Abstract

Effective strategies to restrain 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 a mask is among the non-pharmaceutical intervention measures that can be used to cut the primary source of SARS-CoV2 droplets expelled by an infected individual. Regardless of discourse on medical resources and diversities in masks, all countries are mandating coverings over the nose and mouth in public. To contribute towards communal health, this paper aims to devise a highly accurate and real-time technique that can efficiently detect non-mask faces in public and thus, enforcing to wear mask. The proposed technique is ensemble of one-stage and two-stage detectors to achieve low inference time and high accuracy. We start with ResNet50 as a baseline 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 experiment is conducted with three popular baseline models viz. 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 achieves high accuracy (98.2%) when implemented with ResNet50. Besides, the proposed model generates 11.07% and 6.44% higher precision and recall in mask detection when compared to the recent public baseline model published as RetinaFaceMask detector. The outstanding performance of the proposed model is highly suitable for video surveillance devices.

Chapter 1: Introduction

1. Introduction

The 209th report of the world health organization (WHO) published on 16th August 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 the researchers at the University of Edinburgh reveals that wearing a face mask or other covering over the nose and mouth cuts the risk of Coronavirus spread by avoiding forward distance travelled by a 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. The findings reveal that near universal adoption (80%) of even weak masks (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 the use of the face masks in general public to curtail the spread of Coronavirus. Further, with the reopening of countries from COVID-19 lockdown, Government and Public health agencies are recommending face mask as essential measures to keep us safe when venturing into public. To mandate the use of facemask, it becomes essential to devise some technique that enforce individuals to apply a mask before exposure to public places.

Face mask detection refers to detect whether a person is wearing a mask or not. In fact, the problem is reverse engineering of face detection where the face is detected using different machine learning algorithms for the purpose of security, authentication and surveillance. Face detection is a key area in the field of Computer Vision and Pattern Recognition. A significant body of research has contributed sophisticated to algorithms for face detection in past. The primary research on face detection was done in 2001 using the design of handcraft feature and application of traditional machine learning algorithms to train effective classifiers for detection and recognition [6], [7]. The problems encountered with this approach include high complexity in feature design and low detection accuracy. In recent years, face detection methods based on deep convolutional neural networks (CNN) have been widely developed [8], [9], [10], [11] to improve detection performance.

Although numerous researchers have committed efforts in designing efficient algorithms for face detection and recognition but there exists an essential difference between 'detection of the face under mask' and 'detection of mask over face'. As per available literature, very little body of research is attempted to detect mask over face. Thus, our work aims to a develop technique that can accurately detect mask over the face in public areas (such as airports. railway stations, crowded markets, bus stops, etc.) to curtail the spread of Coronavirus and thereby contributing to public healthcare. Further, it is not easy to detect faces with/without a mask in public as the dataset available for detecting masks on human faces is relatively small leading to the hard training of the model. So, the 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 face images including faces with masks, faces without masks, faces with and without masks in one image and confusing images without masks. With an extensive dataset containing 45,000 images, our technique achieves outstanding accuracy of 98.2%. The major contribution of the proposed work is given below:

1. Develop a novel object detection method that combines one-stage and two-stage detectors for accurately detecting the object in real-time from video streams with transfer learning at the back end.

2. Improved affine transformation is developed to crop the facial areas from uncontrolled real-time images having differences in face size, orientation and background. This step helps in better localizing the person who is violating the facemask norms in public areas/ offices.

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 purposes.

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 the performance of the proposed technique with various pre-trained models over different parameters of speed and accuracy. Finally, Section 5 concludes the work with possible future directions.

Chapter 2: Related Work

2. Related work

Pattern learning and object recognition are the inherent tasks that a computer vision (CV) technique must deal with. Object recognition encompasses both image classification and object detection [12]. The task of recognizing the mask over the face in the pubic area can be achieved by deploying an efficient object recognition algorithm through surveillance devices. The object recognition pipeline consists of generating the region proposals followed by classification of each proposal into related class [13]. We review the recent development in region proposal techniques using single-stage and two-stage detectors, general technique for improving detection of region proposals and pre-trained models based on these techniques.

2.1 Single-stage detectors

The single-stage detectors treat the detection of region proposals as a simple regression problem by taking the input image and learning the class probabilities and bounding box coordinates. OverFeat [8] and DeepMultiBox [9] 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 when compared with two-stage detectors; especially when small objects are taken into consideration [10]. Basically, the YOLO network divides an image into a grid of size GxG, and each grid generates N predictions for bounding boxes. Each bounding box is limited to have only one class during the prediction, which restricts the network from finding smaller objects. Further, YOLO network was improved to YOLOv2 that included batch normalization, high-resolution classifier and anchor boxes. Furthermore, the development of YOLOv3 is built upon YOLOv2 with the addition of an improved backbone classifier, multi-sale prediction and a new network for feature extraction. Although, YOLOv3 is executed faster than Single-Shot Detector (SSD) but does not perform well in terms of classification accuracy [14], [15]. Moreover, YOLOv3 requires a large amount of computational power for inference, making it not suitable for embedded or mobile devices. Next, SSD networks have superior performance than YOLO due to small convolutional filters, multiple feature maps and prediction in multiple scales. The key difference between the two architectures is that YOLO utilizes two fully connected layers, whereas the SSD network uses convolutional layers of varying sizes. Besides, the RetinaNet [11] proposed by Lin is also a single-stage object detector that uses featured image pyramid and focal loss to detect the

dense objects in the image across multiple layers and achieves remarkable accuracy as well as speed comparable to two-stage detectors.

2.2 Two-stage detectors

In contrast to single-stage detectors, two-stage detectors follow a long line of reasoning in computer vision for the 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. Region-based convolutional neural network also abbreviated as R-CNN [16] described in 2014 by Ross Girshick et al. It may have been one of the first large-scale applications of CNN to the problem of object localization and recognition. The model was successfully demonstrated on benchmark datasets such as VOC-2012 and ILSVRC-2013 and produced state of art results. Basically, R-CNN applies a selective search algorithm to extract a set of object proposals at an initial stage and applies SVM (Support Vector Machine) classifier for predicting objects and related classes at later stage. Spatial pyramid pooling SPPNet [17] (modifies R-CNN with an SPP layer) collects features from various region proposals and fed into a fully connected layer for classification. The capability of SPNN to compute feature maps of the entire image in a single-shot resulted in significant improvement in object detection speed by the magnitude of nearly 20 folds greater than R-CNN. Next, Fast R-CNN is an extension over R-CNN and SPPNet [18], [12]. 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 [19].

Further, Faster R-CNN is an 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 the object detection system in a single step [20], [21]. Although this integration leads to the accomplishment of break-through for the speed bottleneck of Fast R-CNN but there exists a computation redundancy at the subsequent detection stage. The Region-based Fully Convolutional Network (R-FCN) is the only model that allows complete backpropagation for training and inference [22], [23]. Feature Pyramid Networks (FPN) can detect non-uniform objects, but least used by researchers due to high computation cost and more memory usage [24]. Furthermore, Mask R-CNN

strengthens Faster R-CNN by including the prediction of segmented masks on each RoI [25]. Although two-stage yields high object detection accuracy, but it is limited by low inference speed in real-time for video surveillance [14].

2.3. Techniques for improving detectors

Several techniques for improving the performance of single-stage and two-stage detectors have been proposed in past [26]. 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 [27]. Modification in context information is another approach used to improve detection accuracy or speed. MS-CNN [20], DSSD [21] and TDN [22] strengthen the feature representation by enriching the context of coarser features by including an additional layer in a top-down manner for better object detection. BlitzNet improved SSD by adding semantic segmentation layer to achieve high detection accuracy [27]. The object detection architectures discussed so far have several open-source models which are pre-trained on large datasets like ImageNet [28], COCO [29] and ILSVRC [30]. 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. 1 summarizes various pre-trained models based on CNN architectures commenced from 2012 to 2018. 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. 1.

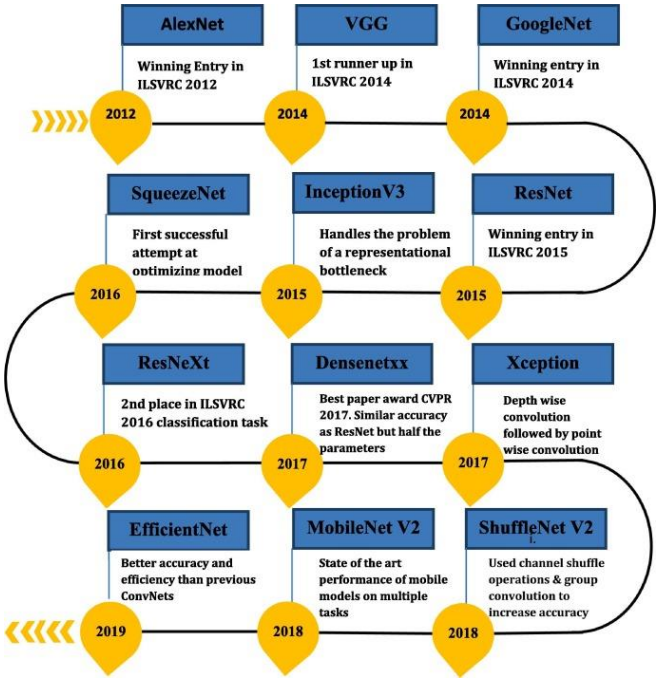

Fig.1 Various Pre-trained Models based on CNN Architectures.

To enforce mask over faces in public areas to curtail community spread of Coronavirus, a machine learning approach based on the available pre-trained model is highly recommended for the welfare of the society. These pre-trained models are required to be finely tuned with benchmark datasets. The number of datasets with diverse features pertaining to human faces with and without mask are given in Table 1 .

Table 1

Different Categories of Datasets.

Type of Datasets	Dataset	Scale	#Faces	#masked face images	Occlusion
Masked face detection Datasets	FDDB [31]	2845	5171	–	–
	MALF [32]	5250	11931	–	✓
	calebA [33]	200000	202599	–	–
	WIDERFACE [34]	32203	194000	–	✓
Face masked datasets	MAFA [35]	30811	37824	35806	✓
	RMFRD [36]	95000	9200	5000	✓
	SMFRD [36]	85000	5000	5000	✓
	MFDD [36]	500000	500000	24771	✓

An extensive study conducted on available face-related datasets reveal that there exist principally two kinds of datasets. These are: i) masked face and ii) face masked datasets. The masked face datasets are more concentrated on including the face images with a variant degree of facial expression and landmarks whereas face mask centric datasets include those images of faces that are mainly

characterized by occlusions and their positional coordinates near the nose and mouth area. Table 1 summarizes these two kinds of prevalent datasets. 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 a few models are currently capable of handling COVID related face masked datasets

2. The available face masked datasets are scarce and need to strengthen with varying degrees 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. But none of them truly meets the requirement of real-time video surveillance devices. These devices are limited by less computational power and memory [37]. So, they require optimized object detection models that can perform surveillance in real-time with less memory consumption and without a notable reduction in accuracy. Single-stage detectors are good for real-time surveillance but limited by low accuracy, whereas two-stage detectors can easily produce accurate results for complex inputs but at the cost of computational time. All these factors necessitate to develop an integrated model for surveillance devices which can produce benefits in terms of computational time as well as accuracy.

To solve these problems, a deep-learning model based on transfer learning which is trained on a highly tuned customized face mask dataset and compatible with video surveillance is being proposed and discussed in detail in the next section.

Chapter 3: Proposed Architecture

The proposed model is based on the object recognition benchmark given in [38]. According to this benchmark, all the tasks related to an object recognition problem can be ensembled under three main components: Backbone, Neck and Head as depicted in Fig. 2 . Here, the backbone corresponds to a baseline convolutional neural network capable of extracting information from images and converting them to a feature map. In the proposed architecture, the concept of transfer learning is applied on the backbone to utilize already learned attributes of a powerful pre-trained convolutional neural network in extracting new features for the model.

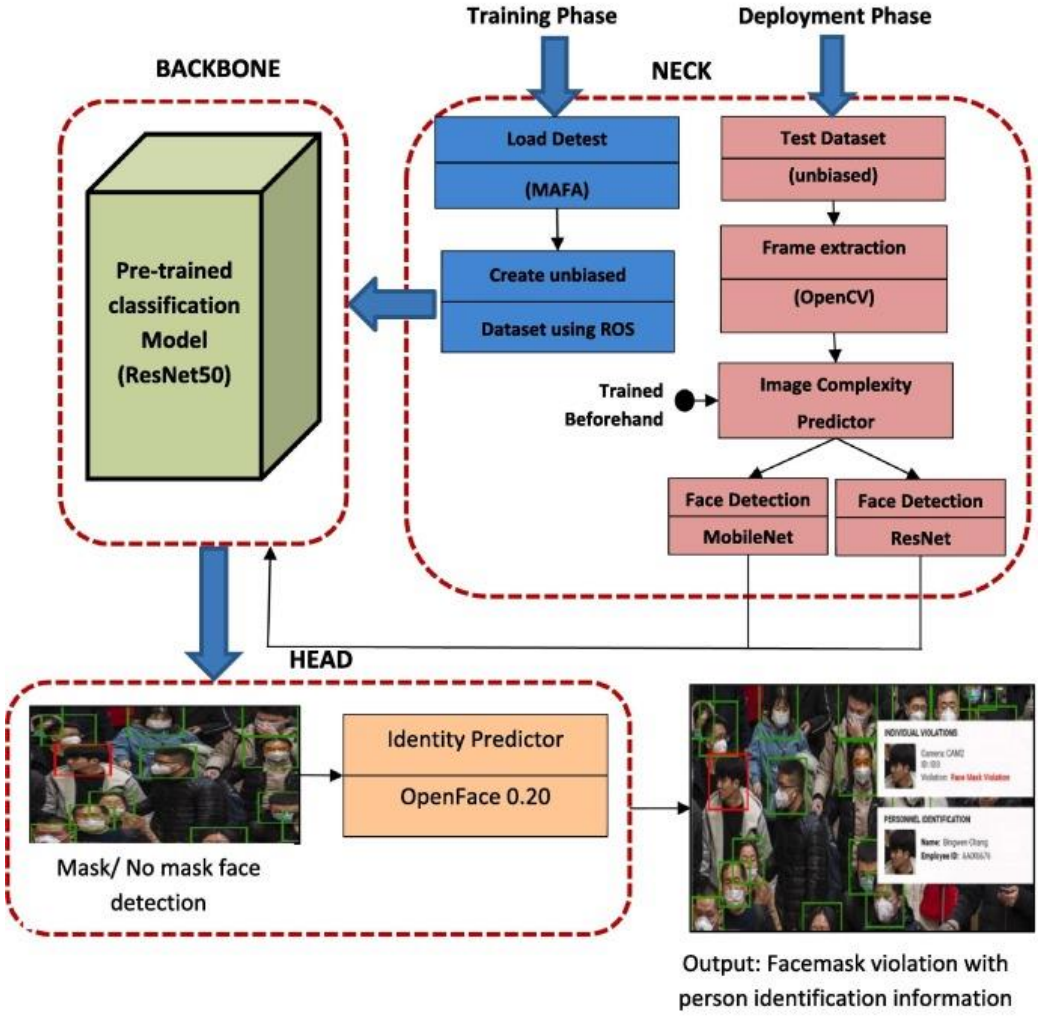

Fig.2 Proposed Architecture

An exhaustive backbone building strategy with three popular pre-trained models namely ResNet50, MobileNet and AlexNet are conducted for obtaining the best results for facemask detection. The ResNet50 is found to be optimized choice for building the backbone (Refer Section 4.2) of the proposed model. The novelty of our work is being proposed in the Neck component. The intermediate component, the Neck contains all those pre-processing tasks that are needed before the actual classification of images. To make our model compatible with surveillance devices, Neck applies different pipelines for the training and deployment phase. The training pipeline follows the creation of an unbiased customized dataset and fine-tuning of ResNet50. The deployment pipeline consists of real-time frame extraction from video followed by face detection and extraction. In order to achieve trade-off between face detection accuracy and computational time, we propose an image complexity predictor (Refer. Section 3.3). The last component, Head stands for identity detector or predictor that can achieve the desired objective of deep-learning neural network. In the proposed architecture, the trained facemask classifier obtained after transfer learning is applied to detect mask and no mask faces. The ultimate objective of enforcement of wearing of face mask in public area will only be achieved after retrieving the personal identification of faces, violating the mask norms. The action can further, be taken as per government/ office policy. Since there may exist differences in face size and orientation in cropped ROI, affine transformation is applied to identify facial using OpenFace 0.20 [38], [39]. The detailed description of each task in the proposed architecture is given in the following subsections.

3.1. Creation of unbiased facemask dataset

A facemask-centric dataset, MAFA [35] with a total of 25,876 images, categorized over two classes namely masked and non-masked was initially considered. The number of masked images in MAFA are 23,858 whereas non-masked images are only 2018. 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 analyze the performance of the image classifier once with the original MAFA set (biased) and then with the proposed dataset (unbiased).

3.1.1. Supervised pre-training

We discriminatively pre-trained the CNN on the original biased MAFA dataset. The pre-training was performed using the open-source Caffe python library [7]. In short, our CNN model nearly matches the performance of Madhura et al. [11], obtaining a top-1 error rate 1.8% higher on MAFA validation set. This discrepancy may cause due to 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 $\boldsymbol{\rho}$ = 11.82 (original) to $\boldsymbol{\rho}$ = 1.07. The formula used for computing the imbalance ratio is given by equation (1).

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

Here, *D* refers to image Dataset, *majority (Di)* and *minority (Di)* return the majority and minority class of *D*. Count(X) returns the number of images in any arbitrary class x. After data balancing, stochastic gradient descent (SGD) training of CNN parameters with a learning rate of 0.003 is set over wrapped region proposals. The low learning rate allows fine-tuning of the 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 a reduction in the top-1 error rate of 3.7%.

3.2. Fine-tuning of pre-trained model

In the proposed work, facemask detection is achieved through deep neural networks because of their better performance than other classification algorithms. But training a deep neural network is expensive because it is a time-consuming task and requires high computational power. To train the network faster and cost-effective, deep-learning-based transfer learning is applied here. Transfer learning allows to transferring of the trained knowledge of the neural network in terms of parametric weights to the new model. It boosts the performance of the new model even when it is trained on a small dataset. There are several pre-trained models like AlexNet, MobileNet, ResNet50 etc. that had been trained with 14 million images from the ImageNet dataset [40]. In the proposed model, ResNet50 is chosen as a pre-trained model for facemask classification. The last layer of ResNet50 is fine-tuned by adding five new layers. The newly added layers include an average pooling layer of pool size equal to 5 × 5, a flattering layer, a dense ReLU layer of 128 neurons, a dropout of 0.5 and a decisive layer with softmax activation function for binary classification as shown in Fig. 3 .

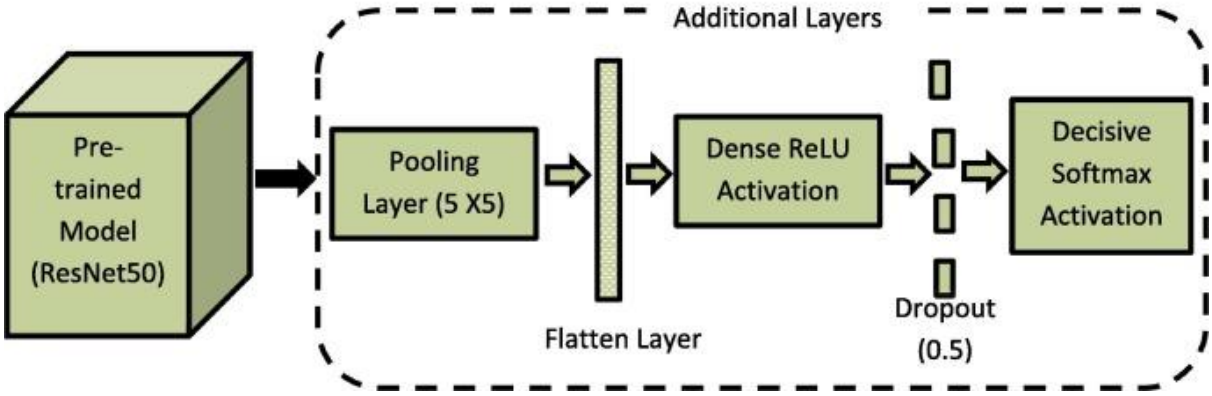

Fig.3. Fine-tuning

3.3. Image complexity predictor for face detection

To address problem 3 identified in Section 2, various face images are analyzed in terms of processing complexity. It is observed that dataset, we consider primarily, contains two major classes that is, mask and non-mask class but the mask class further, contains an inherent variety of occlusions other than surgical/cloth facemask, for example, occlusion of ROI by other objects like a person, hand, hair or some food item as shown 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 face detection is not a trivial task. So, an image complexity predictor is being proposed here. Its purpose is to split data into soft versus hard images at the initial level followed by a mask and non-mask classification at a later level through a facemask classifier. The important question that we need to answer is how to determine whether an image is soft or hard. The answer to this question is given by the "Semi-supervised object classification strategy" proposed by Lonescu et al. [41]. The Semi-supervised object classification strategy is suitable for our task because it predicts objects without localizing them. For implementing this strategy, we took three sets of image samples: the first set (L) contains labelled (hard/soft) training images, the second set (U) contains unlabelled training images and the third set (T) contains unlabelled test images. We further, applied the curriculum learning approach as suggested in [26], which operates iteratively, by training the hard/soft predictor at each iteration on enlarged training set L. The training set L is enlarged by randomly moving k samples from U to L. We stopped the learning process when L grew three times its original size. Initially, 500 labelled samples are populated in L. The initial labelling of samples in L is done using the three most correlated image properties that make the image complex. These properties are namely object density (including full, truncated and occluded faces), mean area covered by object normalized by image size and image resolution. The object density is

evaluated by human annotators. We took 50 trusted annotators and each annotator is shown 10 images.

Fig.4. Variety of Occlusions Present in Dataset

We asked two questions to each annotator. The questions were: "Is there a human being in the image?" and "How many human faces including full, truncated and occluded faces are present in the given image?". We ensured the annotation task is not trivial by presenting images in a random order such that if the answer to one image is positive then for another image, it may be negative. We recorded the response time of each annotator to answer the questions. We removed all response time longer than 30 s to avoid bias. Further, each annotator response time is normalized by subtracting it from meantime and dividing by standard deviation. We computed the geometric mean of all response times per image and saved the values as object density. We further, observed that image complexity is positively related to object density whereas negatively related to object size and image resolution. Based on these image properties, a ground truth visibility difficulty score is assigned to each image.

To automatically predict the hardness of images in T, we further used VGG-f with ν- support vector regression as discussed in [26]. The last layer of VCG-f is replaced by a fully connected layer. Each test image is divided into three bins of 1 × 1, 2 × 2 and 3 × 3 size, to get pyramid representation of the image for better performance. The image is also flipped horizontally and the same pyramid is applied over it. The 4096 features extracted from each bin are combined to obtain a single feature vector followed by normalization using L2-norm. The obtained normalized feature vector is further, used to regress the image complexity score. Thus, the model automatically predicts image complexity for each image in T. Having identified the hardness of the test images using an image complexity predictor, a soft image is proposed to process through a fast single-stage detector while the 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 predicting hard images. The algorithm for image complexity predictor is outlined below:

Algorithm

Image_Complexity_Predictor ()

1. Input:
2. Image ← input image
3. Dfast ← single-stage detector
4. Dslow ← two-stage detector
5. C ← Image complexity
6. Computation:
7. If (C = Soft)
    - R ← D_{slow}(Image)
8. else
    - R ← D_{fast}(Image)
9. Output:
10. R ← set of region proposals

Table 2 summarizes mAP score and Computation time for various combinations of MobileNet and ResNet50 over 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 (0–100%). Here, the test data is partitioned based on random split or soft versus hard spilt given by Image Complexity Predictor. To reduce the bias, the average mAp over 5 runs is recorded for random spilt. The elapsed time is measured on Inter I7, 2.5 GHZ CPU with 16 GB RAM.

Table 2

Comparison between MobileNet-SSD, ResNet50 and Their Various Combinations 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 1.92	0.05 3.08	0.05 5.07	– 6.02
Mask detection time (ms)					
Total Computation Time (ms)	0.05	1.97	3.13	5.12	6.02

3.4. Identity prediction

After detecting faces with masks and non-mask in the search proposal, the non-mask faces are passed separately into a neural network for further exploration of a person's identity for being violating the facemask norm. The step requires a fixed-sized input. One possible way of getting a fixed-size input is to reshape the face in the bounding box to 96 × 96 pixels. The potential issue with this solution is that the face could be looking in a different direction. Affine transformation can handle this issue very easily [42]. The technique is similar to deformable part models described in [43]. The use of affine transformation is depicted in Fig. 5 .

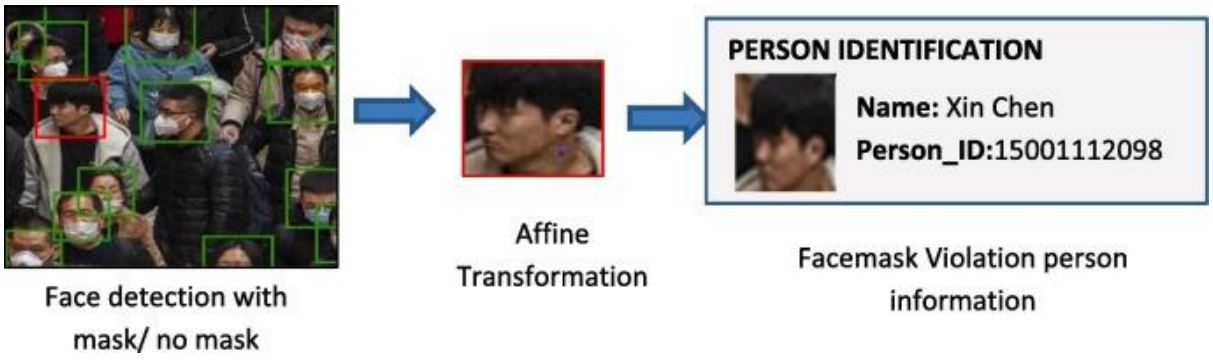

Fig.5. Affine Transformation for localizing the face with no mask

After applying the transformation, the boundary box regression is applied to map region proposal (R) to ground truth bounding box (G). The working of bounding box regression is discussed in detail here. Let each region proposal (face) is represented by a pair (R, G), where R = (R_x, R_y, R_w, R_h) represents the pixel coordinates of the centre 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 transformations are represented as T_x(R), T_y(R), T_w(R) and T_h(R). So, coordinates of the ground truth box can be obtained by equations (2), (3), (4), (5).

$$G_x = T_x(R_x) + R_x \quad (2)$$

$$G_y = T_y(R_y) + R_y \quad (3)$$

$$G_w = T_w(R_w) + R_w \quad (4)$$

$$G_h = T_h(R_h) + R_h \quad (5)$$

Here, each T_i (i denotes one of x, y, w, h) is applied as a linear function of the $Pool_6$ feature of R denoted by f_6 (R). Here, the dependence of f_6 (R) on R is implicitly assumed. Thus, T_i (R) can be obtained by equation (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 equation (7).

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

The regression target (t_i) related to coordinates, width and height of region proposal pair (R, G) are defined by equation (8), (9), (10), (11) respectively.

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

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

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

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

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. An optimization algorithm should 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 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 the loss.

Since the formula for cross-entropy loss given in equation (12) takes two distributions, $t(x)$, the true distribution and $e(x)$, the estimated distribution defined over discrete variable x [44], 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 categorical variable so it is chosen in our proposed model.

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

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

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

Chapter 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 be best fit as a backbone for detecting mask/non-mask faces using transfer learning?

RQ2: How do we evaluate the performance of image complexity predictor?

RQ3: How to check the utility of identity prediction in the proposed model?

RQ4: How does our model perform compared to the existing face mask detection model in terms of accuracy and computational speed?

RQ5: What measures should be considered to avoid overfitting?

4.1. Experimental setup

The experiment is set up by loading different pre-trained models using the Torch Vision package (https://github.com/pytorch/vision). These models are fine-tuned on our dataset using the open-source Caffe Python library. We choose our customized unbiased dataset with 45,000 images available online at https://www.kaggle.com/mrviswamitrakaushik/facedatahybrid. Int-Scenario training strategy is adopted as employed in [8]. The dataset is split into training, testing and validation set with 64:20:16 respectively. The algorithms are implemented using Python 3.7 and face detection is achieved through MobileNet-SSD/ResNet. .dib is used for detecting masks with learning rate = 0.003, momentum = 0.9 and batch size = 64.

4.2. Model comparison

As discussed in Section 3.2, we can apply transfer learning on pre-trained models for image classification 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 viz. ResNet50, AlexNet and MobileNet, based on 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**: It is the time taken by the model to predict the class of input image, that is starting from reading the image, performing all

intermediate transformations and finally generating the high confidence class to which the image belongs.

3. **Number of Parameters**: It is the total count of learnable elements present in all the layers of a model. These parameters directly contribute to prediction capability, model complexity and memory usage [45]. This information is very useful for understanding the minimum amount of memory required for each model. Further, it had been analysed by Simone Bianco et. al. that we require optimum number of learnable parameters so that trade-off between model accuracy and memory consumption may be achieved [45].

A model with minimum Top-1 error, less inference time on CPU and optimum number of parameters will be considered as a good model for our work.

The confusion matrices for different models during testing are given in Fig. 6 . The accuracy comparison of various models based on Top-1 error is presented graphically in Fig. 7 (a). It may be noted from the graph that the error rate is high in AlexNet and least in ResNet50. Next, we compared the model based on inference time. Test images are supplied to each model and inference times for all iterations are averaged out. It may be observed from Fig. 7(b) that MobileNet takes more time to infer images whereas ResNet and AlexNet take almost equal inference time for images. Further, the memory usage comparison among underlying models is done by finding the number of learnable parameters. These parameters can be obtained by generating model summary in Google colab for each model. It may be noted in Fig. 7(c) that the number of parameters present in AlexNet is around 28 million for our customised dataset. Furthermore, the number of parameters present in MobileNet and ResNet 50 are around 3.5 million and 25 million respectively.

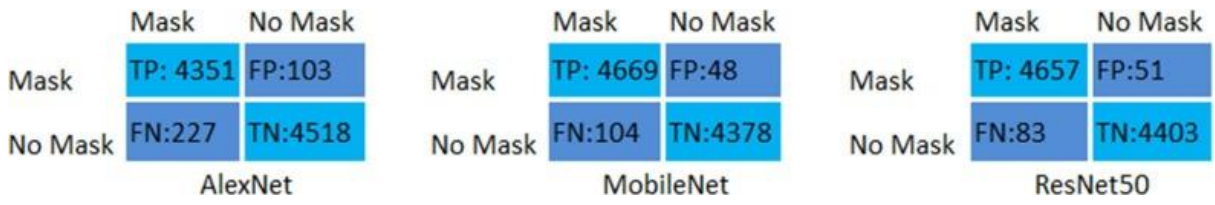

Fig.6. Confusion Matrix Obtained for Various Pre-trained Models.

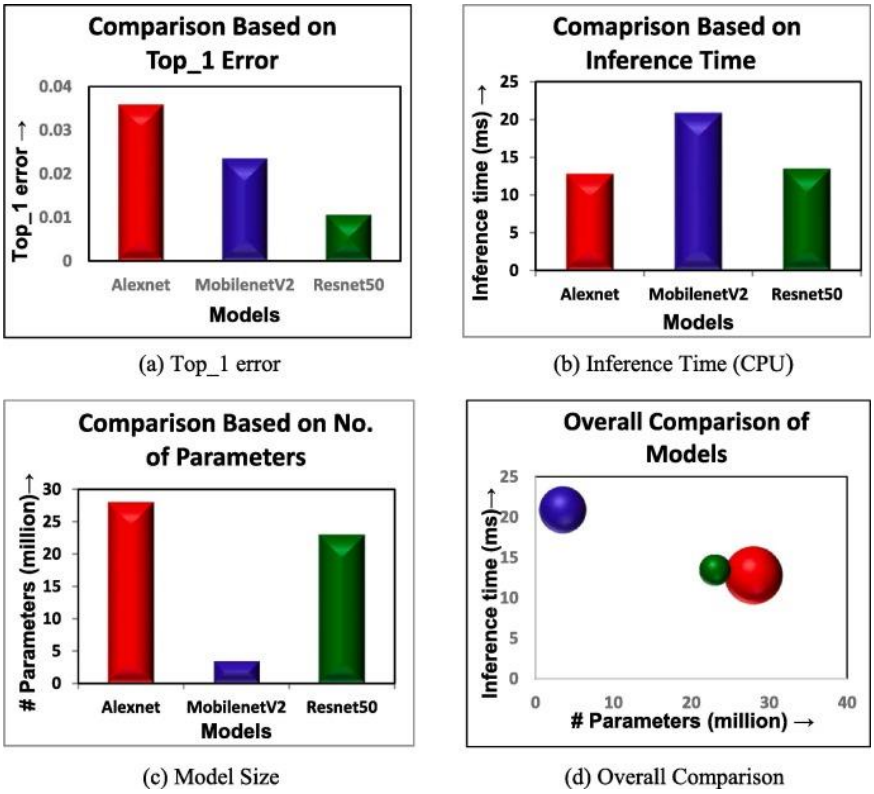

Fig.7. Comparison of Various Models on Different Performance Criteria.

After analyzing the performance of each model on various criteria, we then, squeezed all these details into a single bubble chart by taking no. of parameters as X-coordinate and inference time as Y-coordinate. The bubble size represents the Top-1 error (small bubble is better). The overall comparison of all models is represented by a bubble graph in Fig. 7(d).

It may be observed from Fig. 7, smaller bubbles are better in terms of accuracy and bubbles near the origin are better in terms of memory usage and inference 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, speed and memory usage for detecting face mask using transfer learning.

4.3. Performance analysis of image complexity predictor

For performance evaluation of the Image complexity predictor, we use Kendall's coefficient τ (tau). We compute Kendall's rank correlation coefficient τ between the predicted image complexity score and ground truth visual difficulty score. The Kendall's rank correlation coefficient is a suitable measure for our analysis

because it is invariant to different ranges of scoring methods. Based on image properties, each human annotator assigns a visual difficulty score to an image from a range that is different from the range, predicted image complexity score is assigned. The Kendal's rank correlation coefficient is computed in Python using kendalltau()SciPy function. The function takes two scores as arguments and returns the correlation coefficient. Our predictor attains Kendall's rank correlation coefficient τ of 0.741, implying the remarkable performance of the image complexity predictor. It may be observed from Fig. 8 that a very strong correlation exists between ground truth and predicted complexity scores.

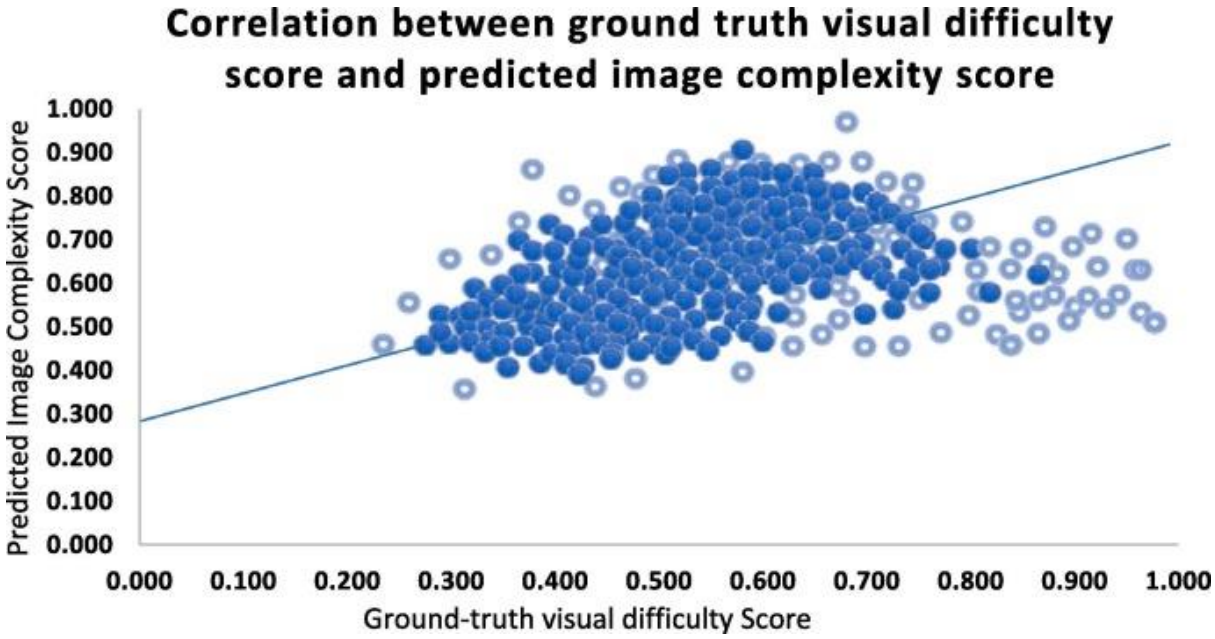

Fig.8. Correlation between Ground Truth Visual difficulty Score and Predicted Image Complexity Score

It may be further noted from Fig. 8 that the cloud of points forms a slanted Gaussian with principle component aligned towards diagonal, verifies a strong correlation between two scores.

4.4. Performance analysis of identity predictor

In order to impose wearing of face mask in public areas such as schools, airports, markets etc., it becomes essential to find out the identity of those faces which are violating the rules, means either not wearing or not correctly wearing a face mask. Typically, these identities can be found by training our model with persons faces.

For this purpose, the photographs of 2160 students are collected and populated in our customized dataset which is available online at https://www.kaggle.com/mrviswamitrakaushik/facedatahybrid. In order to well-train our system, we have taken five photographs of each student, ensuring face looking in different directions with different backgrounds. To further, proceed with the experiment, the video streaming from four CCTV cameras located at different locations in Department of Computer Applications, J. C. Bose University of Science and Technology, Faridabad, India is analysed. We captured the images from real-time video. Fig. 9 shows samples of images captured through different locations: Lecture room LT01, Corridor 2nd floor and staircase 2nd floor. To further, proceed with the experiment, the video streaming from four CCTV cameras located at different locations in Department of Computer Applications, J. C. Bose University of Science and Technology, Faridabad, India is analysed. We captured the images from real-time video. Fig. 9 shows samples of images captured through different locations: Lecture room LT01, Corridor 2nd floor and staircase 2nd floor.

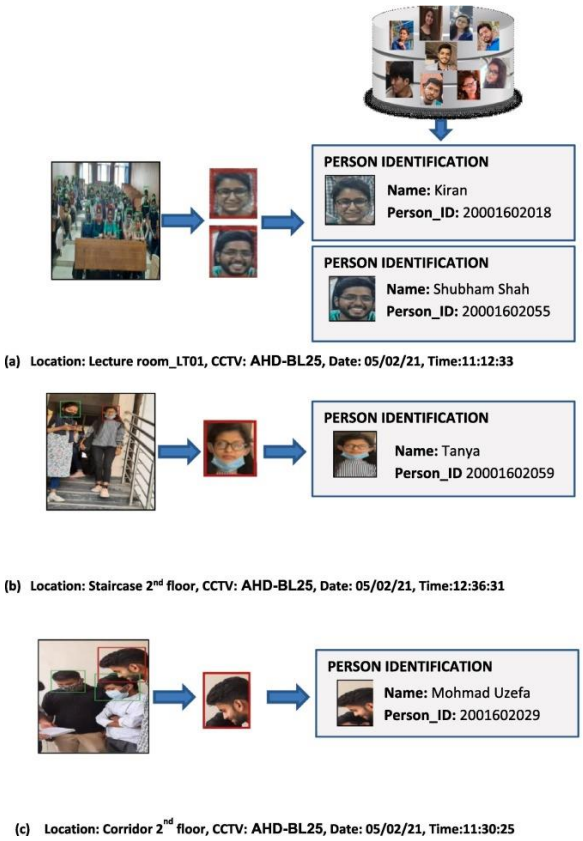

Fig.9. Identity Detection of Faces violating Mask Norms.

Precision and Recall are taken as evaluation metrics for identity prediction. The precision and Recall for identity predictor are 98.86% and 98.22% respectively.

4.5. Comparison of proposed model with existing models

In this section, we aim to compare the performance of the proposed model with public baseline results published in RetinaFaceMask [11], which aims to answer RQ2. Since RetinaFaceMask is trained on the MAFA dataset and performance is evaluated using precision and recall for face and mask detection so, for comparison purposes, the performance of the proposed technique is also evaluated in the same environment. We employed two standard metrics namely Precision and Recall for comparing the performance of these two systems. The experimental results are reported in Table 3 . It may be noted from Table 3 that the proposed model with ResNet50 as backbone achieves higher accuracy as compared to RetinaFaceMask.

Table 3

Comparison of Proposed model with Recent face mask detection Model

Model	Face Detection		Mask Detection	
	Precision (%)	Recall (%)	Precision (%)	Recall (%)
RetinaFaceMask based on MobileNet	83.0	95.6	82.3	89.1
RetinaFaceMask based on ResNet	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 the 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 optimized face detector discussed in Section 3.3 for dealing with complex images.

4.6. Controlling overfitting

To address RQ5 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 epochs both for the training and testing phase. The observations are reported in Fig. 10 .

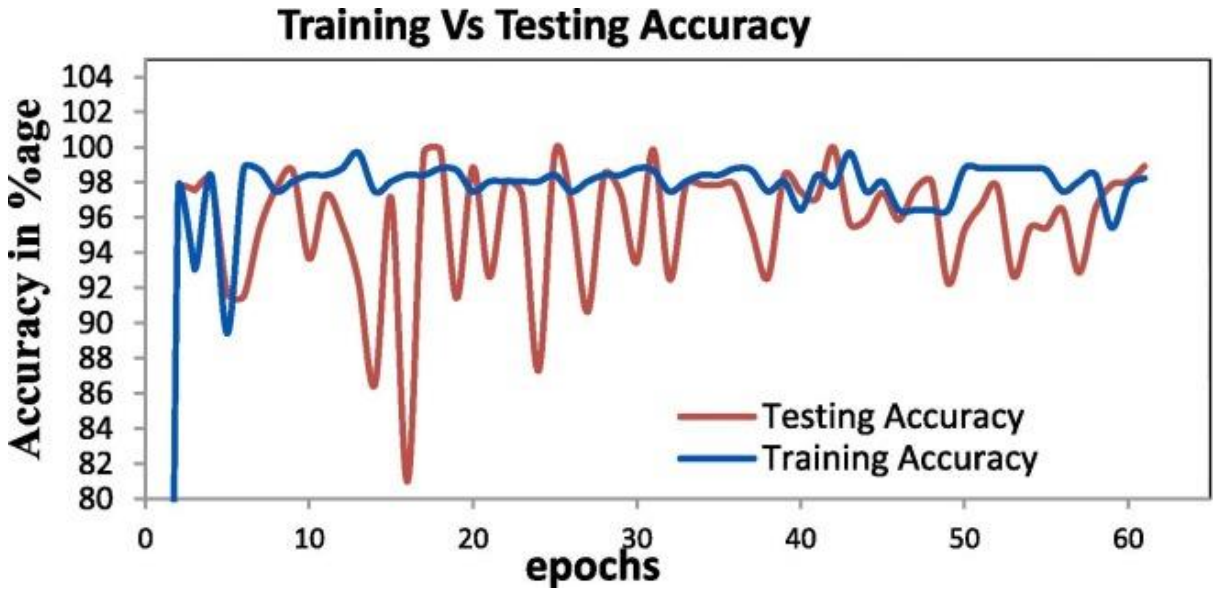

Fig.10. Training and Testing Accuracy over 60 Epochs.

It is further observed that model accuracy keeps on increasing in different epochs and get stable after epoch = 3 as depicted graphically in Fig. 10 above. To summarize the experimental results, we can say that the proposed model achieves high accuracy in face and mask detection with less inference time and less memory consumption as compared to recent techniques. Significant efforts had been put to resolve the data imbalance problem in the existing MAFA dataset, resulting in a new unbiased dataset which is highly suitable for COVID related mask detection tasks. The newly created dataset, optimal face detection approach, localizing the person identity and avoidance of overfitting resulted in an overall system that can be easily installed in an embedded device at public places to curtail the spread of Coronavirus.

Chapter 5: Conclusion And Future Scope

In this work, a deep learning-based approach for detecting masks over faces in public places to curtail the community spread of Coronavirus is presented. The proposed technique efficiently handles occlusions in dense situations 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 an unbiased dataset resulted in a highly robust and low-cost system. The identity detection of faces, violating the mask norms further, increases the utility of the system for public benefits.

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 extended to detect facial landmarks with a facemask for biometric purposes.

References

1. World Health Organization et al. Coronavirus disease 2019 (covid-19): situation report, 96. 2020. - Google Search. (n.d.).
2. Social distancing, surveillance, and stronger health systems as keys to controlling COVID-19 Pandemic, PAHO Director says - PAHO/WHO | Pan American Health Organization. (n.d.).
3. Garcia Godoy L.R. Facial protection for healthcare workers during pandemics: a scoping review, BMJ. *Glob. Heal.* 2020;5(5) doi: 10.1136/bmjgh-2020-002553.
4. Eikenberry S.E. To mask or not to mask: Modeling the potential for face mask use by the general public to curtail the COVID-19 pandemic. *Infect. Dis. Model.* 2020;5:293–308.
5. 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. (n.d.).
6. Nanni L., Ghidoni S., Brahnam S. Handcrafted vs. non-handcrafted features for computer vision classification. *Pattern Recogn.* 2017;71:158–172. doi: 10.1016/j.patcog.2017.05.025.
7. Y. Jia et al., Caffe: Convolutional architecture for fast feature embedding, in: MM 2014 - Proceedings of the 2014 ACM Conference on Multimedia, 2014, doi: 10.1145/2647868.2654889.
8. P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. Lecun, OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks, 2014.
9. Erhan D., Szegedy C., Toshev A., Anguelov D. *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2014. Scalable Object Detection using Deep Neural Networks; pp. 2147–2154.
10. J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016, vol. 2016-Decem, pp. 779–788, doi: 10.1109/CVPR.2016.91.