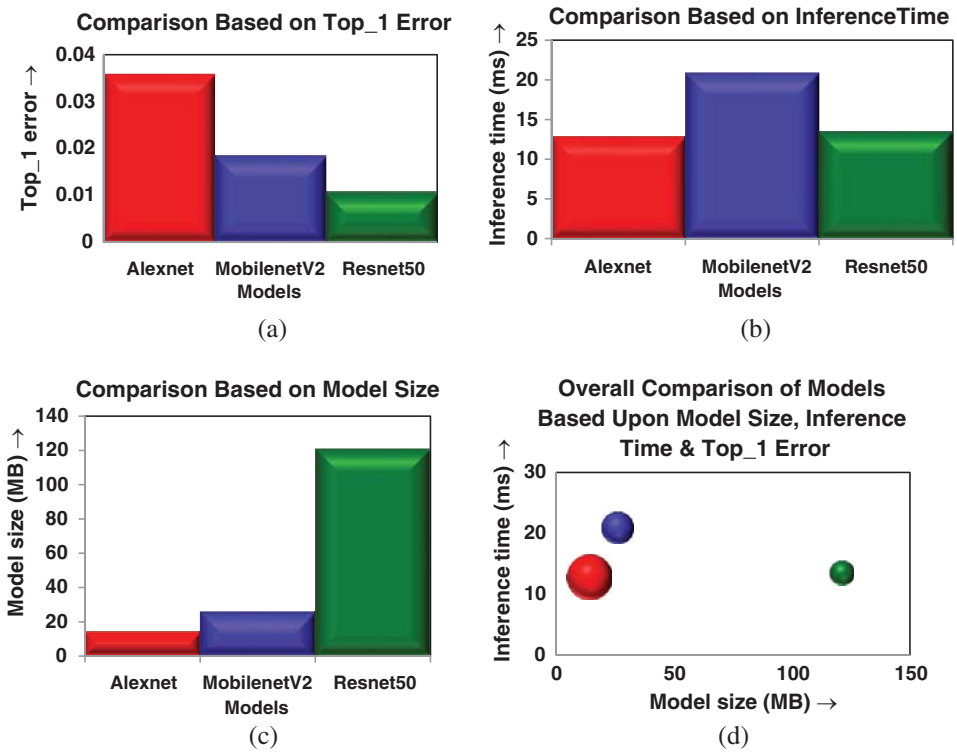

Figure 6: Comparison of various models on different criteria. (a) Top_1 error (b) inference time (c) model size (d) overall comparison

Further, it may be noted in Fig. 6c that AlexNet has a minimum size of .pth file (14 MB) followed by MobileNet (26 MB) and ResNet (121 MB). After analyzing the performance of each model on various criteria, we then squeeze all these details into single bubble chart by taking model size as X-coordinate and inference time as Y-coordinate. The bubble size represents Top_1 error (less is better). The overall comparison of all models is represented by a bubble graph in Fig. 6d.

It may be observed from Fig. 6, smaller bubbles are better in terms of accuracy and bubbles near the origin are better in terms of memory requirement and speed. Now, the answer to RQ1 can be given as follows:

AlexNet has a high error rate.

MobileNet is slow in inferring results.

ResNet50 is an optimized choice in terms of accuracy and speed for detecting face mask using transfer learning.

4.3 Performance Analysis of Proposed Model

The performance of the proposed model using ResNet50 is further evaluated using various metrics such as Accuracy, Intersection over Union (IoU) and AU-ROC using Eqs. (14)–(17), respectively.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{14}$$

$$IoU = \frac{TP}{(TP + FP + FN)} \tag{15}$$

$$Sensitivity = \frac{TP}{(TP + FN)} \tag{16}$$

$$\text{Specificity} = \frac{\text{TN}}{(\text{TN} + \text{FP})} \tag{17}$$

where TP, FP, TN and FN represent True Positive, False Positive, True Negative and False Negative, respectively. TP, FP, TN and FN are obtained through a confusion matrix.

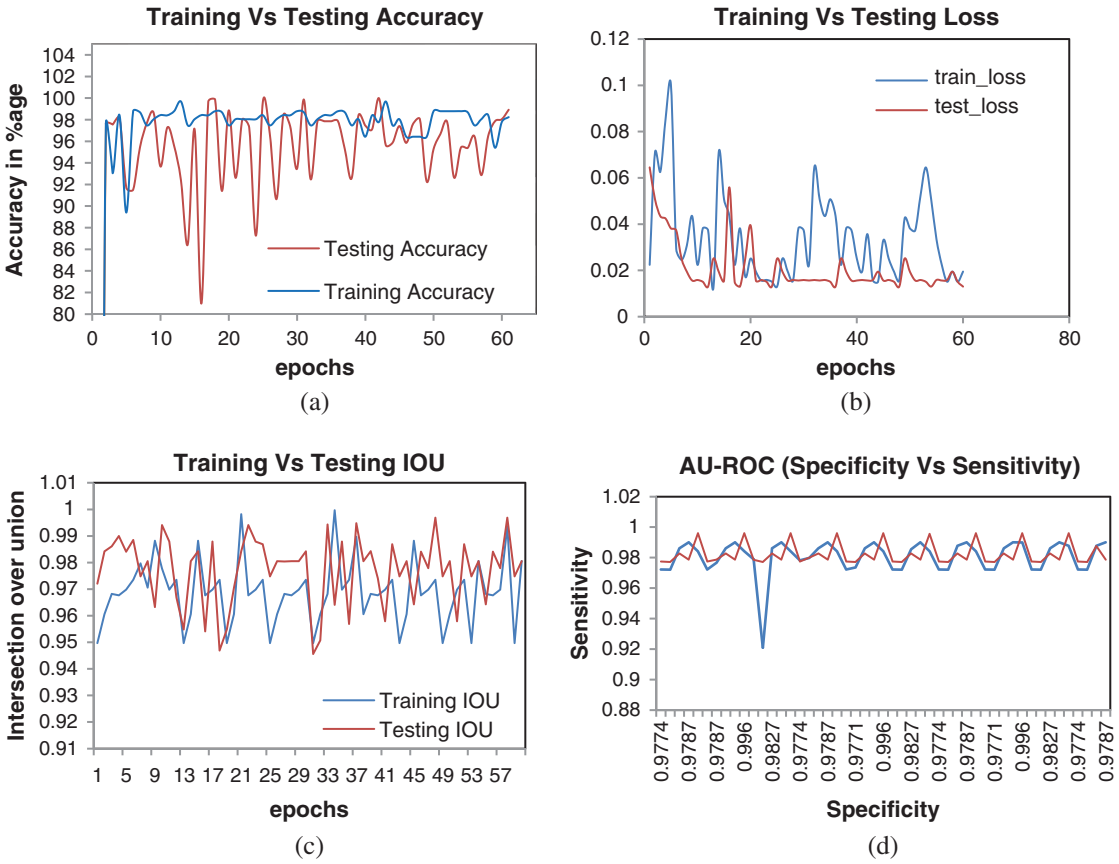

Figure 7: Performance analysis of proposed model during training and testing phase

AU-ROC represents a degree of measure of separability, i.e., it tells us how much proposed model is capable if differentiating between classes. The higher the area under the ROC curve, the better the model is in differentiating between masked faces and non-masked faces. The curve is plotted with Sensitivity against Specificity.

Furthermore, to address RQ3 and avoid the problem of overfitting, two major steps are taken. First, we performed data augmentation as discussed in Section 3.1.2. Second, the model accuracy is critically observed over 60 epoch cycles. It is observed that model accuracy keeps on increasing in different epochs and get stable after epoch = 3 as depicted graphically in Fig. 7a above for fine-tuned unbiased dataset.

4.4 Comparison with Existing Models

In this Section, we aim to compare the performance of the proposed model with public baseline results published in RetinaFaceMask [47] which aims to answer RQ2. Since RetinaFaceMask is trained on MAFA dataset and performance is evaluated using precision and recall for face and mask detection so, for comparison purpose, the performance of the proposed technique is also evaluated in the same environment. The experimental results are reported in Tab. 4. It may be noted from Tab. 4 that the proposed model with ResNet50 as backbone achieves higher accuracy as compared to RetinaFaceMask.

Table 4: Comparison of proposed model with recent face mask detection model on MAFA dataset

Model	Face detection		Mask detection	
	Precision (%)	Recall (%)	Precision (%)	Recall (%)
RetinaFaceMask based on MobileNet [13]	83.0	95.6	82.3	89.1
RetinaFaceMask based on ResNet [13]	91.9	96.3	93.4	94.5
Proposed model based on ResNet50	99.2	99.0	98.92	98.24

Particularly, the proposed model generates 11.75% and 11.07% higher precision in face and mask detection respectively when compared with RetinaFaceMask. The recall is improved by 3.05% and 6.44% in the face and mask detection respectively. We had observed that improved results are possible due to the optimized face detector discussed in Section 3.3 for dealing with complex images.

Besides, the performance of the proposed technique is also compared with a conventional technique such as SVM [42] under a similar experimental setup. It is also observed that the training dataset is difficult to handle by SVM in terms of computational power and memory consumption. So, the experiment is setup with a small subset of training data to analyze the performance of SVM over 9199 images. Even with this small subset, the memory requirement to build a classification model using SVM is 650 MB. Whereas, the proposed model requires only 24.73 MB over the same subset and thus a gain of 96% is achieved in terms of memory which is perfectly suitable for an embedded device used for surveillance purposes. The experimental results are summarized in Tab. 5.

It may be noted from Tab. 5 that a gain of 9.33% is achieved in precision using proposed technique. A gain of 3.85% is achieved in the recall. The reason for lesser accuracy through SVM is also reported. It is found that SVM under performs in the scenario where the number of features per data point exceeds the number of training data samples. The results obtained in

Tabs. 4 and 5 reveal that the proposed technique achieves higher accuracy for facemask detection in real time with less memory consumption and less effort by the amalgam of transfer learning with an ensemble of one stage and two stage image complexity predictor when compared to recent and conventional methods.

Table 5: Comparison of proposed model with conventional face mask classification model on MAFA dataset

Model	Mask detection		
	Precision (%)	Recall (%)	Accuracy (%)
SVM [52]	89.59	94.39	89.93
Proposed technique	98.92	98.24	98.2

4.5 Discussion

The proposed model achieves high accuracy in the face and mask detection with less inference time and less memory consumption as compared to Support Vector Machine [42] and RetinaFaceMask [47]. Significant efforts had been put to resolve the data imbalance problem in the existing MAFA dataset, resulting in new dataset which is highly suitable for COVID related mask detection tasks. The newly created dataset, feature engineering through transfer learning of robust ResNet50 pre-trained model, optimal face detection approach with improved localization using affine transformation and avoidance of overfitting resulted in an overall system that can be easily installed in thermal cameras at public places to curtail spread of Coronavirus.

5 Conclusion and Future Scope

In this work, a deep learning-based approach for detecting masks over faces in a public place to curtail community spread of Coronavirus is presented. The proposed technique efficiently handles varying kinds of occlusions in the dense situation by making use of an ensemble of single and two stage detectors at the pre-processing level. The ensemble approach not only helps in achieving high accuracy but also improves detection speed considerably. Furthermore, the application of transfer learning on pre-trained models with extensive experimentation over unbiased dataset resulted in a highly robust and low-cost system.

Finally, the work opens interesting future directions for researchers. Firstly, the proposed technique can be integrated into any high-resolution video surveillance devices and not limited to mask detection only. Secondly, the model can be trained and upgraded to mask datasets that include different images related to correctly/incorrectly wear mask [53] and achieve the ultimate purpose of detecting facemask for cutting down the risk of contagious diseases.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. World Health Organization (2020). Coronavirus disease 2019 (COVID-19): Situation report, 96. https://www.who.int/docs/default-source/coronaviruse/situation-reports/20200816-covid-19-sitrep-209.pdf?sfvrsn=5dde1ca2_2.
2. Pan American Health Organization (2020). Social distancing, surveillance, and stronger health systems as keys to controlling COVID-19 pandemic. PAHO Director says-PAHO/WHO | Pan American Health Organization. https://www.paho.org/en/news/2-6-2020-social-distancing-surveillance-and-stronger-health-systems-keys-controlling-covid-19.
3. Garcia Godoy, L. R., Jones, A. E., Anderson, T. N., Fisher, C. L., Seeley, K. M. L. et al. (2020). Facial protection for healthcare workers during pandemics: A scoping review. *BMJ Global Health, 5(5),* e002553. DOI 10.1136/bmjgh-2020-002553.
4. Eikenberry, S. E., Mancuso, M., Iboi, E., Phan, T., Eikenberry, K. et al. (2020). To mask or not to mask: Modeling the potential for face mask use by the general public to curtail the COVID-19 pandemic. *Infectious Disease Modelling, 5(8),* 293–308. DOI 10.1016/j.idm.2020.04.001.
5. University of Maryland (2020). Wearing surgical masks in public could help slow COVID-19 pandemic's advance: Masks may limit the spread diseases including influenza, rhinoviruses and coronaviruses. *ScienceDaily*. https://www.sciencedaily.com/releases/2020/04/200403132345.htm.
6. Hjelmås, E., Low, B. K. (2001). Face detection: A survey. *Computer Vision and Image Understanding, 83(3),* 236–274. DOI 10.1006/cviu.2001.0921.
7. Nanni, L., Ghidoni, S., Brahnam, S. (2017). Handcrafted *vs.* non-handcrafted features for computer vision classification. *Pattern Recognition, 71(10),* 158–172. DOI 10.1016/j.patcog.2017.05.025.
8. Al-Allaf, O. N. A. (2014). Review of face detection systems based artificial neural networks algorithms. *International Journal of Multimedia & Its Applications, 6(1),* 1–16. DOI 10.5121/ijma.2014.6101.
9. Yang, S., Luo, P., Loy, C. C., Tang, X. (2016). Wider face: A face detection benchmark. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 5525–5533. Las Vegas, NV, USA.
10. Ge, S., Li, J., Ye, Q., Luo, Z. (2017). Detecting masked faces in the wild with LLE-CNNs. *Proceedings-30th IEEE Conference on Computer Vision and Pattern Recognition*, pp. 426–434. Janua, Honolulu, HI, USA.
11. Ulhaq, A., Khan, A., Gomes, D., Paul, M. (2020). Computer vision for COVID-19 control: A survey, pp. 1–24. DOI 10.31224/osf.io/yt9sx.
12. Girshick, R., Donahue, J., Darrell, T., Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 580–587. Columbus, OH, USA.
13. Roy, B., Nandy, S., Ghosh, D., Dutta, D., Biswas, P. et al. (2020). MOXA: A deep learning based unmanned approach for real-time monitoring of people wearing medical masks. *Transactions of the Indian National Academy of Engineering, 5(3),* 509–518. DOI 10.1007/s41403-020-00157-z.
14. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R. et al. (2013). OverFeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv: 1312.6229.
15. Erhan, D., Szegedy, C., Toshev, A., Anguelov, D. (2014). Scalable object detection using deep neural networks. *IEEE Conference on Computer Vision and Pattern Recognition.*
16. Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 779–788. Las Vegas, NV, USA.
17. Chun, L. Z., Dian, L., Zhi, J. Y., Jing, W., Zhang, C. (2020). YOLOv3: Face detection in complex environments. *International Journal of Computational Intelligence Systems, 13(1),* 1153–1160. DOI 10.2991/ijcis.d.200805.002.
18. https://towardsdatascience.com/face-mask-detection-using-yolov5-3734ca0d60d8.
19. Addagarla, S. K., Chakravarthi, G. K., Anitha, P. (2020). Real time multi-scale facial mask detection and classification using deep transfer learning techniques. *International Journal, 9(4),* 4402–4408. DOI 10.30534/ijatcse/2020/33942020.

20. Huang, R., Pedoeem, J., Chen, C. (2018). YOLO-LITE: A real-time object detection algorithm optimized for non-GPU computers. *IEEE International Conference on Big Data*, pp. 2503–2510. Seattle, WA, USA. IEEE.
21. Yadav, S. (2020). Deep learning based safe social distancing and face mask detection in public areas for COVID-19 safety guidelines adherence. *International Journal for Research in Applied Science and Engineering Technology, 8(7),* 1368–1375. DOI 10.22214/ijraset.2020.30560.
22. https://www.pyimagesearch.com/2020/05/04/covid-19-face-mask-detector-with-opencv-keras-tensorflow-and-deep-learning/.
23. Girshick, R., Donahue, J., Darrell, T., Malik, J. (2016). Region-based convolutional networks for accurate object detection and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38(1),* 142–158.
24. Wu, W., Yin, Y., Wang, X., Xu, D. (2019). Face detection with different scales based on faster R-CNN. *IEEE Transactions on Cybernetics, 49(11),* 4017–4028. DOI 10.1109/TCYB.2018.2859482.
25. He, K., Zhang, X., Ren, S., Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 37(9),* 1904–1916. DOI 10.1109/TPAMI.2015.2389824.
26. Girshick, R. (2015). Fast R-CNN. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448. Santiago, Chile.
27. Nguyen, N. D., Do, T., Ngo, T. D., Le, D. D. (2020). An evaluation of deep learning methods for small object detection. *Journal of Electrical and Computer Engineering,* 1–18. DOI 10.1155/2020/3189691.
28. Jiang, H., Learned-Miller, E. (2017). Face detection with the faster R-CNN. *12th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2017-1st International Workshop on Adaptive Shot Learning for Gesture Understanding and Production, Biometrics in the Wild, Heteroge*, pp. 650–657. Washington DC, USA.
29. Sun, X., Wu, P., Hoi, S. C. H. (2017). Face detection using deep learning: An improved faster RCNN approach. *Neurocomputing, 299(2),* 42–50. DOI 10.1016/j.neucom.2018.03.030.
30. Dai, J., Li, Y., He, K., Sun, J. (2016). R-FCN: Object detection via region-based fully convolutional networks. *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 379–387.
31. Wang, Y., Ji, X., Zhou, Z., Wang, H., Li, Z. (2017). Detecting faces using region-based fully convolutional networks. http://arxiv.org/abs/1709.05256.
32. Liang, Z., Shao, J., Zhang, D., Gao, L. (2018). Small object detection using deep feature pyramid networks. *Pacific Rim Conference on Multimedia*, pp. 554–564. Cham: Springer.
33. He, K., Gkioxari, G., Dollar, P., Girshick, R. (2017). Mask R-CNN. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2980–2988. Venice, Italy.
34. Lin, K. H., Zhao, H. M., Lv, J. J., Li, C. Y., Liu, X. Y. et al. (2020). Face detection and segmentation based on improved mask R-CNN. *Discrete Dynamics in Nature and Society, 2020,* 1–11. DOI 10.1155/2020/9242917.
35. Qin, B., Li, D. (2020). Identifying facemask-wearing condition using image super-resolution with classification network to prevent COVID-19. *Sensors, 20(18),* 5236. DOI 10.21203/rs.3.rs-28668/v1.
36. Soviany, P., Ionescu, R. T. (2018). Optimizing the trade-off between single-stage and two-stage deep object detectors using image difficulty prediction. *2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, pp. 209–214. Timisoara, Romania.
37. Lin, T. Y., Goyal, P., Girshick, R., He, K., Dollar, P. (2020). Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, DOI 10.1109/TPAMI.2018.2858826.
38. Cai, Z., Fan, Q., Feris, R. S., Vasconcelos, N. (2016). A unified multi-scale deep convolutional neural network for fast object detection. *European Conference on Computer Vision*, pp. 354–370. Cham: Springer.
39. Fu, C. Y., Liu, W., Ranga, A., Tyagi, A., Berg, A. C. (2017). DSSD: Deconvolutional single shot detector. http://arxiv.org/abs/1701.06659.
40. Shrivastava, A., Sukthankar, R., Malik, J., Gupta, A. (2016). Beyond skip connections: Top-down modulation for object detection. http://arxiv.org/abs/1612.06851.

41. Dvornik, N., Shmelkov, K., Mairal, J., Schmid, C. (2017). BlitzNet: A real-time deep network for scene understanding. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4174–4182. Venice, Italy.
42. Loey, M., Manogaran, G., Taha, M. H. N., Khalifa, N. E. M. (2020). A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic. *Measurement, 167(5),* 108288. DOI 10.1016/j.measurement.2020.108288.
43. Deng, J., Dong, W., Socher, R., Li, L. J., Li, K. et al. (2010). ImageNet: A large-scale hierarchical image database. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. Miami, FL, USA.
44. Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P. et al. (2014). Microsoft COCO: Common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (Eds.), *European Conference on Computer Vision,* vol. 8693, pp. 740–755. Zurich, Switzerland.
45. Apostolopoulos, I. D., Mpesiana, T. A. (2020). COVID-19: Automatic detection from X-ray images utilizing transfer learning with convolutional neural networks. *Physical and Engineering Sciences in Medicine, 43,* 635–640. DOI 10.1007/s13246-020-00865-4.
46. Jignesh, C. G., Punn, N. S., Sonbhadra, S. K., Agarwal, S. (2020). Face mask detection using transfer learning of inceptionV3. *Lecture Notes in Computer Science,* vol. 12581. Sonepat, India, Cham: Springer.
47. Jiang, M., Fan, X., Yan, H. (2020). RetinaMask: A face mask detector. http://arxiv.org/abs/2005.03950.
48. Jain, V., Learned-Miller, E. (2010). Fddb: A benchmark for face detection in unconstrained settings. *UMass Amherst Technical Report*.
49. Yang, B., Yan, J., Lei, Z., Li, S. Z. (2015). Fine-grained evaluation on face detection in the wild. *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition*, pp. 1–7. Ljubljana, Slovenia.
50. Liu, Z., Luo, P., Wang, X., Tang, X. (2018). Large-scale celebfaces attributes (celeba) dataset. http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html.
51. Ge, S., Li, J., Ye, Q., Luonn, Z. (2017). MAFA (MAsked FAces)-Datasets-CKAN. http://221.228.208.41/gl/dataset/0b33a2ece1f549b18c7ff725fb50c561.
52. Wang, Z., Wang, G., Huang, B., Xiong, Z., Hong, Q. et al. (2020). Masked face recognition dataset and application. http://arxiv.org/abs/2003.09093.
53. Cabani, A., Hammoudi, K., Benhabiles, H., Melkemi, M. (2020). MaskedFace-Net-A dataset of correctly/incorrectly masked face images in the context of COVID-19. *Smart Health, 19,* 100144. DOI 10.1016/j.smhl.2020.100144.
54. Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y. et al. (2019). MMDetection: Open MMLab detection toolbox and benchmark. http://arxiv.org/abs/1906.07155.
55. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J. et al. (2014). Caffe: Convolutional architecture for fast feature embedding. *Proceedings of the 2014 ACM Conference on Multimedia*, Orlando, Florida, USA.
56. Ionescu, R. T., Alexe, B., Leordeanu, M., Popescu, M., Papadopoulos, D. P. et al. (2017). How hard can it be? Estimating the difficulty of visual search in an image. *IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA.
57. Qiao, S., Liu, C., Shen, W., Yuille, A. (2018). Few-shot image recognition by predicting parameters from activations. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 7229–7238. Salt Lake City, UT, USA.
58. Bandyopadhyay, H. (2020). The right loss function [PyTorch]. *Heartbeat*. https://heartbeat.fritz.ai/the-right-loss-function-PyTorch-58d2c0d77404.