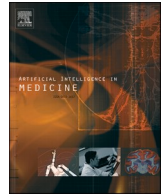




Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Elsevier hereby grants permission to make all its COVID-19-related research that is available on the COVID-19 resource centre - including this research content - immediately available in PubMed Central and other publicly funded repositories, such as the WHO COVID database with rights for unrestricted research re-use and analyses in any form or by any means with acknowledgement of the original source. These permissions are granted for free by Elsevier for as long as the COVID-19 resource centre remains active.



Research Paper

CoSumNet: A video summarization-based framework for COVID-19 monitoring in crowded scenes

Ambreen Sabha^{*}, Arvind Selwal

Department of Computer Science and Information Technology, Central University of Jammu, Samba, Jammu and Kashmir 181143, India



ARTICLE INFO

Keywords:

COVID-19
Object detection
Face mask detection
Video summarization
Deep learning

ABSTRACT

The outbreak of COVID-19 pandemic poses new challenges to research community to investigate novel mechanisms for monitoring as well as controlling its further spread via crowded scenes. Moreover, the contemporary methods of COVID-19 preventions are enforcing strict protocols in the public places. The emergence of robust computer vision-enabled applications leverages intelligent frameworks for monitoring of the pandemic deterrence in public places. The employment of COVID-19 protocols via wearing face masks by human is an effective procedure that is implemented in several countries across the world. It is a challenging task for authorities to manually monitor these protocols particularly in densely crowded public gatherings such as, shopping malls, railway stations, airports, religious places etc. Thus, to overcome these issues, the proposed research aims to design an operative method that automatically detects the violation of face mask regulation for COVID-19 pandemic. In this research work, we expound a novel technique for COVID-19 protocol desecration via video summarization in the crowded scenes (CoSumNet). Our approach automatically yields short summaries from crowded video scenes (i.e., with and without mask human). Besides, the CoSumNet can be deployed in crowded places that may assist the controlling agencies to take appropriate actions to enforce the penalty to the protocol violators. To evaluate the efficacy of the approach, the CoSumNet is trained on a benchmark “Face Mask Detection ~12K Images Dataset” and validated through various real-time CCTV videos. The CoSumNet demonstrates superior performance of 99.98 % and 99.92 % detection accuracy in the seen and unseen scenarios respectively. Our method offers promising performance in cross-datasets environments as well as on a variety of face masks. Furthermore, the model can convert the longer videos to short summaries in nearly 5–20 s approximately.

1. Introduction

The coronavirus disease 2019 (COVID-19) is an ongoing global pandemic that is caused by severe acute respiratory syndrome (SARS-CoV-2). In December 2019, the virus originated from the city of Wuhan of China and rapidly spread across the world. According to a report of WHO (World Health Organization) that is released on August 16, 2020, the coronavirus disease has infected more than 6 million individuals worldwide and a casualty of more than 379,941 peoples was reported [1]. The COVID-19 has impacted every aspect of human life as well as hindered the global economy. As per the reports of WHO, the COVID-19 has spread in more than 213 countries across the world [2,3]. To tackle the spread of the virus, several countries enforced complete lockdown resulting in closure of their mundane facilities, utilities, educational institutions, industries, shopping malls, public transport, etc. In nutshell,

the pandemic has posed several challenges ranging from daily routine to other commercial activities in intra as well as inter-countries.

The majority of the nations are using precautionary measures to prevent the spread of virus by employing COVID-19 protocols such as lockdown, compulsory wearing of face masks, appropriate physical distance, random testing at public places, etc. Among all, face mask wearing is an effective measure that is widely followed in most of the nations. However, a strict enforcement of face masks wearing protocol in public places is a challenging task for security agencies, particularly in the densely crowded places such as, shopping malls, railway station, airport, religious places, etc. The public health agencies make use of several steps to mitigate COVID-19 spread by using artificial intelligence techniques such as machine or deep learning algorithms. To overcome the challenges in monitoring face mask protocols via traditional methods, the intelligent video analysis is a viable mechanism where the

^{*} Corresponding author.

E-mail address: ambreensabha45@gmail.com (A. Sabha).

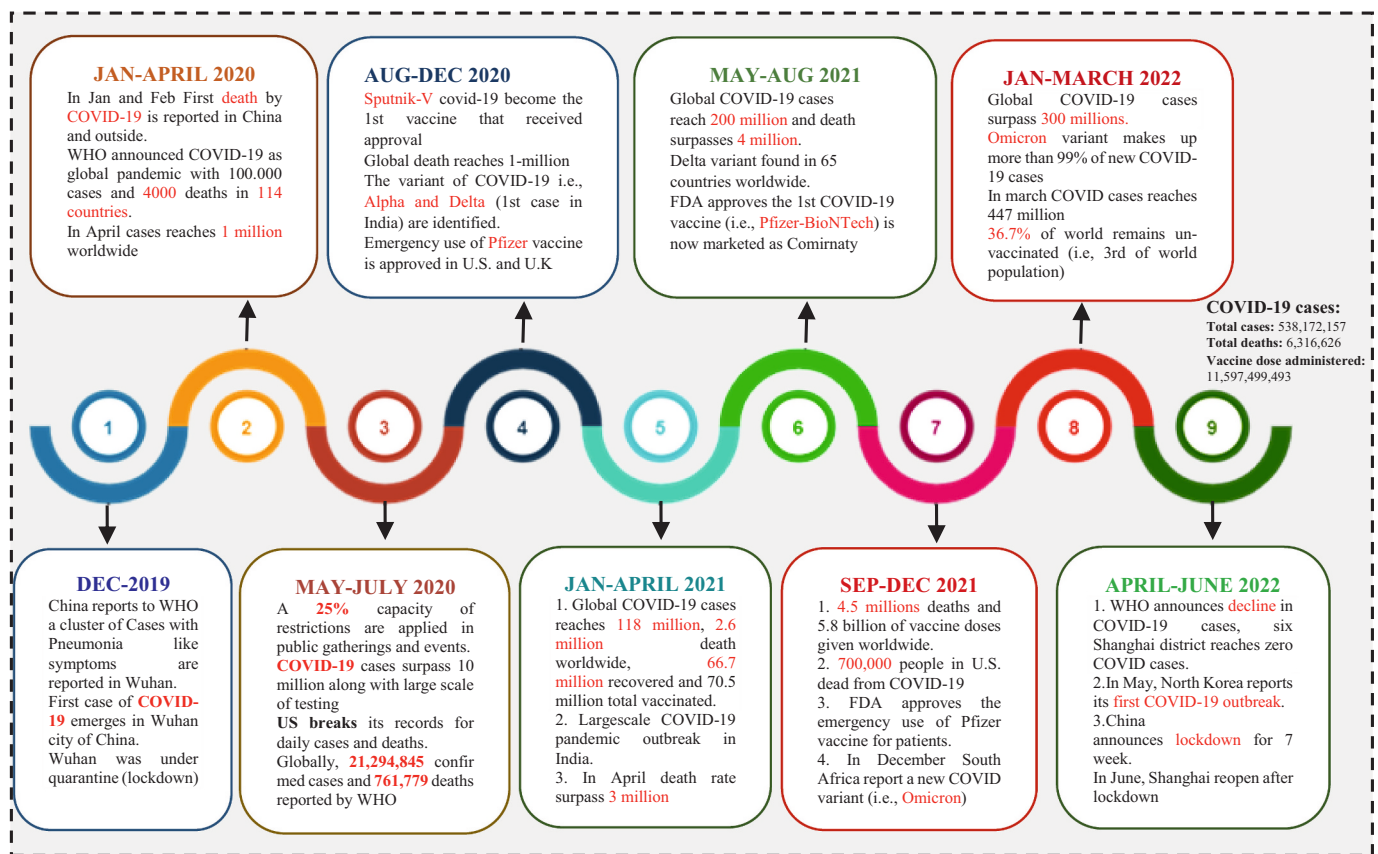


Fig. 1. The breakthrough and progression of COVID-19 pandemic since December 2019.

violators can be accurately detected. Additionally, the video summarization is used to generate a shorter summary from long videos by satisfying a specific criterion that makes use of key-frames selection approach [4,5]. Moreover, video summarization includes variety of applications in various sectors such as sports [6–12], medical [13–15], education, surveillance [16–22], egocentric videos [23–27] and etc. The notion can be used to detect human (i.e., with or without mask) in crowded scenes and it produce short video summaries accordingly. The technique of face mask detection refers to predict a face in a video frame with or without face masks. Although, several research contributions are available in the existing literature for face mask detection in the images, but to the best of our knowledge there exist no studies on face mask-based video summarization. Thus, the main objective of this research work is to design a novel video summarization model that creates smaller video summaries of COVID-19 face mask violators from the real-time videos. Apart from this, CoSumNet yield accurate face mask rule violators in the public gathering to minimize the further spread of the pandemic.

Hence, the key motivation behind the proposed work is to develop a novel and efficient framework that accurately detect and generate summaries of persons with and without face mask to monitor the spread of the disease and thereby assisting the public health workforces. Furthermore, to build a robust video summarization model the notion of domain adaptation is employed in CoSumNet where a pre-trained finetuned deep convolutional neural network (DCNN) is adopted.

The major contributions of our work are listed as follows:

- i. A novel CoSumNet framework is proposed for COVID-19 face mask protocols monitoring in crowded places.
- ii. The CoSumNet employs a potent pre-trained DCNN for efficient classification of video frames to with and without face mask video summaries.
- iii. The proposed model is trained on benchmark Face Mask Detection ~12 K Images Dataset and evaluated on real-time CCTV footage.
- iv. Our approach demonstrates its effectiveness in both known and unknown environments as well as excellent performance on a variety of unseen face masks samples.
- v. The CoSumNet aids in detecting and monitoring COVID-19 violations of face mask protocols with promising accuracy and low training overhead.

2. Background

The outbreak of COVID-19 in December 2019 that has spread across the world is an ongoing pandemic, which has witnessed several waves hitherto [2,3,28–30]. The deadly virus that was first reported in an outbreak in the Chinese city of Wuhan has appeared in a variety of variants. The common symptoms of the COVID-19 disease include such as, fever, dry cough, fatigue, body aches, lack of taste, etc. The virus mainly spreads when people come into close contact with respiratory droplets produced by an infected person's cough, sneeze, or exhale. As these droplets are too dense to swing in the air for long distances, they quickly settle on the floors or surfaces and can spread when the individuals touch the surface contaminated with the virus followed by touching their own face (such as the eyes, nose, and mouth). The emergence of COVID-19 since December 2019 up to June 2022 is

depicted through a timeline as shown in Fig. 1. The timeline clearly highlights the breakthrough that has been reported during the period [3,28,29,31].

As per the recent studies, the second wave of COVID-19 that hit India during 2021 [32] was more severe and the virus spread comparatively quicker than the first wave in 2020. According to a report by health ministry of India, the number of cases of COVID-19 increased from 8105 to 97,894 within 110 days in the first wave. Whereas, the second wave began in February 2021 that exhibited an upward trend with drastic increase in COVID-19 patients from 8365 cases per day to 1,03,558 cases per day within 62 days [33].

To tackle the biggest pandemic of the century, researchers started exploring various AI-driven mechanisms that are assisted by computer-vision enabled solutions. Nowadays, several issues in medicine are successfully addressed by the researchers via the field of computer vision such as video analysis, object identification, segmentation, localization, and classification. Among all, video analysis is one of the momentous solutions that process huge number of videos and thereafter can classify these to different categories. In early 1990s, the traditional techniques were employed for analysis and classification problems, but there are some limitations in these approaches such as, manual features extraction and limited accuracy. To tackle these problems, the paradigm has shifted from traditional machine learning algorithms to contemporary deep convolutional models. These DCNN models involve the mechanisms of automatic features extraction from the given input videos or images. The deep learning methods extract deep level of features from images that is not possible in the case of their conventional counterparts. The detection of individuals following COVID-19 protocol by wearing face mask in public places is one of the challenging tasks for controlling the further spread of the virus. Thus, the automatic and effective face mask detection techniques in video scenes are a crucial task. Some of the researchers have contributed work in this field. Earlier, Goyal H. et al. [34] proposed a face mask detection model for static and real-time videos that distinguish between images with and without face masks. The model is trained and evaluated on the dataset that is available on Kaggle platform, which include 4000 samples and it shows an accuracy of 98%. The comparison of the proposed model with existing pre-trained architecture such as, MobileNet-V2, DenseNet-121, VGG-19, and Inception-V3 reveals that it is more accurate and computationally efficient. Later, Sethi S. et al. [35] proposed an ensemble-based method that use one-stage and two-stage object detectors to identify human with face mask. In addition, they make use of a bounding box transformation method to locate and identify people without face masks. Three well-known pre-trained models, ResNet50, AlexNet, and MobileNet are used in the experiment, where ResNet50 results in promising accuracy of 98.2%. As face recognition algorithms frequently require a front face input image, head pose classification is frequently employed for the pre-processing prior to face recognition. However, for COVID-19 safety, individuals wear face masks that cover the entire face. There occurs a problem in the classification of head pose along with facial recognition and the issue is overcome by Li S. et al. [36], where they employed CNN (Convolution Neural Network) to identify human with or without face mask. They created the HGL approach, which make use of line portraits and colour texture analysis to categorise head pose utilising masks for faces. The side-to-side accuracy is evaluated as 87.17%, while the front face accuracy is 93.64%.

In the current COVID-19 era, the correct face mask wearing is an important protocol for human safety, but the effectiveness of face masks wearing is declining due to its inappropriate usage. Furthermore, Qin and Li [37] proposed a system that combines picture super-resolution and classification networks to identify the proper wearing of face mask (SRCNet). The benchmark medical masks dataset is used which contain a total of 3835 samples out of which 671 images belongs to without face mask class, 134 images of people wearing face mask incorrectly, and 3030 images of individual wearing correct face mask. The dataset is pre-processed by using various steps such as, face

detection, cropping, image super-resolution, and appropriate face mask-wearing identification. The proposed SRCNet outperformed other similar methods with an accuracy of 98.70%. To mitigate the further spread of virus, it is key to address the real-time detection of face mask wearing by the individuals in public places. In another work, Hussain D. et al. [38] proposed an automatic prediction system for face mask detection that utilize deep learning models. The MobileNetV2 pre-trained model along with a DCNN is used for effective face mask detection. The proposed approach is evaluated on a secondary 'dataset-1' along with self-created 'dataset-2' through real-world scenarios having 2500 samples. The experimental results with DCNN achieves an accuracy of 97% on both the dataset. Whereas, MobileNetV2 offers an accuracy of 98% and 99% on datasets-1 and dataset-2 respectively. Additionally, Kodali R. et al. [39] proposed a three-step approach for face mask detection that include pre-processing, CNN learning, and real-time classification. The pre-processing is the first step, where face images are resized before converting to grayscale followed by image normalization operation. As the problem is binary classification, a SoftMax function is employed in the classification header of the CNN model. In yet another work, Jiang M et al. [40] proposed a model namely; RetinaFaceMask to detect face mask in given images. The model utilizes multiple features maps to form a feature pyramid network for extracting the high-level features from an image. In addition, a context attention model is used that enhances the detection accuracy and it correctly identify images that contains human with face mask. The experimental analysis is performed on a face mask dataset in terms of precision and recall parameters and it results in comparatively 2.3% and 11.0% higher performance than existing similar techniques. Later, Fan X. and Jiang M et al. [41] present a single stage face mask detector model namely; RetinaFaceMask, which is an extended version of their previous work [40]. In this approach, firstly a dataset MAFA (i.e., masked faces) is created that consist of three different classes namely; no mask, correct mask, incorrect mask wearing. Thereafter, a novel context attention module (i.e., CAN) is build that extract deep level features from images for discriminating among the three classes. In addition, the transfer learning approach is used to enhance the performance of model by transferring the knowledge from the base model to the face mask detector. The experimental analysis is performed on both public and private dataset and it demonstrate superior performance against other existing techniques. Manual detection of face masks in crowded places such as face detection, facial expression detection and face-based biometric attendance with face mask is a difficult task. In addition, there exist no standard benchmark dataset for face mask detection along with variations of masks. Therefore, to overcome this problem, Wang Z. et al. [42] presented three types of face mask dataset namely; Real-world Masked Face Recognition Dataset (RMFRD), Masked Face Detection Dataset (MFDD), and Simulated Masked Face Recognition Dataset (SMFRD). Among all, RMFRD is the largest face mask dataset and is openly available to research community. The experimental analysis shows that DCNN model achieves an accuracy of 95% on RMFRD dataset. Furthermore, Jayaswal R. et al. [43] proposed a CLAHE-SSD_IV3 model for face mask detection. Initially, for image pre-processing, the contrast limited adaptive histogram equalization (i.e., CLAHE) technique is used to enhance the quality of images. Thereafter, a pre-trained model such as Inceptionv3 along with SSD face detector is used to learn Real-time face mask dataset (RTFMD) for classification purpose. The proposed model is compared with other pre-trained models such as, VGG16, VGG19, MobilenetV2 and Xception at various parameters. Besides, the proposed approach is compared with other similar techniques on face mask dataset and RTFMD and it results in an accuracy of 98% and 97% respectively. In the same year, Jayaswal R. et al. [44] developed another approach for face mask detection model to predict human with and without face mask. The model consists of single shot multi-box face detector along with a pre-trained Inceptionv3 to extract few features and it is named as SSDIV3. The approach is compared with other pre-trained models (such as, VGG16, VGG19,

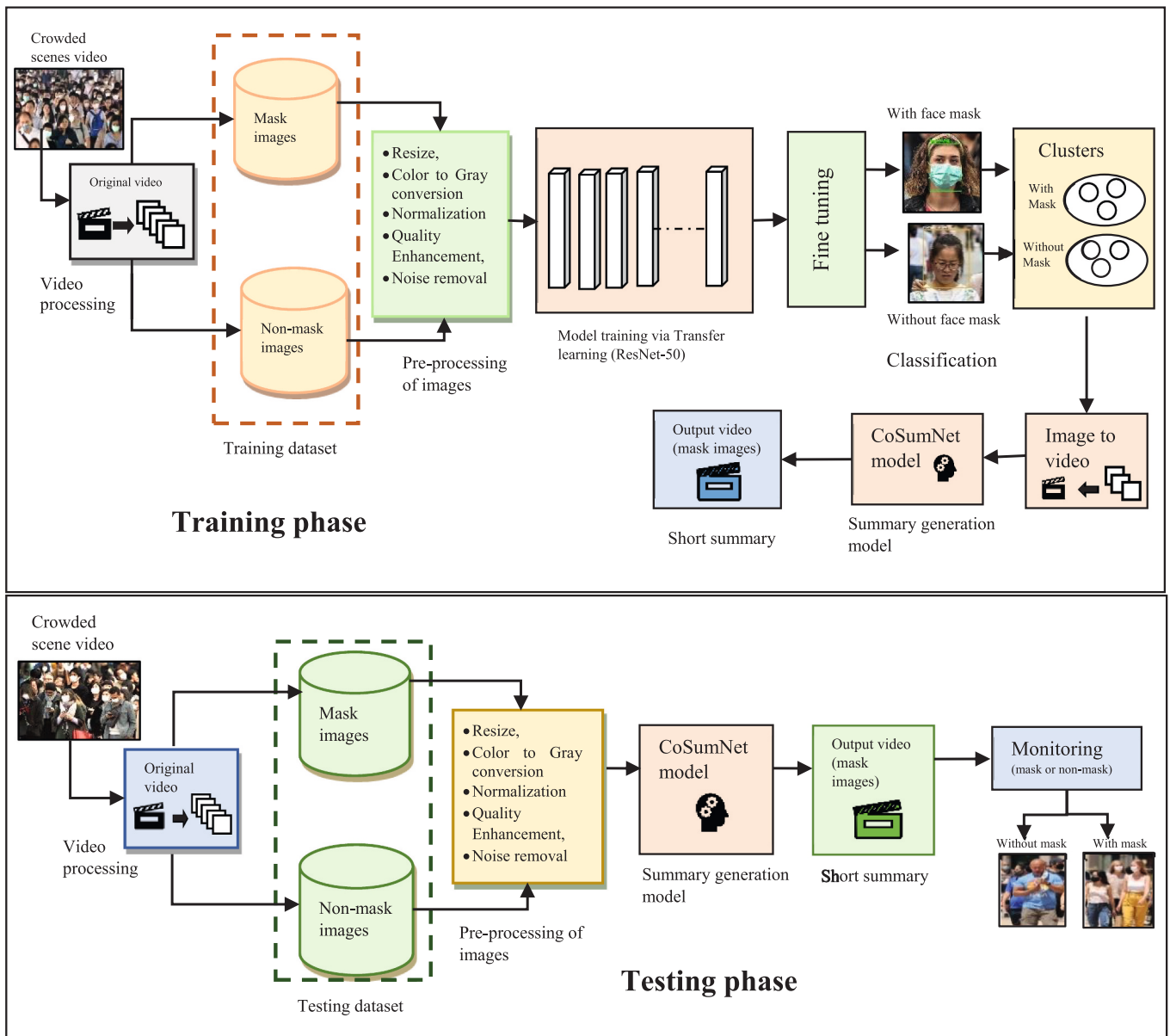


Fig. 2. The proposed framework of CoSumNet model.

Mobilenet V2 and Xception). As compared to their earlier work, the present technique make use of two novel datasets comprising of samples with a variety of face mask in both indoor and outdoor environment. The approach outperforms the existing state-of-the-art techniques with an accuracy of 98 % on newly created dataset in real-time environment. Jayaswal R. et al. [45] extended their earlier work, where a deep learning approach is used to detect face with and without mask. The method used a newly designed face mask dataset in real-time (i.e., FMDRT) to detect whether a person is wearing a mask or not. For image pre-processing, the same CLAHE approach as used in [43], is employed to enhance the quality. Apart from this, to detect a face, a pre-trained Caffe face detector model is utilized along with Xception model for extracting the deep level features from images and named it as CL-SSDXcept. The model is compared with different pre-trained models at various parameters settings on FMDRT dataset. The performance of the model is evaluated against existing techniques on both FMDRT and face mask detection dataset. The experimental analysis reveals that approach achieves an accuracy of 98 % and loss of 0.05, as well as its superiority over other methods.

3. Proposed approach

Though, majority of the COVID-19 studies are related to detection and monitoring of individuals with and without face masks, but there exist no model that can automatically detect the protocol violators in videos of public gatherings. Hence, the current work is aimed at detection and summarization of human with and without face mask in CCTV videos. In this section a detailed depiction of framework of CoSumNet is presented along with a description of the relevant algorithms. In addition, a thorough illustration of each step of training and testing phase is also expounded.

3.1. CoSumNet framework

The proposed CoSumNet framework is presented in Fig. 2, that consist of two phases namely; training and testing. In training phase, initially the video captured through crowded scenes is processed, where the frames are extracted from original CCTV. Thereafter, with and without face mask images in training dataset are pre-processed by

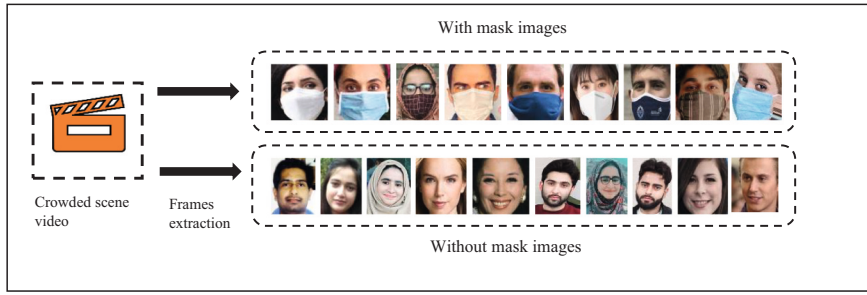


Fig. 3. An illustration of frames extraction from a crowded scene video.

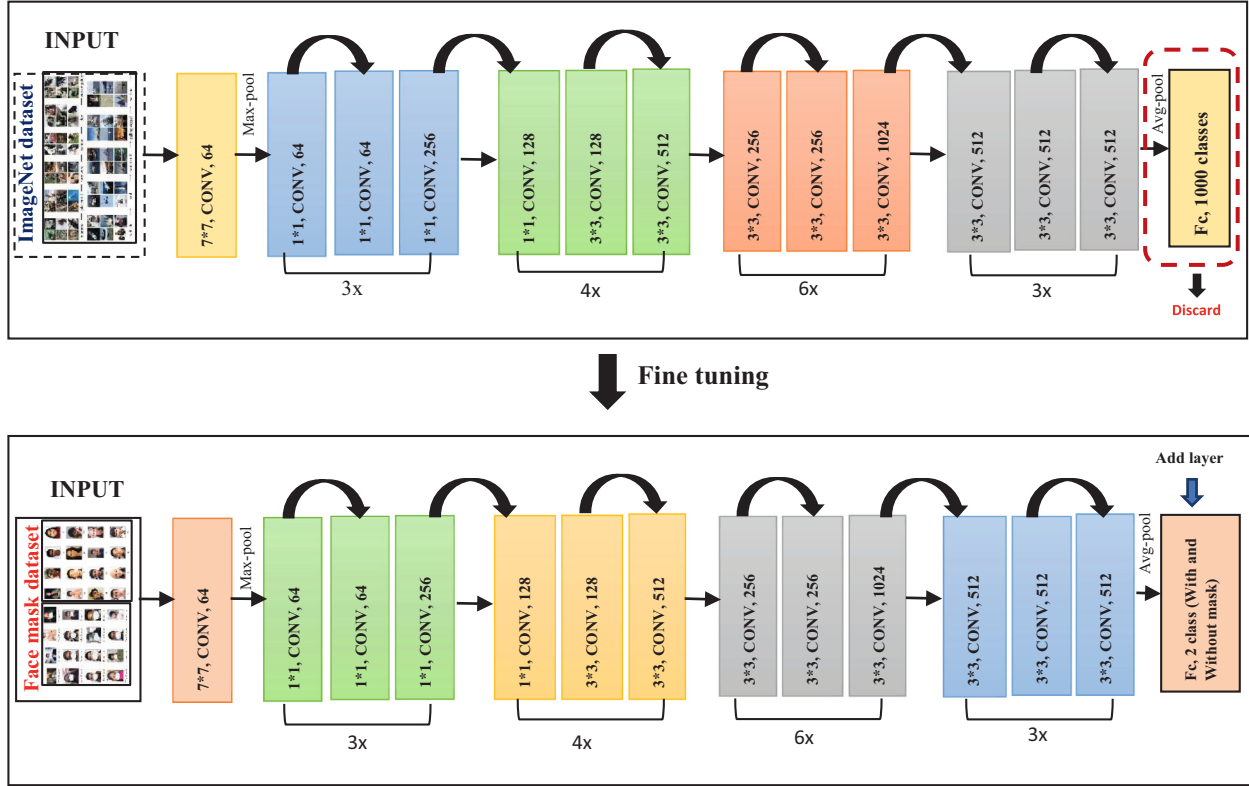


Fig. 4. An architecture of pre-trained ResNet-50 model (adopted from [46]).

applying elementary image processing operations such as resizing, colour conversion, normalization, quality enhancement and noise removal. The notion of transfer learning is used, where a pre-trained model (i.e., ResNet50) is used for the learning purpose to enhance the overall efficiency. Afterwards, the model is fine-tuned by selecting the optimal parameters to increase the performance of CoSumNet. The last layer of the fine-tuned model performs the task of classification of a given frame as with or without mask. Then, resultant classified frames are augmented into two separate clusters. Later, the frames in each cluster are converted to two videos thereby resulting two short summaries of with or without mask(i.e. protocol violators). The stepwise operation of CoSumNet model is discussed in subsequent subsections.

3.1.1. Video frames extraction

A video typically consists of a large number of frames that are captured at a given interval of time. Initially, an original video captured via CCTV camera from crowded scenes is processed to extract frames with a rate of 25–30 fps as shown in Fig. 3.

Mathematically, a video is represented as $f(x,y,t)$, where $f(x,y)$ are the spatial coordinates and 't' represents the temporal representation.

The model takes video as an input from dataset and extract frames from these videos one after another as well as stored in to a list. As depicted in Eqs. (1) and (2), 'V' is the original video that is treated as combination of multiple frames ' f_i ', where i ranges from 1 to n .

$$V = \sum_{i=1}^n (f_i) \quad (1)$$

$$f_i = \sum \{f_1, f_2, f_3, \dots, f_n\} \quad (2)$$

3.1.2. Image pre-processing

Once the frames are extracted from original video, then each frame is assessed for its quality and uniformity by applying some pre-processing techniques. Then images are resized to 224×224 to ensure input data is of same dimension that is pre-requisite for deep learning model. Thereafter, frames are converted to grey scale channel for less computations and are subjected to image enhancement techniques such as, histogram equalization, normalization, noise removal and filtering to improve its brightness and smoothness. The average and smoothing filters are applied to the images to eliminate the undesirable noise, and

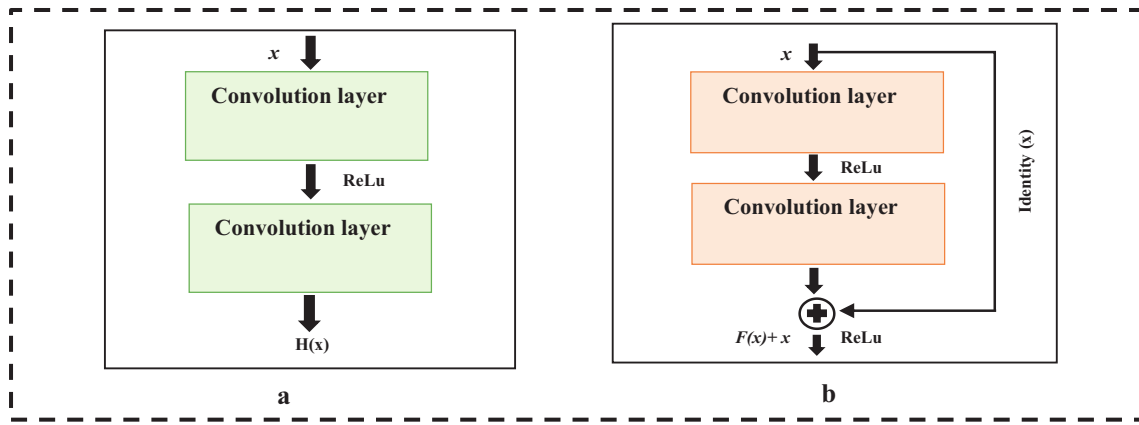


Fig. 5. A depiction of skip connections in ResNet-50 (a) plain layer network (b) Residual learning blocks.

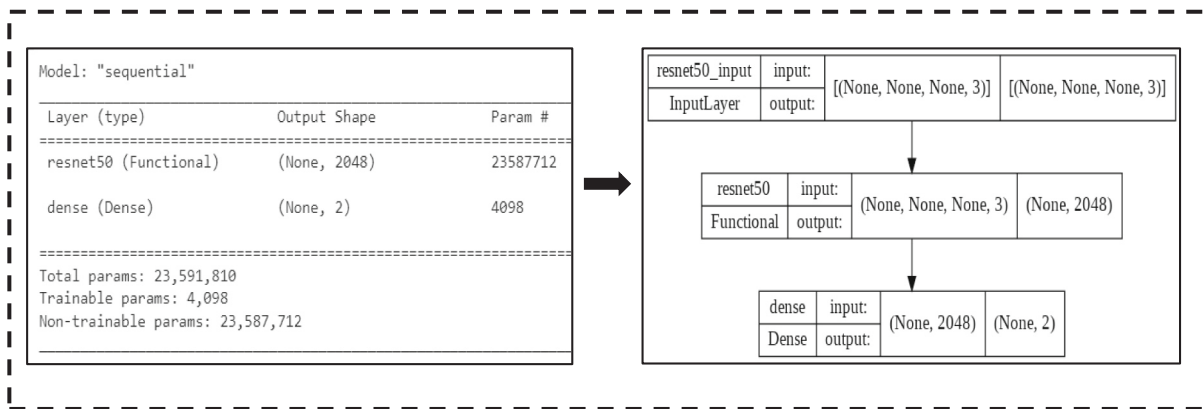


Fig. 6. A summary of fine-tuned ResNet-50 model.

for faster and efficient computing, the training images are normalized by dividing each pixel intensity value with 255 that gives values between 0 and 1. The images are categorized into two classes namely; with mask and without mask. A label encoding method is applied to each frame and assign numerical values to the both the classes i.e., 0 and 1 respectively for with mask and without mask. To further increase the dataset, the concept of data augmentation is used to introduce variations in the images through a series of operations such as, rotation, flipping, scaling zooming, etc. Thereafter, these pre-processed images are further given as an input to the pre-trained deep learning model.

3.1.3. Fine tuning of pre-trained model

In the proposed work, DCNN is used due to its superior performance compared to conventional classification methods for facemask detection via video summarization. The training of DCNN is costly since it takes a long time for training purpose and has a lot of computational overhead. Hence, the conception of transfer learning is used to train the model more quickly and efficiently. As it enables the neural network's trained knowledge in terms of parametric weights to be transferred to the new model to improve the performance that is trained on smaller number of samples. The ImageNet dataset that constitutes of 14 million images and it is used to train several benchmark pre-trained models, such as, VGGNet, AlexNet, MobileNet, ResNet50, GoogleNet, Inception, etc. Among all, the pre-trained model ResNet50 is used in the our work (i.e., CoSumNet), for facemask detection via video summarization as shown

in Fig. 4. The ResNet (i.e., residual network) was first introduced in 2015 by Ren S. et al. [46] in their research paper titled “Deep Residual Learning for Image Recognition”. It has some advantages as compared to other pre-trained models such as the presence of residual blocks and skip connections along with identity mapping function. The ResNet model comprises of these residual blocks and the skip connections are present inside the network’s residual blocks. The problem of training a deep network has been resolved by the employment of these blocks and it is one of the most robust and popular deep learning models. The architecture is then transformed into the residual network by these skip connections or residual blocks. The CoSumNet model has undergone hyper tuning, and the original ReNet50 includes 1000 classes in the last fully-connected layer. To build CoSumNet, the last layer from the original model is removed and a new layer with two classes is augmented as depicted in Fig. 4.

Another critical issue in DCNNs include the problem of exploding and vanishing gradient that is overcome by ResNet pre-trained architecture via skip connections. The researchers are using DCNN for computer vision related tasks by adding a greater number of layers for achieving high accuracy. These additional numbers of layers may help to solve the complex problems more accurately. While higher number of layers are added to the model, it can extract the deep level features, but these models result in the issue of degradation. In other words, as the number of layers increases, the accuracy may eventually get saturated and begin to slowly deteriorate. As a result, the model’s performance

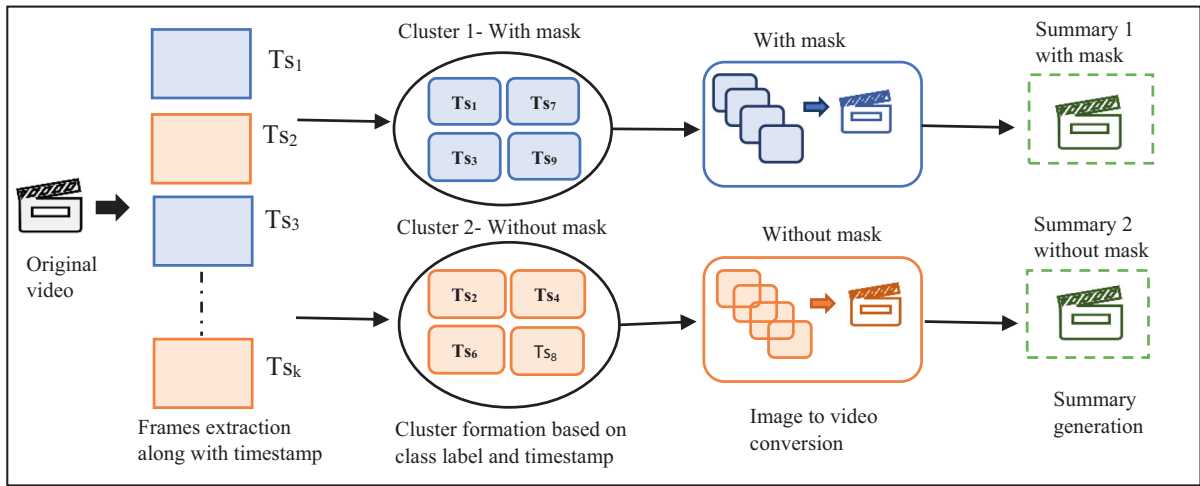


Fig. 7. An illustration of model building and summary generation.

declines on both the training and testing dataset resulting in overfitting problem. Instead, it might be the result of the network's initialization, an optimization function, or more crucially an issue with vanishing or exploding gradients. Hence, ResNet50 model was developed with the intent of solving this issue by incorporating the concept of skip connections that present inside these residual blocks as depicted in Fig. 5. These skip connections operate in two different ways: firstly, they solve the problem of vanishing gradient by creating an alternative shortcut for the gradient that is to be used. Secondly, these provide the model an ability to learn the identity function to reduce the computational overhead without adding new parameters. The core principle of ResNet50 is the introduction of an "identity shortcut connections" that bypasses one or more network levels. The strategy behind this network is that rather than having layers learn the underlying mapping, they let the network fit the residual mapping via skip connection, which skips training from a few layers and connects directly to the output. Thus, let the network fit, instead of using the initial mapping of $H(x)$ as depicted in Eq. (3) where, $F(x)$ is the mapping function and x is the input of the model. Fig. 5 (a) depicts the plain layer network whereas, residual blocks architecture is depicted in Fig. 5 (b).

$$F(x) = H(x) - x \text{ which gives } H(x) = F(x) + x. \quad (3)$$

As our model does not contain large number of samples in the training dataset, we employ the conception of domain adaptation through ResNet50. The model is fine-tuned by discarding its last fully connected layer that originally contains 1000 units and then replacing it with a two units (i.e., with mask and without mask). The classification layer help to classify frames to binary classes. In last layer, a ReLu and SoftMax activation function is used to produce probabilistic output, the mathematical representation of these functions is shown in Eqs. (4) and (5). The summary of fine-tuned ResNet50 model is shown Fig. 6.

$$Relu : f(x) = \max(0, x) \quad (4)$$

$$SoftMax : (xi) = \frac{\exp(xi)}{\sum_j \exp(xj)} \quad (5)$$

3.1.4. Model building and summary generation

After the fine tuning of pre-trained model, the summarization model is built to generate short summaries of with or without mask as shown in Fig. 7. Thereafter, the model classifies the images into two classes (i.e., with and without mask) and these are stored in two clusters. Each cluster contain the images that belongs to the respective class and the redundant frames are removed. The images in each cluster are timestamped (ts). As the image-1 that occurs at ts-1 belong to with mask class is to be placed in cluster 1 and similarly, for ts-2 the image-2 is placed under cluster 2 that belong to without mask class.

The clusters containing face mask images (i.e., with and without mask) are converted to video by using the VideoWriter function to generate a short summary of a larger video. Thus, the summarized video consist of two summaries that include human wearing face mask and without face mask.

3.1.5. Covid-19 monitoring

In the testing phase, the trained CoSumNet model monitors the video captured through CCTV videos in crowded scenes. Initially the frames are extracted from the video and then pre-processing of each frame is carried out. Thereafter, each pre-processed frame is examined by applying CoSumNet model to detect the frames with mask and without-masks. The frame with mask are converted to a shorter video and for human without mask frames, an alert is sent to concerned COVID-19 protocol violators. In this manner, our CoSumNet is used for monitoring the further spread of the COVID-19 in crowded scenes.

3.2. Proposed algorithms

An algorithm 1 for building the CoSumNet model is presented in this section. Suppose ' V_i ' be the video dataset that is given as an input to the model and it returns output as ' M ' i.e., CoSumNet model. Let ' l ' is the size of learning dataset ' V_i ' and Φ_M , Φ_N and L are the empty lists for face mask detection with two classes (i.e., Φ_M for with mask and Φ_N for without mask).

Algorithm 1. To build the CoSumNet model.

Table 1
A summary of benchmark face mask dataset.

| Class | Training images | Validation images | Testing images |
|--------------|-----------------|-------------------|----------------|
| With mask | 5000 | 400 | 483 |
| Without mask | 5000 | 400 | 509 |
| Total | 10,000 | 800 | 992 |

label the mask and non-mask frames such as ‘0’ for \hat{Z}_M and ‘1’ for \hat{Z}_N . The pre-trained model i.e., P_t is loaded and stored in Ψ , in pre-trained model the last layer i.e., fully connected layer is discarded and a new layer is added having 2 class (i.e., with and without mask) along with the SoftMax activation function. The model is fitted on training dataset on ‘1’ number of iterations, wherein the x-train and y-train are the dataset and the labels. The model then predicts on x-test and y-test dataset to check

```

Input: Video dataset ‘ $V_l$ ’ (learning dataset)
Output: CoSumNet ‘ $M$ ’

Begin
   $l \leftarrow |V_l|$ 
   $\Phi_M = [], \Phi_N = [], L = []$ 
  for  $k = 1$  to  $l$  do
     $L \leftarrow L.append[frames\_extraction-V_l(k)]$ 
  end for
  for  $f$  in  $L$  do
    if face-mask-detector ( $f$ ) = True
       $\Phi_M \leftarrow \Phi_M.append(f)$ 
    else
       $\Phi_N \leftarrow \Phi_N.append(f)$ 
    end for
  end for
   $\Phi'_M \leftarrow$  pre-processing ( $\Phi_M$ )
   $\Phi'_N \leftarrow$  pre-processing ( $\Phi_N$ )
   $\hat{Z}_M \leftarrow$  extract-label ( $\Phi'_M$ )
   $\hat{Z}_N \leftarrow$  extract-label ( $\Phi'_N$ )
   $[0] \leftarrow$  encode ( $\hat{Z}_M$ )
   $[1] \leftarrow$  encode ( $\hat{Z}_N$ )
   $\Psi \leftarrow$  load-model ( $P_t$ )
   $\Psi \leftarrow \Psi.layers\_fc = false$ 
   $\Psi \leftarrow \Psi.add.Layers.Dense(2, Activation = SoftMax)$ 
   $\Psi \leftarrow \Psi.fit(x\_train, y\_train, iterations = l)$ 
   $\hat{Y} \leftarrow \Psi.predict(x\_test, y\_test)$ 
   $\{C_1: t_{s1}\} \{C_2: t_{s2}\}$ 
   $S_1 \leftarrow$  frame_to_video ( $C_1: t_{s1}$ )
   $S_2 \leftarrow$  frame_to_video ( $C_2: t_{s2}$ )
End
  
```

Initially, the frames are extracted from the original video ‘ V_l ’ and these extracted frames are appended to the list ‘ L ’. Thereafter, the frames ‘ f ’ in list ‘ L ’ are checked by applying face-mask-detector i.e., if frame ‘ f ’ is mask frame then append to list Φ_M otherwise append to list Φ_N . The pre-processing of with mask ‘ Φ_M ’ and without mask frames ‘ Φ_N ’ is done and stored in Φ'_M and Φ'_N . Afterwards labels are extracted from Φ'_M and Φ'_N and stored in \hat{Z}_M and \hat{Z}_N . The label encoding is used to

that whether the model ‘ Ψ ’ predicted the correct output or not. Thereafter, the two clusters C_1 and C_2 along with timestamps t_{s1} and t_{s2} generates the summary S_1 and S_2 by applying the VideoWriter function that converts frames to video.

Algorithm 2. Frames extraction ().

```

Input: Original video ‘ $V$ ’
Output:  $f = [f_1, f_2, f_3 \dots \dots f_n]$ 

Begin
   $I = []$ 
   $\dot{v} =$  VideoCapture ( $V$ )
  Test = True
   $C = 1$ 
  While (Test) do
    Test  $f \leftarrow \dot{v}.read()$ 
    if Test == True
       $I \leftarrow I.append(f)$ 
       $C = C + 1$ 
    else
      do nothing
    end while
  End
  
```

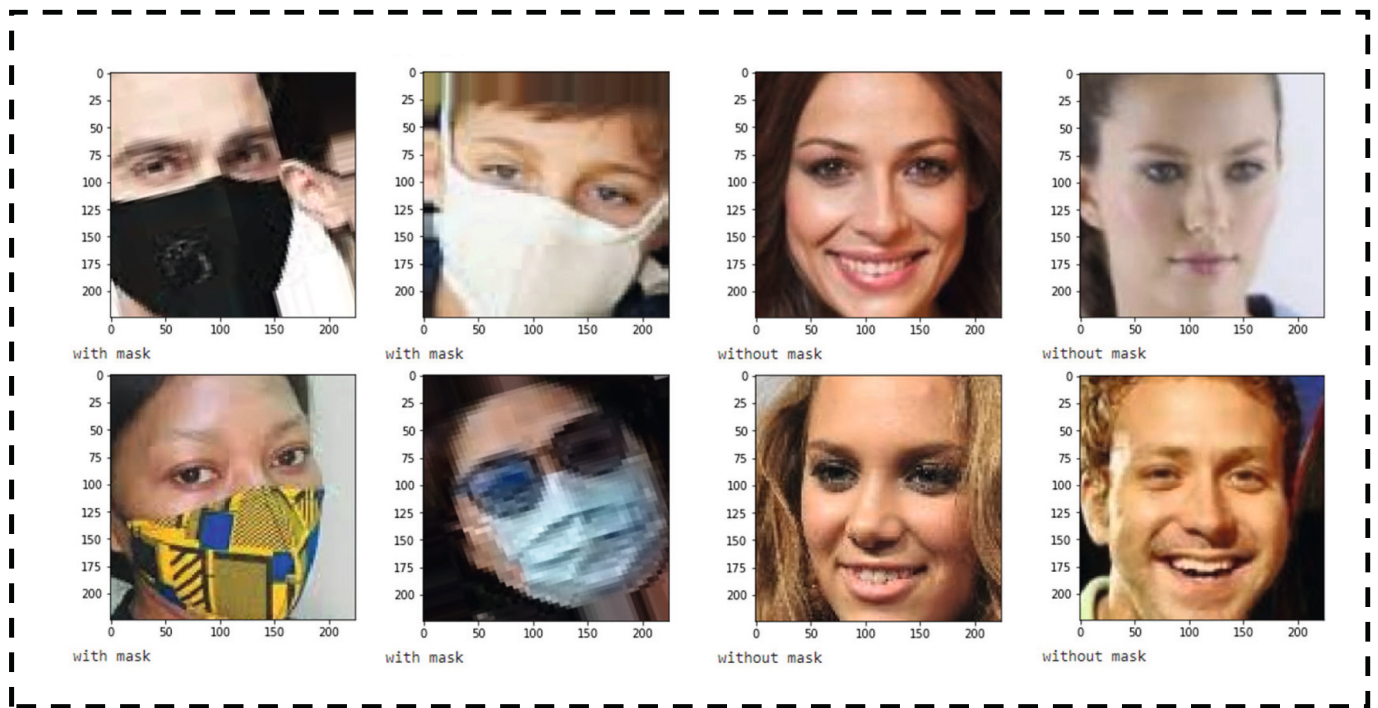


Fig. 8. Few random sample images of training dataset for learning CoSumNet model.

Table 2
The performance evaluation of CoSumNet on various hyperparameters.

| Trial no. | Batch size | Epochs | Units | Activation function | Optimizer | Learning rate | Dropout rate | Accuracy | Loss | Val-accuracy | Val-loss |
|-----------|------------|--------|-------|---------------------|-----------|---------------|--------------|---------------|----------------|---------------|----------------|
| 0000 | 100 | 70 | 416 | Softmax | RMSprop | 0.01 | 0.4 | 0.993125021 | 0.46391993 | 0.99650001 | 0.491533756 |
| 0001 | 40 | 90 | 128 | Sigmoid | Adam | 0.01 | 0.4 | 0.99237501 | 0.020583821 | 0.99750000 | 0.0081721 |
| 0002 | 20 | 70 | 288 | Tanh | SGD | 0.01 | 0.5 | 0.9956250190 | 0.0159838143 | 0.995000004 | 0.019898891 |
| 0003 | 60 | 100 | 448 | sigmoid | SGD | 0.001 | 0.3 | 0.986374974 | 0.0921769291 | 0.99400001 | 0.03324664 |
| 0004 | 80 | 100 | 352 | Sigmoid | RMSprop | 0.01 | 0.3 | 0.99075001 | 0.028082707 | 0.997500002 | 0.01260719 |
| 0005 | 30 | 60 | 96 | Sigmoid | Adam | 0.001 | 0.0 | 0.988125026 | 0.025259740 | 0.99699997 | 0.0150959631 |
| 0006 | 10 | 10 | 352 | Relu | Nadam | 0.01 | 0.5 | 0.99687498 | 0.0080511840 | 0.9965000152 | 0.018461737 |
| 0007 | 40 | 70 | 416 | Relu | SGD | 0.001 | 0.5 | 0.9958750009 | 0.01495882589 | 0.9975000023 | 0.0114982053 |
| 0008 | 40 | 10 | 96 | linear | SGD | 0.0001 | 0.3 | 0.9883750081 | 0.0396207235 | 0.9969999790 | 0.01204310636 |
| 0009 | 10 | 10 | 128 | tanh | Adam | 0.0001 | 0.4 | 0.9916250109 | 0.0239780042 | 0.9975000023 | 0.0083800777 |
| 0010 | 100 | 50 | 512 | tanh | Adam | 0.001 | 0.5 | 0.9980000257 | 0.0060474420 | 0.9975000023 | 0.0150552224 |
| 0011 | 20 | 70 | 512 | tanh | Adadelata | 0.001 | 0.0 | 0.9791250228 | 0.14507345855 | 0.9810000061 | 0.1381151527 |
| 0012 | 20 | 70 | 192 | sigmoid | Nadam | 0.01 | 0.0 | 0.9996250271 | 0.0017552506 | 0.99750000238 | 0.0107171405 |
| 0013 | 60 | 70 | 384 | softmax | Adadelata | 0.001 | 0.2 | 0.43112498521 | 0.69376713037 | 0.01099999994 | 0.69807255268 |
| 0014 | 60 | 10 | 64 | softmax | Adadelata | 0.0001 | 0.4 | 0.44150000810 | 0.70367795228 | 0.99759999238 | 0.6385808587 |
| 0015 | 100 | 50 | 416 | linear | SGD | 0.0001 | 0.3 | 0.9973750114 | 0.00859333295 | 0.99750000238 | 0.01076564751 |
| 0016 | 10 | 100 | 160 | softmax | RMSprop | 0.001 | 0.3 | 0.99812501668 | 0.28012663125 | 0.99650001525 | 0.31615498661 |
| 0017 | 80 | 100 | 384 | sigmoid | Nadam | 0.01 | 0.2 | 0.99449998140 | 0.01822070591 | 0.99650001525 | 0.01256569381 |
| 0018 | 40 | 100 | 448 | linear | RMSprop | 0.01 | 0.5 | 0.98587501049 | 0.12540563941 | 0.99699997901 | 0.03086781315 |
| 0019 | 80 | 10 | 64 | softmax | Adadelata | 0.01 | 0.3 | 0.62787497043 | 0.68803036212 | 0.87449997663 | 0.68458998203 |
| 0020 | 40 | 70 | 160 | softmax | Adam | 0.0001 | 0.1 | 0.99787497520 | 0.29425156116 | 0.99650001525 | 0.33429950475 |
| 0021 | 20 | 70 | 256 | tanh | Adam | 0.0001 | 0.3 | 0.99837499856 | 0.00409954413 | 0.99750000238 | 0.01237275544 |
| 0022 | 10 | 30 | 320 | linear | Nadam | 0.001 | 0.3 | 0.99075001478 | 0.04417301341 | 0.99650001525 | 0.02367977052 |
| 0023 | 80 | 100 | 352 | relu | Nadam | 0.01 | 0.1 | 0.99800002574 | 0.00698381755 | 0.99599999189 | 0.03181468322 |
| 0024 | 80 | 70 | 416 | tanh | Adam | 0.0001 | 0.5 | 0.99774998426 | 0.00526154320 | 0.99750000238 | 0.01320550963 |
| 0025 | 10 | 30 | 352 | linear | Nadam | 0.01 | 0.5 | 0.99512499570 | 0.03881689161 | 0.99550002813 | 0.05906426906 |
| 0026 | 10 | 70 | 448 | sigmoid | RMSprop | 0.001 | 0.0 | 0.99175000190 | 0.02545415237 | 0.99650001525 | 0.01455886196 |
| 0027 | 100 | 30 | 192 | linear | SGD | 0.0001 | 0.3 | 0.98912501335 | 0.03286636620 | 0.99650001525 | 0.011849628761 |
| 0028 | 60 | 10 | 448 | softmax | Adadelata | 0.01 | 0.5 | 0.54350000619 | 0.69263589382 | 0.92750000953 | 0.69002425670 |
| 0029 | 60 | 70 | 128 | tanh | Adam | 0.001 | 0.3 | 0.99787497520 | 0.00570006947 | 0.99699997901 | 0.01659318804 |
| 0030 | 40 | 10 | 480 | softmax | RMSprop | 0.01 | 0.2 | 0.99225002527 | 0.47251915931 | 0.99599999189 | 0.50032711029 |
| 0031 | 60 | 30 | 320 | relu | RMSprop | 0.01 | 0.3 | 0.99624997377 | 0.01784181781 | 0.99500000476 | 0.07296397536 |
| 0032 | 60 | 10 | 224 | tanh | Adam | 0.001 | 0.3 | 0.99112498760 | 0.02337403036 | 0.99250000715 | 0.03457861766 |
| 0033 | 100 | 70 | 160 | sigmoid | Nadam | 0.01 | 0.2 | 0.99849998950 | 0.00392116140 | 0.99750000238 | 0.00905037485 |
| 0034 | 60 | 10 | 64 | softmax | Adadelata | 0.0001 | 0.4 | 0.64275002479 | 0.67733192443 | 0.42399999499 | 0.69378501176 |
| 0035 | 40 | 90 | 128 | sigmoid | Adam | 0.01 | 0.4 | 0.98925000429 | 0.024343026801 | 0.99699997901 | 0.01468339934 |
| 0036 | 80 | 100 | 352 | sigmoid | RMSprop | 0.01 | 0.3 | 0.99225002527 | 0.02433313056 | 0.99699997901 | 0.01320157386 |
| 0037 | 40 | 70 | 416 | relu | SGD | 0.001 | 0.5 | 0.99550002813 | 0.01449323166 | 0.99650001525 | 0.01474225241 |
| 0038 | 10 | 10 | 128 | tanh | Adam | 0.0001 | 0.4 | 0.99162501096 | 0.020106781274 | 0.99699997901 | 0.01258400175 |
| 0039 | 100 | 50 | 512 | tanh | Adam | 0.001 | 0.5 | 0.99862498044 | 0.00359291164 | 0.99599999189 | 0.01905821822 |

(continued on next page)

Table 2 (continued)

| Trial no. | Batch size | Epochs | Units | Activation function | Optimizer | Learning rate | Dropout rate | Accuracy | Loss | Val-accuracy | Val-loss |
|-----------|------------|--------|-------|---------------------|-----------|---------------|--------------|----------------|---------------|----------------|----------------|
| 0040 | 20 | 70 | 192 | sigmoid | Nadam | 0.01 | 0.0 | 0.99299997091 | 0.02120191790 | 0.99699997901 | 0.01040361449 |
| 0041 | 100 | 50 | 416 | linear | SGD | 0.0001 | 0.3 | 0.99762499332 | 0.00685428734 | 0.99750000238 | 0.007510609459 |
| 0042 | 20 | 70 | 256 | tanh | Adam | 0.0001 | 0.3 | 0.99312502145 | 0.01714055985 | 0.99750000238 | 0.01688857376 |
| 0043 | 80 | 70 | 416 | tanh | Adam | 0.0001 | 0.5 | 0.98912501335 | 0.03256998211 | 0.99650001525 | 0.00971710495 |
| 0044 | 100 | 70 | 160 | sigmoid | Nadam | 0.01 | 0.2 | 0.99424999952 | 0.01894477568 | 0.99750000238 | 0.01195430755 |
| 0045 | 60 | 30 | 96 | sigmoid | Adam | 0.001 | 0.0 | 0.99000000953 | 0.02578973025 | 0.99699997901 | 0.01249769981 |
| 0046 | 100 | 50 | 416 | linear | SGD | 0.0001 | 0.3 | 0.99762499332 | 0.00779821584 | 0.99699997901 | 0.01404683198 |
| 0047 | 20 | 70 | 256 | tanh | Adam | 0.0001 | 0.3 | 0.99887502193 | 0.00337565620 | 0.99750000238 | 0.01615223102 |
| 0048 | 100 | 70 | 160 | sigmoid | Nadam | 0.01 | 0.2 | 0.99987500905 | 0.00051083037 | 0.99800002574 | 0.01699433475 |
| 0049 | 40 | 90 | 128 | sigmoid | Adam | 0.01 | 0.4 | 0.99900001287 | 0.00269682239 | 0.99800002574 | 0.00905867107 |
| 0050 | 100 | 70 | 160 | sigmoid | Nadam | 0.01 | 0.2 | 0.99312502145 | 0.02075749635 | 0.99800002574 | 0.00551498727 |
| 0051 | 40 | 90 | 128 | sigmoid | Adam | 0.01 | 0.4 | 0.99762499332 | 0.00559038296 | 0.99750000238 | 0.010341539047 |
| 0052 | 100 | 90 | 192 | relu | SGD | 0.001 | 0.5 | 0.99725002050 | 0.00867352169 | 0.99750000238 | 0.01019121147 |
| 0053 | 10 | 50 | 192 | linear | Adadelata | 0.01 | 0.4 | 0.90287500619 | 0.24673399329 | 0.96450001001 | 0.17362597584 |
| 0054 | 80 | 70 | 128 | relu | RMSprop | 0.001 | 0.3 | 0.99712502956 | 0.01154503319 | 0.99849998950 | 0.01019868999 |
| 0055 | 100 | 90 | 128 | tanh | Adadelata | 0.01 | 0.2 | 0.93924999237 | 0.24461148679 | 0.95399999618 | 0.25854030251 |
| 0056 | 40 | 30 | 320 | linear | SGD | 0.0001 | 0.0 | 0.99924999475 | 0.00246139499 | 0.99750000238 | 0.00962431635 |
| 0057 | 20 | 50 | 128 | linear | SGD | 0.01 | 0.4 | 0.99699997901 | 0.00976431369 | 0.9959999189 | 0.01776131056 |
| 0058 | 40 | 100 | 64 | sigmoid | SGD | 0.001 | 0.0 | 0.99787497520 | 0.01462342496 | 0.99550002813 | 0.02185641415 |
| 0059 | 100 | 10 | 128 | sigmoid | Adadelata | 0.001 | 0.2 | 0.96112501621 | 0.34548604488 | 0.93449997901 | 0.43978336453 |
| 0060 | 60 | 50 | 384 | linear | Adam | 0.01 | 0.4 | 0.98949998617 | 0.05898441374 | 0.99849998950 | 0.01144394278 |
| 0061 | 100 | 30 | 288 | relu | Adam | 0.001 | 0.1 | 0.99825000762 | 0.00607765838 | 0.99650001525 | 0.02240196429 |
| 0062 | 100 | 10 | 480 | linear | SGD | 0.0001 | 0.5 | 0.99512499570 | 0.01213162485 | 0.99599999189 | 0.01220408640 |
| 0063 | 20 | 10 | 512 | softplus | Adam | 0.001 | 0.2 | 0.99787497520 | 0.00629788264 | 0.99599999189 | 0.01175250671 |
| 0064 | 60 | 30 | 384 | softplus | RMSprop | 0.001 | 0.5 | 0.98487502336 | 0.20643137395 | 0.99750000238 | 0.02703285962 |
| 0065 | 40 | 30 | 224 | sigmoid | Adadelata | 0.001 | 0.3 | 0.91925001144 | 0.34768009185 | 0.94150000810 | 0.47824099659 |
| 0066 | 10 | 90 | 224 | relu | Nadam | 0.0001 | 0.3 | 0.99049997329 | 0.02616733685 | 0.99800002574 | 0.01088339369 |
| 0067 | 80 | 70 | 128 | relu | RMSprop | 0.001 | 0.3 | 0.73925000429 | 0.6918494105 | 0.47949999570 | 0.69314712285 |
| 0068 | 60 | 50 | 384 | linear | Adam | 0.01 | 0.4 | 0.99750000238 | 0.01080444920 | 0.99800002574 | 0.03180281072 |
| 0069 | 60 | 30 | 384 | softplus | RMSprop | 0.001 | 0.5 | 0.98937499523 | 0.04778867959 | 0.99849998950 | 0.01026585325 |
| 0070 | 10 | 90 | 224 | relu | Nadam | 0.0001 | 0.3 | 0.99150002002 | 0.02445608936 | 0.99800002574 | 0.01116688549 |
| 0071 | 100 | 90 | 192 | relu | SGD | 0.001 | 0.5 | 0.99699997901 | 0.01010649185 | 0.99750000238 | 0.01146344561 |
| 0072 | 80 | 70 | 128 | relu | RMSprop | 0.001 | 0.3 | 0.99774998426 | 0.01048607006 | 0.99849998950 | 0.00804141070 |
| 0073 | 60 | 30 | 384 | softplus | RMSprop | 0.001 | 0.5 | 0.99725002050 | 0.01082730572 | 0.99750000238 | 0.01936586014 |
| 0074 | 10 | 70 | 352 | linear | Adadelata | 0.001 | 0.2 | 0.99287497997 | 0.02770804986 | 0.99099999666 | 0.03427483886 |
| 0075 | 10 | 90 | 256 | softmax | Adadelata | 0.001 | 0.3 | 0.56887501478 | 0.69162487983 | 0.44100001454 | 0.69366991519 |
| 0076 | 60 | 100 | 384 | relu | Nadam | 0.0001 | 0.5 | 0.99699997901 | 0.01184887718 | 0.99750000238 | 0.01586479693 |
| 0077 | 20 | 10 | 288 | linear | Adam | 0.001 | 0.1 | 0.99787497520 | 0.01028930861 | 0.99849998950 | 0.00782436970 |
| 0078 | 80 | 10 | 224 | sigmoid | SGD | 0.001 | 0.3 | 0.99725002050 | 0.01098053716 | 0.99599999189 | 0.01502980943 |
| 0079 | 20 | 70 | 160 | tanh | SGD | 0.01 | 0.3 | 0.99637502431 | 0.01242144778 | 0.99699997901 | 0.01124484837 |
| 0080 | 10 | 90 | 384 | softplus | RMSprop | 0.001 | 0.3 | 0.99800002574 | 0.00988086126 | 0.99650001525 | 0.02425273135 |
| 0081 | 40 | 100 | 96 | softmax | SGD | 0.01 | 0.2 | 0.99800002574 | 0.06501466035 | 0.99650001525 | 0.08573645353 |
| 0082 | 20 | 10 | 288 | linear | Adam | 0.001 | 0.1 | 0.98799997568 | 0.10711442679 | 0.99800002574 | 0.02327713184 |
| 0083 | 60 | 100 | 384 | relu | Nadam | 0.0001 | 0.5 | 0.99837499856 | 0.00645826803 | 0.99750000238 | 0.02095812000 |
| 0084 | 20 | 70 | 160 | tanh | SGD | 0.01 | 0.3 | 0.99787497520 | 0.00702037755 | 0.99650001525 | 0.01156758423 |
| 0085 | 80 | 10 | 192 | softplus | RMSprop | 0.001 | 0.2 | 0.99849998950 | 0.00707628438 | 0.99900001287 | 0.00133153598 |
| 0086 | 40 | 100 | 32 | relu | Adadelata | 0.01 | 0.1 | 0.99087500572 | 0.03839541971 | 0.99000000953 | 0.05220552161 |
| 0087 | 80 | 50 | 320 | sigmoid | Adadelata | 0.01 | 0.4 | 0.38774999976 | 1.14498078823 | 0.99999128788 | 0.24693456292 |
| 0088 | 60 | 50 | 448 | tanh | Adadelata | 0.001 | 0.5 | 0.99437499046 | 0.02235530875 | 0.99299997091 | 0.02573654428 |
| 0089 | 40 | 100 | 128 | relu | Nadam | 0.001 | 0.3 | 0.99874997138 | 0.00353561015 | 0.99750000238 | 0.02200219407 |
| 0090 | 80 | 50 | 256 | tanh | SGD | 0.0001 | 0.2 | 0.99862498044 | 0.00709087122 | 0.99599999189 | 0.01359499897 |
| 0091 | 40 | 100 | 320 | sigmoid | SGD | 0.001 | 0.2 | 0.99612498283 | 0.01538820471 | 0.99500000476 | 0.01834912598 |
| 0092 | 60 | 70 | 352 | relu | Adam | 0.01 | 0.1 | 0.99075001478 | 0.02881800197 | 0.99849998950 | 0.00620197458 |
| 0093 | 40 | 30 | 320 | softmax | Nadam | 0.001 | 0.5 | 0.99750000238 | 0.16746371984 | 0.99650001525 | 0.19839082658 |
| 0094 | 80 | 90 | 448 | relu | Adadelata | 0.01 | 0.1 | 0.99225002527 | 0.06781533360 | 0.98799997568 | 0.07827771455 |
| 0095 | 10 | 100 | 160 | sigmoid | SGD | 0.0001 | 0.4 | 0.99587500095 | 0.02279310673 | 0.99550002813 | 0.01907427795 |
| 0096 | 10 | 10 | 96 | sigmoid | Nadam | 0.0001 | 0.0 | 0.99924999475 | 0.00184981222 | 0.99800002574 | 0.01036994066 |
| 0097 | 10 | 50 | 352 | sigmoid | Nadam | 0.01 | 0.0 | 0.99924999475 | 0.00367802148 | 0.99750000238 | 0.00890957936 |
| 0098 | 100 | 100 | 224 | tanh | RMSprop | 0.01 | 0.2 | 0.99087500572 | 0.03419919312 | 0.99650001525 | 0.01702136732 |
| 0099 | 60 | 90 | 384 | Sigmoid | Nadam | 0.001 | 0.1 | 0.99837499856 | 0.00462611112 | 0.99699997901 | 0.01155780535 |
| 0100 | 40 | 70 | 320 | tanh | Nadam | 0.01 | 0.2 | 0.99937498569 | 0.00156018149 | 0.99650001525 | 0.02188012003 |
| 0101 | 100 | 50 | 192 | Softmax | Adadelata | 0.001 | 0.3 | 0.626999974250 | 0.69016116857 | 0.97149997949 | 0.68385344743 |
| 0102 | 40 | 90 | 384 | relu | RMSprop | 0.01 | 0.3 | 0.98837500810 | 0.05198936536 | 0.99750000238 | 0.02010488510 |
| 0103 | 80 | 90 | 192 | tanh | Nadam | 0.0001 | 0.3 | 0.99900001287 | 0.00244123954 | 0.99849998950 | 0.01031357888 |
| 0104 | 100 | 100 | 256 | Softmax | RMSprop | 0.0001 | 0.5 | 0.99774998426 | 0.29207515716 | 0.99650001525 | 0.32633757591 |
| 0105 | 60 | 70 | 352 | relu | Adam | 0.01 | 0.1 | 0.99924999475 | 0.00254925177 | 0.99800002574 | 0.00936054345 |
| 0106 | 80 | 90 | 192 | tanh | Nadam | 0.0001 | 0.3 | 0.99349999427 | 0.01714254729 | 0.99750000238 | 0.00784064270 |
| 0107 | 10 | 10 | 96 | Sigmoid | Nadam | 0.0001 | 0.0 | 0.99924999475 | 0.00266717863 | 0.99750000238 | 0.01300043705 |
| 0108 | 10 | 50 | 352 | Sigmoid | Nadam | 0.01 | 0.0 | 0.99937498569 | 0.00228726188 | 0.998000025749 | 0.00993016175 |
| 0109 | 40 | 90 | 384 | relu | RMSprop | 0.01 | 0.3 | 0.98912501335 | 0.04849689453 | 0.99750000238 | 0.01548696681 |
| 0110 | 60 | 70 | 352 | relu | Adam | 0.01 | 0.1 | 0.99924999475 | 0.00419729528 | 0.99849998950 | 0.01151769515 |
| 0111 | 10 | 50 | 352 | Sigmoid | Nadam | 0.01 | 0.0 | 0.99874997138 | 0.00398347852 | 0.99800002574 | 0.01053883228. |

The **Algorithm 2** explains the process of frames extraction. It takes an original video ‘V’ as input and it gives output as extracted frames ‘f’ ranges from 1 to n. Initially an empty list ‘I’ is taken to store the frames extracted from the video and initialize the counter. The VideoCapture function is used to capture the video ‘V’ from a given path and while checking the condition it extracts frames one after another. These extracted frames are appended to the list ‘I’ and counter is incremented till it reaches the last frame of original video.

To enhance the quality of mask and non-mask images present in the video dataset the **Algorithm 3** is used for the pre-processing of these images. Initially, the algorithm takes Φ_M and Φ_N i.e., original frames as input and the output are Φ'_M, Φ'_N i.e., processed frames. The x-train and y-train are the training set of the dataset, and ‘ n_1 ’ and ‘ n_2 ’ be the size of mask and non-mask frames. For the two classes Φ_M , and Φ_N , firstly the resizing ‘ β ’ of image ‘ I_j ’ occurs that can resize all the images into a standard size and stores into ‘ r_j ’. Thereafter, by applying histogram normalization techniques the resize image ‘ r_j ’ stores as ‘ n_j ’. In addition, the normalization takes place by dividing all the pixels of images with the highest grey level i.e., 255 and gives output as enhance image ‘ q_j ’. Finally, the processed images ‘ q_j ’ are appended to x-train set and the label encoding of ‘ q_j ’ are stored into ‘ L_j ’ and afterwards these labels ‘ L_j ’ are stored into the y-train set. Finally, the x-train contains the processed mask and non-mask frames ‘ Φ'_M ’ and ‘ Φ'_N ’, whereas, y-trains contains the labels of ‘ Φ'_M ’ and ‘ Φ'_N ’.

Algorithm 3. Pre-processing ().

```

Input:  $\Phi_M, \Phi_N$ 
Output:  $\Phi'_M, \Phi'_N$ 
Begin
  x-train = [ ], y-train = [ ]
   $n_1 = |\Phi'_M|, n_2 = |\Phi'_N|$ 
  Class = [ $\Phi'_M, \Phi'_N$ ]
  for c in class do
    for j = 1 to  $n_1$  do
       $r_j \leftarrow \square(I_j)$ 
       $n_j \leftarrow \text{Hist\_norm}(r_j)$ 
       $q_j \leftarrow n_j/255$ 
      x-train  $\leftarrow$  x-train. Append ( $q_j$ )
       $L_j \leftarrow \text{label-encode}(q_j)$ 
      y-train  $\leftarrow$  y-train. Append ( $L_j$ )
    end for
   $\Phi'_M, \Phi'_N = [\text{x-train}], [\text{y-train}]$ 
  return  $\Phi'_M, \Phi'_N$ 
end for
End

```

To validate the CoSumNet model ‘M’ the **Algorithm 4** is proposed, wherein the input given to the model is ‘M’ and output is the performance of summarization in terms of ACC, precision, recall and F-score. Initially, the test video is given as an input to ‘M’, that can extract the frames and stored into an empty list ‘L’ and thereafter each frame is checked for face-mask-detector. If the frame belongs to mask class, then it is appended to ‘ Φ_M ’ and for non-mask it is appended to ‘ Φ_N ’. Afterwards, the pre-processing of frames takes place by applying the resizing, normalization, quality enhancement operation to get the enhanced

```

... Trial 110 Complete [00h 03m 18s]
val_accuracy: 0.9975000023841858

Best val_accuracy So Far: 1.0
Total elapsed time: 04h 48m 03s

Search: Running Trial #111

```

| Value | Best Value So Far | Hyperparameter |
|-------|-------------------|---------------------|
| 0.1 | 0.4 | dropout_rate |
| relu | softmax | activation |
| 60 | 60 | batch_size |
| 70 | 10 | epochs |
| 352 | 64 | units |
| 0.01 | 0.0001 | learning_rate |
| Adam | Adadelata | optimizer |
| 30 | 2 | tuner/epochs |
| 10 | 0 | tuner/initial_epoch |
| 2 | 3 | tuner/bracket |
| 2 | 0 | tuner/round |
| 0105 | None | tuner/trial_id |

Fig. 9. The optimal parameters selected after hyperparameter tuning of CoSumNet model.

image as ‘ Φ'_M ’ and ‘ Φ'_N ’. Then the labels are extracted from these processed images and stored as ‘ \hat{Z}_M ’ and ‘ \hat{Z}_N ’ along with encoding of 0 and 1. Finally the model ‘M’ is predicted on testing set and compute the performance by evaluating the actual summary S_{a1} and S_{a2} with that of predicted summaries i.e., S_{p1} and S_{p2} .

Suppose that a video ‘v’ of ‘k’ frames with size of each frame is $m \times n$. The overall complexity to process a video in our CoSumNet model is $O(kmn)$.

Algorithm 4. To validate the CoSumNet model.

```

Input: CoSumNet model 'M'
Output: Performance of summarization model (ACC, Precision, Recall, F-score)

Begin
  l ← |Vl|
  ΦM = [], ΦN = [], L = []

  for k = 1 to l do
    L ← L.append [frames_extraction-Vl(k)]
  end for

  for f in L do
    if face-mask-detector (f) = True
      ΦM ← ΦM.append(f)
    else
      ΦN ← ΦN.append(f)
    end for

    Φ'M ← pre-processing (ΦM)
    Φ'N ← pre-processing (ΦN)
    ẐM ← extract-label (Φ'M)
    ẐN ← extract-label (Φ'N)
    [0] ← encode (ẐM)
    [1] ← encode (ẐN)
    Y-pred ← M.predict (x-test, y-test)
    ACC, Precision, Recall, F-score ← Evaluate [[Sa1, Sa2], [Sp1, Sp2]]
  end
  
```

Table 3
The performance of CoSumNet at various search space.

| S.no. | Parameter | Search space | Performance | | | Selected parameter |
|-------|---------------------|--|-------------|----------|---------|--------------------|
| | | | Value | Accuracy | Loss | |
| 1. | Learning rate | [0.01, 0.001, 0.0001] | 0.01 | 0.90287 | 0.09713 | 0.0001 |
| | | | 0.001 | 0.91925 | 0.08075 | |
| | | | 0.0001 | 0.99837 | 0.00163 | |
| 2. | Epochs | [10, 30, 50, 70, 90, 100] | 10 | 0.96112 | 0.03888 | 90 |
| | | | 30 | 0.71325 | 0.28675 | |
| | | | 50 | 0.62699 | 0.37301 | |
| | | | 70 | 0.98637 | 0.01363 | |
| | | | 90 | 0.99312 | 0.00688 | |
| 3. | Optimizer | ['Adam', 'SGD', 'Nadam', 'RMSprop', 'Adadelta'] | Adam | 0.99825 | 0.00175 | Adam |
| | | | SGD | 0.98637 | 0.01363 | |
| | | | Nadam | 0.99041 | 0.00959 | |
| | | | RMSprop | 0.73925 | 0.26075 | |
| | | | Adadelta | 0.91925 | 0.08075 | |
| 4. | Batch size | [10, 20, 40, 60, 80, 100] | 10 | 0.90287 | 0.09713 | 60 |
| | | | 20 | 0.97912 | 0.02088 | |
| | | | 40 | 0.98837 | 0.01163 | |
| | | | 60 | 0.99112 | 0.00888 | |
| | | | 80 | 0.62787 | 0.07213 | |
| 5. | Activation function | ['softmax', 'softplus', 'relu', 'tanh', 'sigmoid', 'linear'] | Softmax | 0.99812 | 0.00188 | Softmax |
| | | | softplus | 0.98487 | 0.01513 | |
| | | | relu | 0.98837 | 0.01163 | |
| | | | tanh | 0.97912 | 0.02088 | |
| | | | sigmoid | 0.96112 | 0.03888 | |
| 6. | Dropout rate | [0.0, 0.1, 0.2, 0.3, 0.4, 0.5] | 0.0 | 0.97912 | 0.02088 | 0.4 |
| | | | 0.1 | 0.96212 | 0.03788 | |
| | | | 0.2 | 0.93924 | 0.06076 | |
| | | | 0.3 | 0.91925 | 0.08075 | |
| | | | 0.4 | 0.99312 | 0.00688 | |
| | | | 0.5 | 0.98912 | 0.01088 | |

* Highlighted values in Table 3 indicates the optimal hyperparameters for CoSumNet.

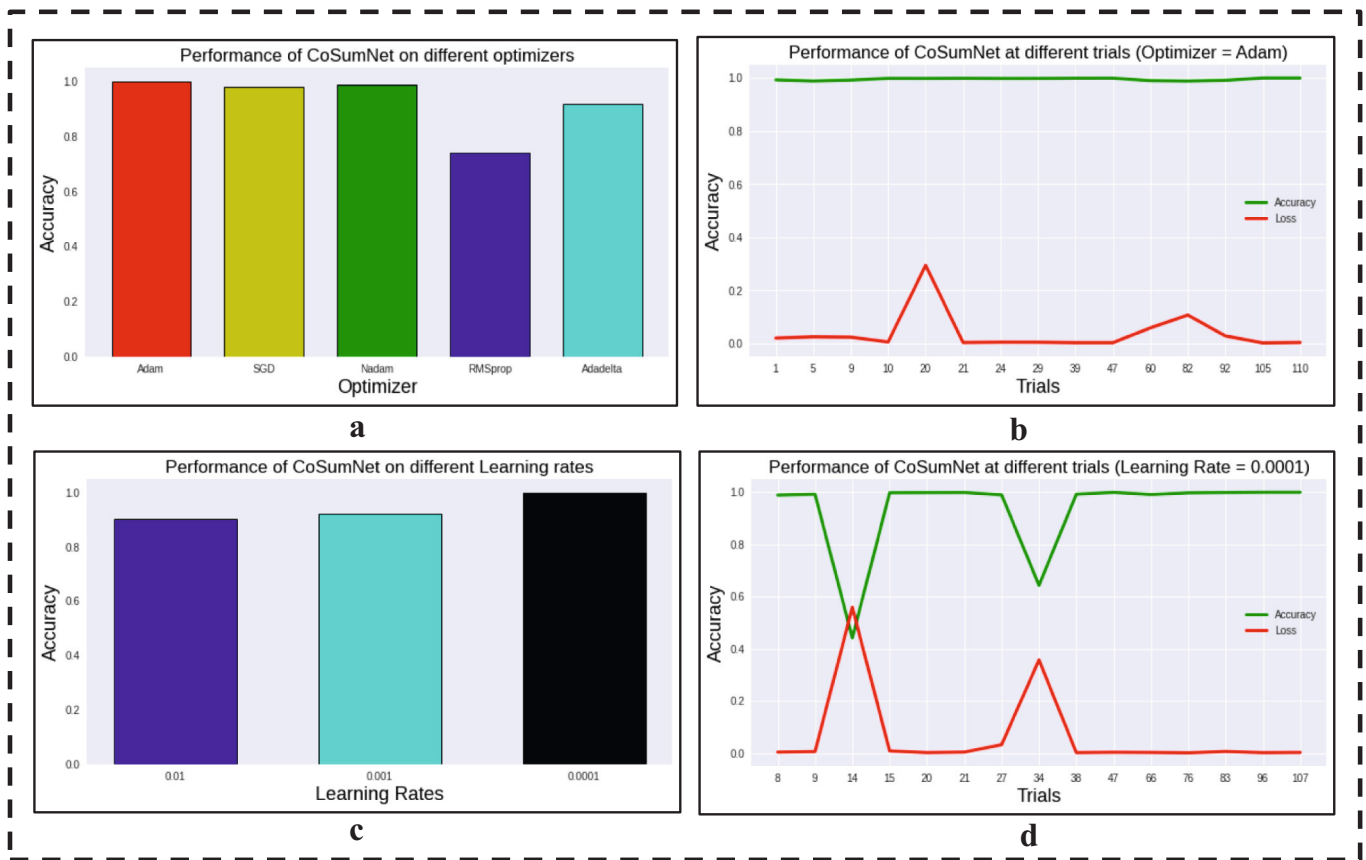


Fig. 10. Performance of CoSumNet model (a), (c). depicts model's accuracy at various values of optimizers and learning rate. (b), (d). shows performance of Adam optimizer and 0.0001 learning rate at different trails ranging frame 1 to 110.

4. Result analysis and discussions

In this section, we evaluate the performance of proposed CoSumNet on different experiments. Firstly, we present a description of benchmark dataset namely; Face Mask Detection ~12 K Images Dataset, that includes training, testing and validation set. Secondly, various experiments are carried out that include hyper tuning of proposed CoSumNet model along with the evaluation of model performance in known as well as known environment. Additionally, the proposed model is evaluated on cross-dataset scenario along with a variety of un-seen face masks. Lastly, the COVID-19 face mask violators summary generation testing is performed, where the separate summaries of with and without face masks are generated.

4.1. Benchmark face-mask-dataset

The face mask dataset contain the images of human with and without mask and is downloaded from Kaggle website named as "Face Mask Detection ~12K Images Dataset" [47]. The benchmark dataset that are used for our work contains 12,000 images captured through different video scenes covering diverse environments. The sample face masks images were acquired through a high-definition video camera. The images were bifurcated into two categories i.e., with mask and without mask images. Furthermore, the dataset consists of three subsets namely, training, testing and validation as shown in Table 1.

The samples of with and without face mask images present in dataset are shown in Fig. 8. As the face mask dataset constitutes of traing and testing set that includes with and without mask images of persons.

4.2. Evaluation metrics

To measure the performance of CoSumNet model, we choose standard metrics such as accuracy, precision, recall, f-score and error rate to evaluate our propose model.

Accuracy: It's the proportion of subjects who have been correctly identified to the total number of subjects. The accuracy may be computed by using Eq. (6).

$$Accuracy = \frac{True\ Positive + True\ Negative}{Total\ number\ of\ samples} \quad (6)$$

Precision: The number of true positives divided by the total number of positive predictions is known as precision. The precision may be calculated by using Eq. (7).

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (7)$$

Recall: The number of observations model accurately predicted out of the total number of observations is referred to as recall. The recall may be computed by using Eq. (8).

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (8)$$

F-score: It's the precision and recall's harmonic mean (average). The F1Score may be computed by Eq. (9).

$$F - score = 2 * \frac{precision * Recall}{Precision + Recall} \quad (9)$$

Error rate It is calculated as the number of all incorrect predictions divided by the total number of the dataset as depicted in Eq. (10).

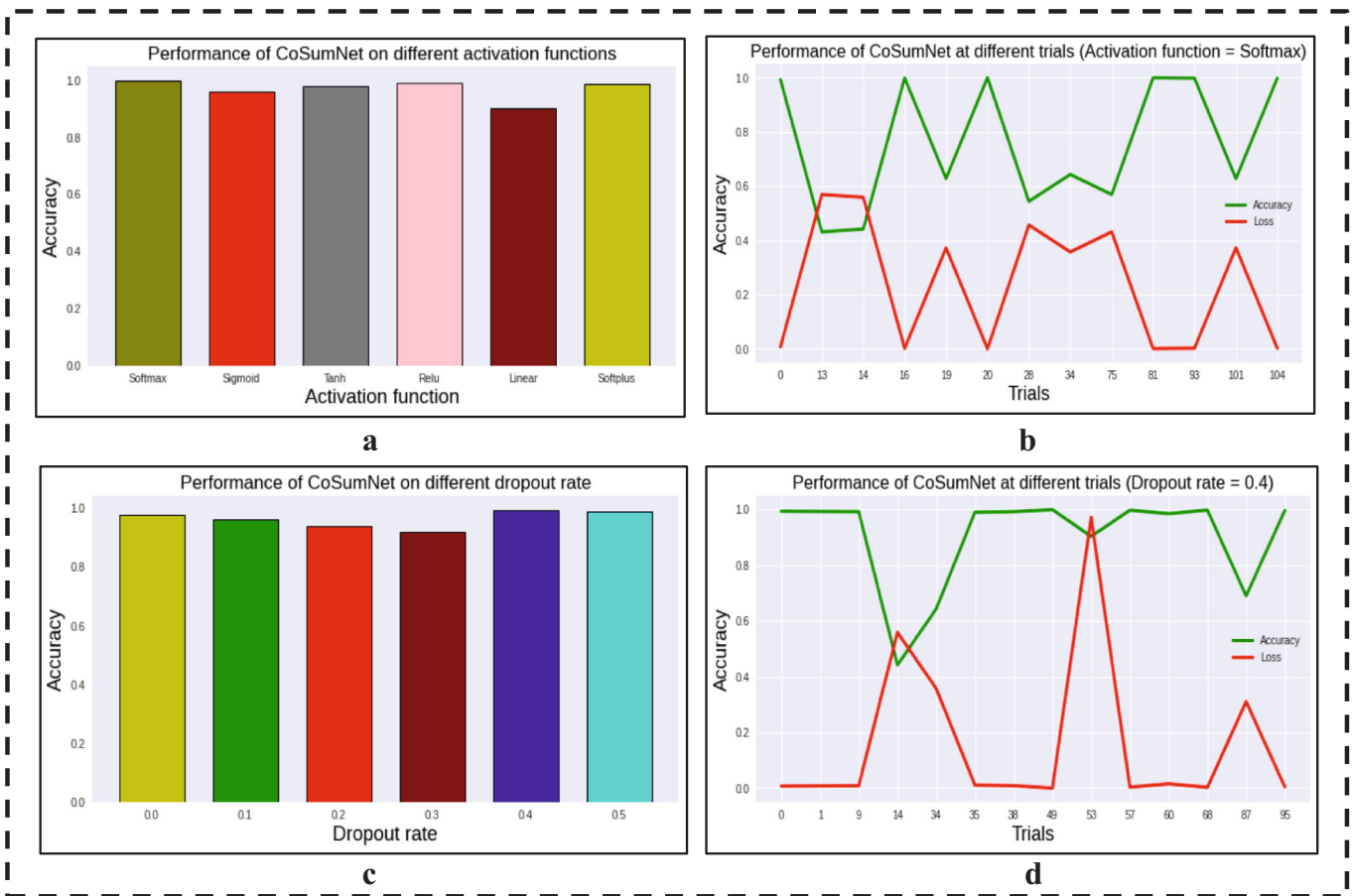


Fig. 11. Performance of CoSumNet model (a), (c). depicts model’s accuracy at various values of activation function and dropout rate. and (b), (d). shows performance of Softmax activation function and 0.4 dropout rate at different trails ranging frame 0 to 104.

$$Error = \frac{False\ Positive + False\ Negative}{Total\ number\ of\ samples} \tag{10}$$

4.3. Experimental results

In this section, the effectiveness of the CoSumNet model is analysed through a series of experiments.

4.3.1. Hyper parameter tuning of CoSumNet model

The CoSumNet is hyperparameter tuned to choose the optimal parameters of model so that it is ready for real-time testing. The various parameters include batch size, epochs, units, activation function, optimizer, learning rate, and dropout rate as shown in Table 2. The CoSumNet model is tuned via a large number of trials ranging from 1 to 110 and at each trial the model performance is evaluated based on training accuracy and loss along with validation accuracy and loss. In each trial the model is trained on the various hyperparameters and then the optimal parameters are selected, and thereafter, the selected parameters are used to test the CoSumNet model in an unknown environment.

Fig. 9 shows the highest performance of CoSumNet at trial 110 along with the optimal parameters. It can be observed that an accuracy of 99.9 % is achieved with the selected optimal parameters of the model as well as it takes 3 min and 18 s. The optimal values of various hyperparameters for the CoSumNet model include such as “Adadelata” optimizer, 0.0001 learning rate, 10 epochs, 60 batch size, SoftMax activation function, and 0.4 dropout rate.

The selection of optimal hyperparameters has a significant impact on the performance of proposed CoSumNet model. Table 3 depicts the

performance of each hyperparameters at different search space along with the value of optimal parameter selected so far. The CoSumNet model evaluated at various search space ranging from learning rate {0.01, 0.001, 0.0001}, optimizer {Adam, Nadam, SGD, RMSprop, Adadelata}, epochs {10,30,50,70,90,100}, batch size {10,20,40,60,80,100}, activation function {Relu, Softmax, Softplus, linear, Sigmoid, tanh}, and dropout rate {0.0, 0.1, 0.2, 0.3, 0.4, 0.5}. Finally, the model performs better on search space values such as, 0.0001 learning rate, 90 epochs, Adam optimizer, 60 batch size, softmax activation function and 0.4 dropout rate.

The analysis of CoSumNet model at various values of optimizers and learning rate is depicted in Fig. 10. It is clearly inferred from Fig. 10 (a) that the model exhibit promising results at “Adam” optimizer and Fig. 10 (b) depicts the performance of “Adam” in terms of accuracy and loss at different trials ranging from 1 to 110. Similarly, the Fig. 10 (c) infers that CoSumNet model indicates superior performance at a learning rate of 0.000, and whereas, Fig. 10 (d) depicts the accuracy and loss at a learning rate of 0.0001 with various trials ranging from 8 to 107.

The Fig. 11 depicts an analysis of CoSumNet model at various values of activation functions and dropout rates. It is clearly inferred from Fig. 11 (a), that the model performs better at “Softmax” activation function and Fig. 11 (b) depicts the performance of “Softmax” activation in terms of accuracy and loss with different trials ranging from 0 to 104. Similarly, the analysis from Fig. 11 (c) signifies that CoSumNet model shows better performance at a dropout rate of 0.4. Fig. 11 (d) depicts the accuracy and loss at a dropout rate of 0.4 with different trials ranging from 0 to 96.

The performance analysis of CoSumNet model at different epochs and batch sizes is shown in Fig. 12. It can be clearly seen from Fig. 12 (a) that

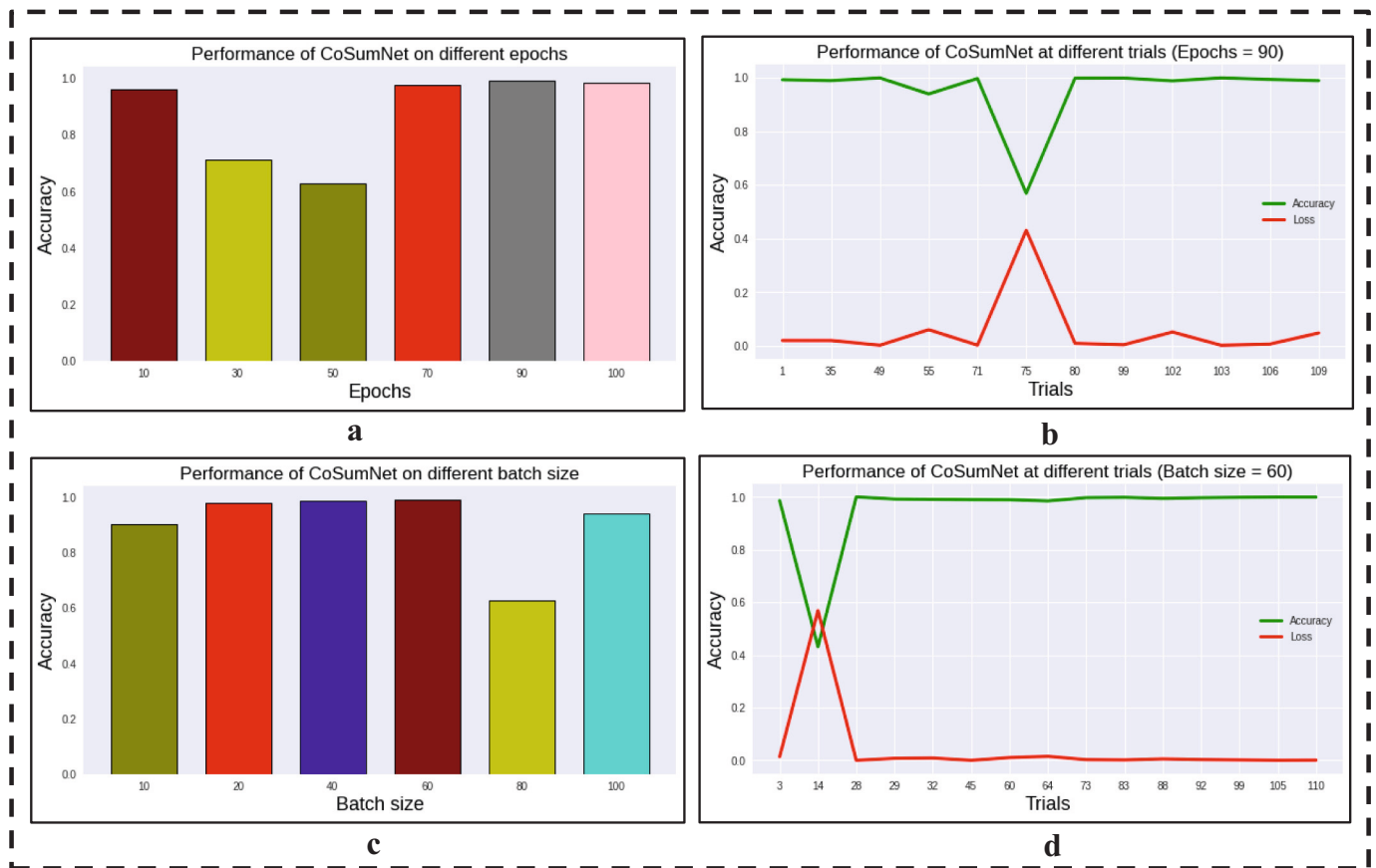


Fig. 12. Performance of CoSumNet model (a), (c). depicts model’s accuracy at various values of epochs and batch size. (b), (d). represents performance of 90 epochs and 60 batch size at different trials ranging frame 1 to 109.

the model performs better at 90 epochs and whereas, Fig. 12 (b) depicts the performance at an epoch of 90 in terms of accuracy and loss with different trials ranging from 1 to 109. Similarly, the Fig. 12 (c) depicts that CoSumNet exhibit better performance at a batch size of 60. On the

other hand, Fig. 12 (d) illustrates the accuracy and loss of at a batch size of 60 with different trials ranging from 3 to 110.

Table 4

An evaluation of CoSumNet model in known environment.

| Epochs | Loss | Accuracy | Validation loss | Validation accuracy | Execution time |
|--------|--------|----------|-----------------|---------------------|------------------|
| 3 | 0.0330 | 0.9948 | 0.0257 | 0.9975 | 17 s 108 ms/step |
| 5 | 0.0184 | 0.9966 | 0.0144 | 0.9987 | 16 s 106 ms/step |
| 10 | 0.0097 | 0.9984 | 0.0068 | 0.9987 | 17 s 107 ms/step |
| 15 | 0.0058 | 0.9986 | 0.0047 | 0.9953 | 17 s 108 ms/step |
| 20 | 0.0048 | 0.9984 | 0.0036 | 0.9946 | 17 s 110 ms/step |
| 25 | 0.0030 | 0.9994 | 0.0029 | 0.9971 | 17 s 110 ms/step |
| 30 | 0.0029 | 0.9992 | 0.0025 | 0.9975 | 17 s 108 ms/step |
| 35 | 0.0015 | 0.9996 | 0.0022 | 0.9978 | 17 s 110 ms/step |
| 40 | 0.0015 | 0.9985 | 0.0019 | 0.9981 | 17 s 108 ms/step |
| 45 | 0.0013 | 0.9998 | 0.0017 | 0.9983 | 17 s 111 ms/step |
| 50 | 0.0013 | 0.9987 | 0.0018 | 0.9982 | 17 s 110 ms/step |
| 55 | 0.0012 | 0.9988 | 0.0026 | 0.9987 | 17 s 108 ms/step |
| 60 | 0.0011 | 0.9989 | 0.0025 | 0.9987 | 17 s 109 ms/step |
| 65 | 0.0010 | 0.9990 | 0.0020 | 0.9987 | 17 s 109 ms/step |
| 70 | 0.0009 | 0.9991 | 0.0021 | 0.9987 | 17 s 110 ms/step |
| 75 | 0.0008 | 0.9992 | 0.0018 | 0.9987 | 17 s 108 ms/step |
| 80 | 0.0009 | 0.9991 | 0.0017 | 0.9987 | 17 s 108 ms/step |
| 85 | 0.0007 | 0.9993 | 0.0015 | 0.9987 | 17 s 110 ms/step |
| 90 | 0.0006 | 0.9994 | 0.0017 | 0.9987 | 17 s 111 ms/step |
| 95 | 0.0010 | 0.9990 | 0.0018 | 0.9987 | 17 s 108 ms/step |
| 100 | 0.0011 | 0.9989 | 0.0015 | 0.9987 | 17 s 109 ms/step |

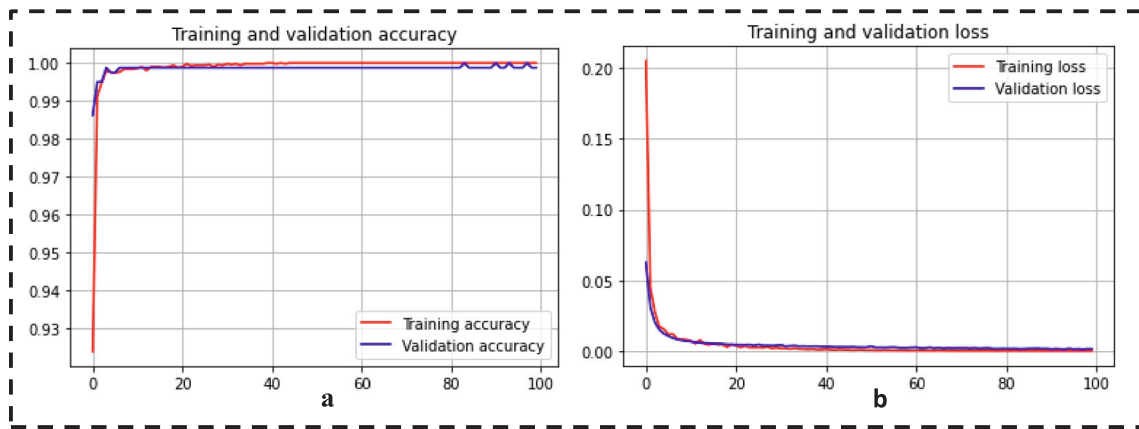


Fig. 13. The learning curves of CoSumNet model in known environment (a). validation accuracy (b) validation loss.



Fig. 14. A depiction of CoSumNet model performance in known samples.

Table 5
Optimal Hyperparameters used for testing data.

| Optimal parameters | Values |
|---------------------|---------------------------------|
| Optimizer | Adadelta |
| Learning rate | 0.0001 |
| Epochs | 10 |
| Steps per epoch | 156 |
| Batch size | 60 |
| Loss function | sparse_categorical_crossentropy |
| Activation function | Softmax |
| Dropout rate | 0.4 |

Table 6
Performance of CoSumNet model on testing dataset.

| Proposed CoSumNet model | | | | | |
|-------------------------|----------|-----------|---------|----------|---------|
| Classes | Accuracy | Precision | Recall | F-score | Loss |
| With mask (0) | 0.99923 | 0.98919 | 0.99791 | 0.993526 | 0.00077 |
| Without Mask (1) | 0.99923 | 0.98919 | 0.99791 | 0.993526 | 0.00077 |

4.3.2. Performance of CoSumNet in known environment

The performance of CoSumNet model evaluated on known samples is depicted in Table 4 and it is tested up to an epoch of 100. From the

analysis, it is clearly observed that the proposed model attains a superior performance at epoch 45, where an accuracy of 99.98 % is achieved with a loss of 0.0013 %.

The graphical representation of performance achieved in evaluation of the CoSumNet model is depicted in Fig. 13. The analysis from Fig. 13 (a) clearly infers that model results in better performance in the validation set. It achieves an accuracy of 99.98 % and a loss of 0.0013 % in known environment.

Few samples of with and without mask images of “face mask detection dataset” that are correctly predicted and classified by the CoSumNet model are shown in Fig. 14.

It is clearly revealed that CoSumNet model performed well in known environment. As all the known samples of “face mask detection dataset” are correctly detected as “with” or “without mask” by the proposed CoSumNet model.

4.3.3. Mask detection in unseen environment

The performance of CoSumNet model in unknown environment is evaluated in this experiment. Table 5 shows the values of optimal parameters that are used for the testing the unseen face images. In the unknown or unseen environment, the samples of with and without face mask images are used to evaluate the performance of CoSumNet model. These samples are the test images that are not present in the benchmark

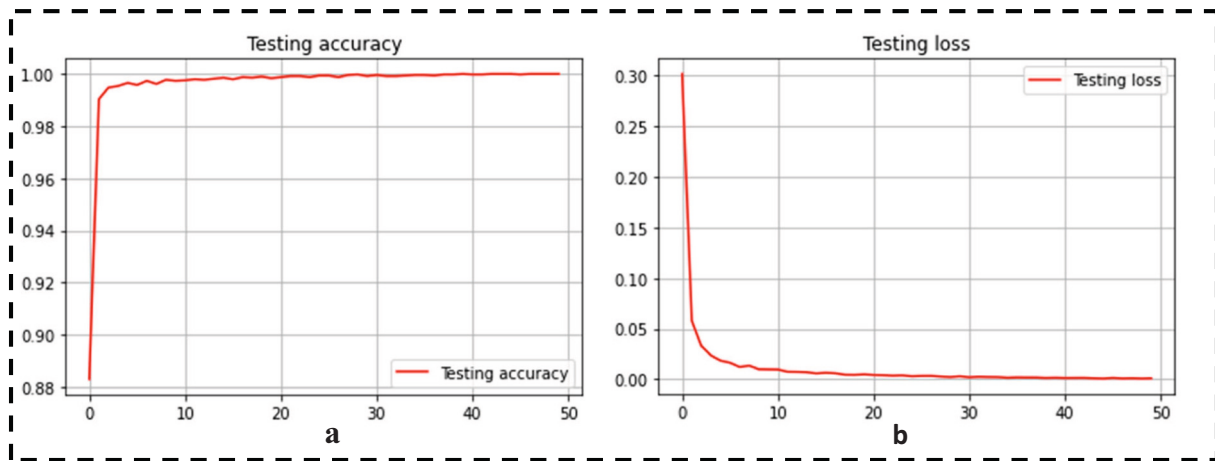


Fig. 15. The testing curve of CoSumNet model in unknown environment (a). testing accuracy (b). testing loss.

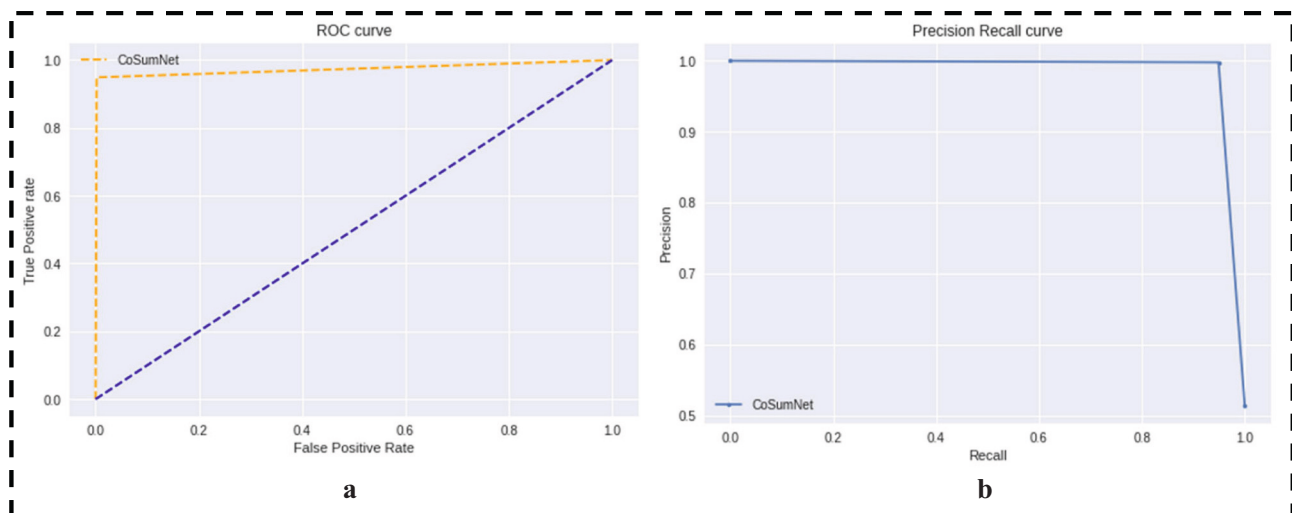


Fig. 16. Performance of CoSumNet model a. ROC curve b. Precision recall curve.

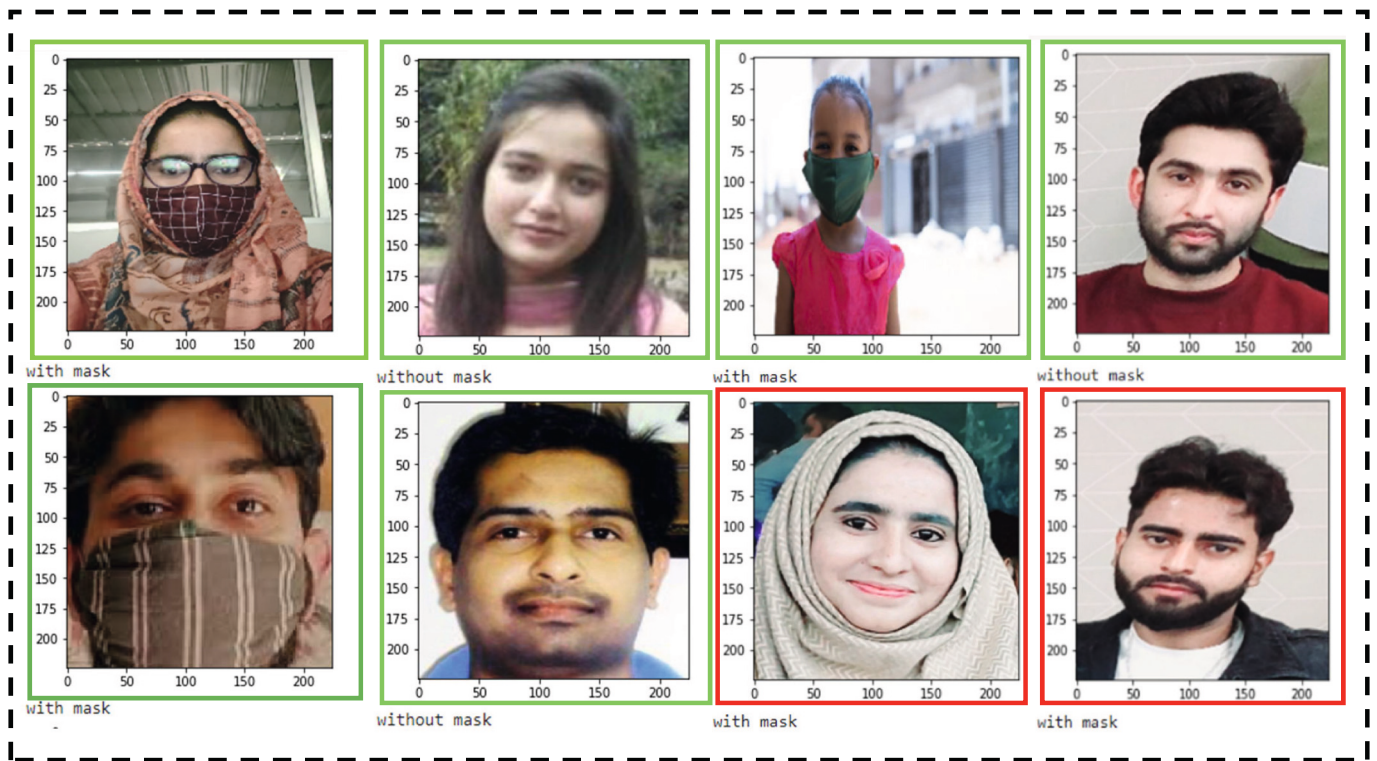


Fig. 17. A depiction of CoSumNet model performance in unknown samples.



Fig. 18. Performance evaluation of CoSumNet on a variety of face mask detection.

“face mask dataset” that is originally used for training the CoSumNet. The performance of CoSumNet model in the unknown or unseen environment is shown in Table 6. Furthermore, the various evaluation metrics are used such as accuracy, precision, recall, F-score and error

rate to predict the model performance. It clearly inferred that CoSumNet model performs well on testing dataset as an accuracy of 99.923 % and a loss of 0.00077 % is attained.

A graphical representation of model in terms of accuracy and loss is

Table 7
Comparison of proposed model with state-of-the-art face mask detection techniques.

| Year | Author(s) | Method | Performance evaluation | | | |
|------|-------------------------|--|------------------------|-----------|----------|------------|
| | | | Accuracy | Precision | Recall | Error rate |
| 2020 | Jiang M. et al. [40] | RetinaFaceMask based on MobileNetV1 | – | 82.3 % | 89.1 % | – |
| 2021 | Goyal H. et al. [34] | CNN-based face mask detection | 98.0 % | 97.0 % | 98.0 % | 0.02 |
| 2022 | Sethi S. et al. [35] | Ensemble face mask detection based on ResNet50 | 98.2 % | 98.92 % | 92.4 % | 0.018 |
| 2022 | Hussain D. et al. [38] | DCNN | 97.0 % | 98.0 % | 96.0 % | 0.02 |
| | | MobileNet-V2 | 98.0 % | 98.0 % | 98.0 % | 0.014 |
| 2022 | Jayaswal R. et al. [44] | Single Shot Multi-box detector based on Inception V3 | 98.75 % | 98.12 % | 97.45 % | 0.05 |
| 2023 | Our proposed model | CoSumNet | 99.923 % | 98.919 % | 99.791 % | 0.0007 |

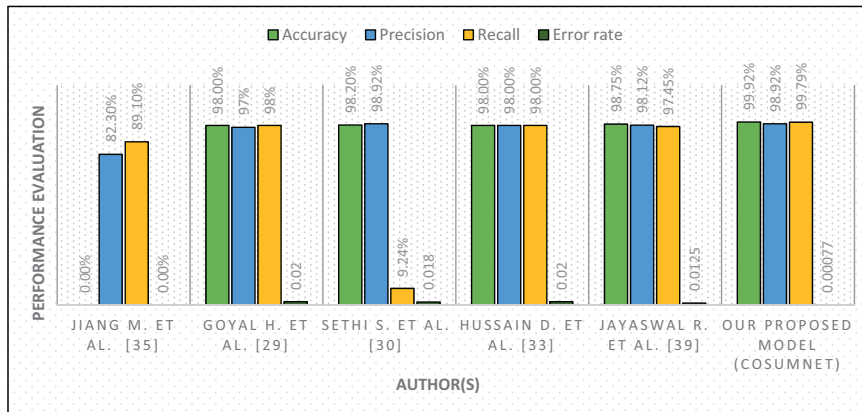


Fig. 19. A comparative analysis of our approach with existing similar SOTA face mask detection.

Table 8
Performance evaluation of our face mask detector with existing standard dataset.

| Year | Name | No. of images | | Performance | |
|------|---|---------------|--------------|-------------|-------|
| | | With mask | Without mask | Accuracy | Loss |
| 2020 | Face Mask Detection 7553 Images [48] | 3725 | 3828 | 97.731 % | 0.227 |
| 2020 | Face Mask Detection ~12 K Images Dataset [47] | 6000 | 6000 | 99.923 % | 0.001 |
| 2021 | Face Mask Detection [49] | 2994 | 2994 | 98.22 % | 0.018 |
| 2021 | Face Data Hybrid [50] | 22.6 K | 22.4 K | 97.102 % | 0.029 |

Table 9
The performance evaluation of CoSumNet model for summary generation.

| Original video | Length of video | No. of frames extracted | Scene | No. of selected frames | Summary length |
|----------------|-----------------|-------------------------|---------------|------------------------|----------------|
| Video 1 | 20 s | 600 frames | Indoor scene | 150 frames | 5 s |
| Video 2 | 1 min | 1800 frames | Outdoor scene | 450 frames | 15 s |
| Video 3 | 1 min 20 s | 2400 frames | Indoor scene | 600 frames | 20 s |
| Video 4 | 30 s | 900 frames | Outdoor scene | 240 frames | 8 s |
| Video 5 | 1 min | 1800 frames | Indoor scene | 450 frames | 15 s |

shown in Fig. 15 and depicts the accuracy of CoSumNet in unseen samples reaches up to 99.923 % as well as a loss up to 0.00077 %. It shows that our model results in a promising performance on testing dataset. Fig. 15 (a) highlight the performance in terms of accuracy on

testing set and Fig. 15 (b) depicts the testing loss of CoSumNet.

The performance of CoSumNet model in terms of ROC (i.e., receiver operating curve) is depicted in Fig. 16. The analysis from the figure clearly reveals that model performs well on testing dataset as the AUC (i.e., area under curve) of proposed model is 0.999. Fig. 16 (a) shows the ROC of CoSumNet, whereas Fig. 16 (b) present the precision recall curve of model.

The performance of CoSumNet model on testing images or unknown samples is depicted in Fig. 17. It is clearly inferred that majority of the images are correctly classified and predicted by the CoSumNet except last two samples that are bounded in red coloured box. The actual prediction of last two image is “without mask” whereas the model classifies these as “with mask”. Hence, the model demonstrates satisfactory performance on unknown samples.

Moreover, we have evaluated the effectiveness of our video face mask detector with a variety of unseen face masks that has not been seen by the model. The performance of CoSumNet for detecting few unknown face masks prediction is shown in Fig. 18.

It is clearly observed that in all the selected sample images the face mask is correctly predicted along with green bounding box. For evaluating the proposed model, we have specifically chosen a variety of face masks created with different fabrics.

4.3.4. Comparison with SOTA face mask detection techniques

In this section, our method for face mask detection employed in CoSumNet is compared with existing similar approaches. The comparative analysis is depicted in Table 7, which clearly exhibit that our approach outperforms similar methods in terms of performance particularly with an accuracy of 99.923 %. However, other similar methods have also achieved promising performance with different face masks, but our approach is robust in detecting a variety of face masks in the video frames with very small error rate of 0.0007. Moreover, our face mask detection technique in videos shows excellent performance both in known and unknown scenarios specifically in detecting face masks which has not been seen by the model. In addition, the technique used in

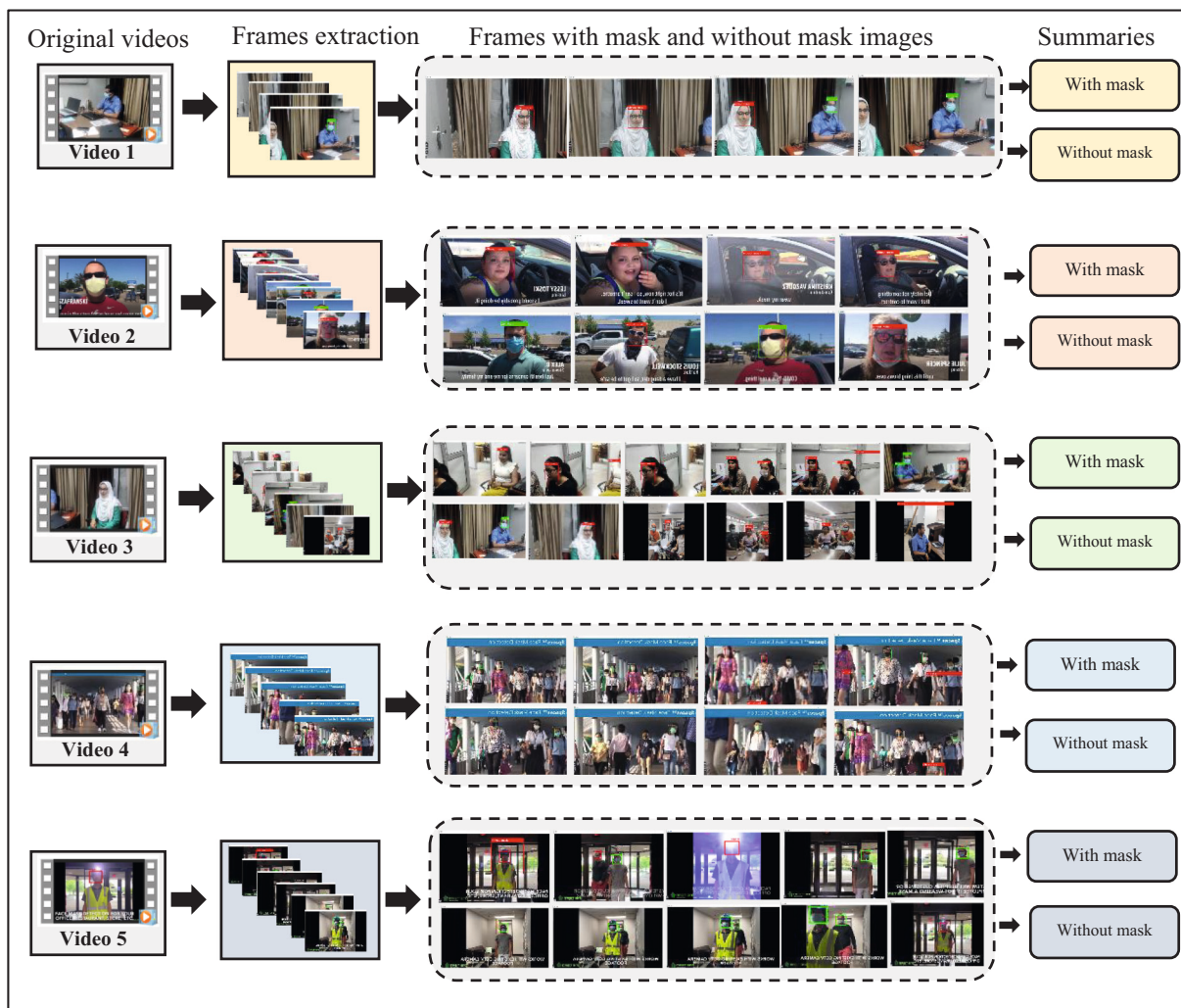


Fig. 20. The CoSumNet summaries generated from real-time video captured from crowded scenes.

our CoSumNet video summarization model demonstrates high precision and recall for intelligently detecting face masks with precision and recall of 98.919 % and 99.791 % respectively.

The comparative analysis of CoSumNet face mask detector with other similar methods in terms of various performance metrics is graphically illustrated in Fig. 19. The efficacy is clearly seen from Fig. 19 that indicates the superiority of the face mask detector.

Additionally, the face mask detection method employed in CoSumNet is evaluated on a set of publicly available standard face mask datasets and results are highlighted in Table 8. It can be observed that the CoSumNet face mask detector is also very effective and shows excellent performance in the cross-dataset environment.

4.3.5. CoSumNet for video summary generation

The CoSumNet model is used to evaluate and generate the summaries from the original videos and these summaries are depicted in Table 9. In addition, a detailed description of length of original video along with the number of frames extracted from that specific video is also presented. Moreover, the scene in which the video is captured and after the summarization process the selected frames as well as the length of summary is evaluated. The analysis from Table 9 clearly inferred that length of summarized video is far less than original video and it satisfies the criteria of video summarization. The frames that are present in summarized video are the detected and classified samples of with and without mask images. Hence, two summaries are generated namely, with mask and without mask summary.

The summary generation process of the original videos is depicted in Fig. 20. The analysis shows the description of each video, wherein firstly the frames are extracted from an original video and thereafter the CoSumNet model generates the samples of with and without face mask images.

In addition, the two summaries are generated that contain “with” and “without mask” human. The original videos were captured through CCTV installed in indoor or outdoor scenes to evaluate the performance of proposed CoSumNet model. The samples of “with mask” images are shown via green bounding box and, similarly, a red bounding box shows “without face” mask images. Finally, two summaries are generated from the model i.e., “with mask” and “without mask”, where the former includes “with mask” human video scene and the later contain “without mask” human.

The detailed experimental evaluation of the proposed CoSumNet model advocates its effectiveness in COVID-19 protocol monitoring via face mask detection. It offers several advantages in the prevailing conditions of COVID-19 pandemic, that include: (i). CoSumNet exhibits outstanding performance in both seen and unseen scenarios for generating respective short summaries with mask and non-mask categories. (ii). It is an efficient framework that employs the notion of transfer learning, therefore learning the model requires comparatively lesser training overhead. (iii). The proposed approach is robust even with smaller face mask dataset, it is mainly due to the usage of pre-trained ResNet-50 deep learning model. (iv). The real-time deployment of CoSumNet can help the controlling agencies to effectively monitor the

violation of face mask protocol, particularly in densely crowded public scenes.

However, one of the limitations of the CoSumNet is to cover large number of face masks that are manufactured through various fabrics in different regions of the world. Furthermore, appropriate wearing of face mask in the video frames is a further challenge that requisite more exploration in future.

5. Conclusions

In this work, we developed a novel and efficient video summarization-based model for COVID-19 protocol monitoring (i.e., without mask violators). Our approach has been evaluated in a diverse range of scenarios such as cross-dataset analysis, unseen face masks etc. In brief, the proposed technique is effective and efficient due to usage of a fine-tuned pretrained deep neural network (i.e. ResNet-50) for training the CoSumNet model. The efficacy of our approach can be observed in generating accurate short summaries that include human who violate the face mask protocols. Additionally, a benchmark dataset has been used for training the underlying classification model for CoSumNet that include a variety of human with and without face masks. Our approach has been evaluated on shorter length videos for monitoring COVID-19 protocols, therefore the future work may focus on longer videos. Though, the CoSumNet results in an accuracy of 99.92 % in unseen scenario, the further work may involve in enhancing the performance in large number of face masks datasets. Besides, the future work may include to upgrade the CoSumNet for generating automatic triggers (i.e., alerts, chalans) to COVID-19 protocols via video summaries of face mask violators in public places.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

References

- Uherbelau HGJ, Services H. Available. In: WHO's monthly operational update on COVID-19; 2022. p. 1–17. <https://www.who.int/publications/m/item/monthly-operational-update-on-covid-19-july-2022>.
- WHO. Driving impact in every country. Available. 2020. https://apps.who.int/gb/ebwha/pdf_files/WHA73/A73_24-en.pdf.
- WHO. Available. In: Weekly epidemiological update on COVID-19 - 29 June 2022 - Edition 98. Organization World Health; 2022. p. 16. <https://www.who.int/publications/m/item/weekly-epidemiological-update-on-covid-19-29-june-2022>.
- Sabha A, Selwal A. Data-driven enabled approaches for criteria-based video summarization: a comprehensive survey, taxonomy, and future directions. *Multimed Tools Appl* 2023. <https://doi.org/10.1007/s11042-023-14925-w>.
- Sabha A, Selwal A. HAVS: human action-based video summarization, taxonomy, challenges, and future perspectives. In: Proceedings of the 2021 IEEE international conference on innovative computing, intelligent communication and smart electrical systems. ICSES; 2021. p. 1–9. <https://doi.org/10.1109/ICSES52305.2021.9633804>.
- Choroš K. Categorization of sports video shots and scenes in tv sports news based on ball detection. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 8397. LNAI; 2014. p. 591–600. https://doi.org/10.1007/978-3-319-05476-6_60. no. PART 1.
- Javed A, Irtaza A, Khaliq Y, Malik H, Mahmood MT. Replay and key-events detection for sports video summarization using confined elliptical local ternary patterns and extreme learning machine. *Appl Intell* 2019;49(8):2899–917. <https://doi.org/10.1007/s10489-019-01410-x>.
- Hasan T, Boril H, Sangwan A, Hansen JHL. Multi-modal highlight generation for sports videos using an information-theoretic excitability measure. *EURASIP J Adv Signal Process* 2013;2013(1):1–17. <https://doi.org/10.1186/1687-6180-2013-173>.
- Liu C, Huang Q, Jiang S, Xing L, Ye Q, Gao W. A framework for flexible summarization of racquet sports video using multiple modalities. *Comput Vis Image Understand* 2009;113(3):415–24. <https://doi.org/10.1016/j.cviu.2008.08.002>.
- Baillie M, Jose JM. Audio-based event detection for sports video. In: Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics). 2728; 2003. p. 300–9. https://doi.org/10.1007/3-540-45113-7_30.
- Tabish M, Tanooli Zur R, Shaheen M. Activity recognition framework in sports videos. *Multimed Tools Appl* 2021. <https://doi.org/10.1007/s11042-021-10519-6>.
- Shih H. A survey on content-aware video analysis for sports. *IEEE Trans Circ Syst Vid Technol* 2017;99(9):1–20.
- Ali H, Sharif M, Yasmin M, Rehmani MH, Riaz F. A survey of feature extraction and fusion of deep learning for detection of abnormalities in video endoscopy of gastrointestinal-tract. *Artif Intell Rev* 2020;53(4):2635–707. <https://doi.org/10.1007/s10462-019-09743-2>.
- Münzer B, Schoeffmann K, Böszörményi L. Content-based processing and analysis of endoscopic images and videos: a survey. *Multimed Tools Appl* 2018;77(1):1323–62. <https://doi.org/10.1007/s11042-016-4219-z>.
- Feng W, Liu R, Zhu M. Fall detection for elderly person care in a vision-based home surveillance environment using a monocular camera. *Signal Image Video Process* 2014;8(6):1129–38. <https://doi.org/10.1007/s11760-014-0645-4>.
- Luna E, Miguel JCS, Ortego D, Martínez JM. Abandoned object detection in video-surveillance: survey and comparison. *Sensors (Switzerland)* 2018;18(12). <https://doi.org/10.3390/s18124290>.
- Beghdadi A, Asim M, Almaadeed N, Qureshi MA. Towards the design of smart video-surveillance system. In: 2018 NASA/ESA Conference on Adaptive Hardware and Systems (AHS); 2018. p. 162–7.
- Gowsikhaa D, Abirami S, Baskaran R. Automated human behavior analysis from surveillance videos: a survey. *Artif Intell Rev* 2014;42(4):747–65. <https://doi.org/10.1007/s10462-012-9341-3>.
- Sultani W, Chen C, Shah M. Real-world anomaly detection in surveillance videos. *IEEE Xplore* 2018;6479–88. <https://doi.org/10.48550/arXiv.1801.04264>.
- Sreenu G, Saleem Durai MA. Intelligent video surveillance: a review through deep learning techniques for crowd analysis. *J Big Data* 2019;6(1):1–27. <https://doi.org/10.1186/s40537-019-0212-5>.
- Xu J, Sun Z, Ma C. Crowd aware summarization of surveillance videos by deep reinforcement learning. *Multimed Tools Appl* 2021;80(4):6121–41. <https://doi.org/10.1007/s11042-020-09888-1>.
- Muhammad K, Ahmad J, Mehmood I, Rho S, Baik SW. Convolutional neural networks based fire detection in surveillance videos. *IEEE Access* 2018;6(March):18174–83. <https://doi.org/10.1109/ACCESS.2018.2812835>.
- Milotta FLM, Furnari A, Battiato S, Signorello G, Farinella GM. Egocentric visitors localization in natural sites. *J Vis Commun Image Represent* 2019;65(2). <https://doi.org/10.1016/j.jvcir.2019.102664>.
- del Molino AG, Tan C, Lim JH, Tan AH. Summarization of egocentric videos: a comprehensive survey. *IEEE Trans Hum Mach Syst* 2017;47(1):65–76. <https://doi.org/10.1109/THMS.2016.2623480>.
- Sahu A, Chowdhury AS. Multiscale summarization and action ranking in egocentric videos. *Pattern Recogn Lett* 2020;133:256–63. <https://doi.org/10.1016/j.patrec.2020.02.029>.
- Sahu A, Chowdhury AS. Summarizing egocentric videos using deep features and optimal clustering. *Neurocomputing* 2020;398:209–21. <https://doi.org/10.1016/j.neucom.2020.02.099>.
- Ghafoor HA, Javed A, Irtaza A, Dawood H, Dawood H, Banjar A. Egocentric video summarization based on people interaction using deep learning. *Math Probl Eng* 2018;2018:1–12. <https://doi.org/10.1155/2018/7586417>.
- WHO [Online]. Available. In: Considerations in adjusting public health and social measures in the context of COVID-19. World Health Organisation Interim Guidance; 2021. p. 1–13. <https://www.who.int/publications/i/item/consideration-s-in-adjusting-public-health-and-social-measures-in-the-context-of-covid-19-interim-guidance>.
- World Health Organization [Online]. Available. In: COVID-19 : occupational health and safety for health workers. World Health Organization; 2021. p. 1–16. https://www.who.int/publications/i/item/WHO-2019-nCoV-HCW_advice-2021.1.
- World Health Organization [Online]. Available. In: COVID-19 weekly epidemiological update. World Health Organization; 2022. p. 1–33. <https://www.who.int/publications/m/item/covid-19-weekly-epidemiological-update>.
- WHO, ILO. Preventing and mitigating COVID-19 at work [Online]. Available. In: Preventing and mitigating COVID-19 at work: policy brief, 19 May 2021, no. May; 2021. p. 1–21. <https://www.who.int/publications/i/item/WHO-2019-nCoV-workplace-actions-policy-brief-2021-1>.
- Hopkins J. Coronavirus Second Wave, Third Wave and Beyond: What Causes a COVID Surge. <https://www.hopkinsmedicine.org/health/conditions-and-diseases/coronavirus/first-and-second-waves-of-coronavirus>; 2021.
- Kapoor MSM. From 8,000 to 1 lakh in just 62 days; 2nd Covid-19 wave is more infectious than 1st. <https://www.businesstoday.in/coronavirus/story/from-8000-to-1-lakh-in-just-62-days-2nd-covid-19-wave-more-infectious-than-1s-292744-2021-04-06>; 2021.
- Goyal H, Sidana K, Singh C, Jain A, Jindal S. A real time face mask detection system using convolutional neural network. *Multimed Tools Appl* 2022. <https://doi.org/10.1007/s11042-022-12166-x>. no. November 2021.
- Sethi S, Kathuria M, Kaushik T. Face mask detection using deep learning: an approach to reduce risk of coronavirus spread. *J Biomed Inform* 2021;120(June):103848. <https://doi.org/10.1016/j.jbi.2021.103848>.

- [36] Li S, et al. Multi-angle head pose classification when wearing the mask for face recognition under the COVID-19 coronavirus epidemic. In: 2020 International Conference on High Performance Big Data and Intelligent Systems, HPBD and IS 2020; 2020. <https://doi.org/10.1109/HPBDIS49115.2020.9130585>.
- [37] Bosheng Q, Li D. Identifying facemask-wearing condition using image super-resolution with classification network to prevent COVID-19. *MDPI Sensor* 2020;20(18):1–23. <https://doi.org/10.3390/s20185236>.
- [38] Hussain D, Ismail M, Hussain I, Alroobaea R, Hussain S, Ullah SS. Face mask detection using deep convolutional neural network and MobileNetV2-based transfer learning. *Wirel Commun Mob Comput* 2022;2022:1–10. <https://doi.org/10.1155/2022/1536318>.
- [39] Nagavi TC, Sharanya RB, Minchu S, Kumar R, Manuprasad. COVID-19 face mask detection using deep learning. In: Proceedings of IEEE International Conference on Signal Processing, Computing and Control. 2021; 2021. p. 141–5. <https://doi.org/10.1109/ISPC53510.2021.9609487>.
- [40] Jiang M. Retinafacemask: A Face Mask Detector. *arXiv*; 2020. 2005.03950v2 [cs.CV] 8 Jun.
- [41] Fan X, Jiang M. RetinaFaceMask : a single stage face mask detector for assisting control of the COVID-19 pandemic. *IEEE SMC* 2021;2021. <https://doi.org/10.48550/arXiv.2005.03950>.
- [42] Wang Z, et al. Masked face recognition dataset and application. 2003.09093v2 [cs.CV]. In: *Computer Vision and Pattern Recognition*. 23. *arXiv*; 2020. p. 1–3. <https://doi.org/10.48550/arXiv.2003.09093>. Mar 202.
- [43] Jayaswal R, Dixit M. Traitement du signal detection of hidden facial surface masking in stored and real time captured images : a deep learning perspective in Covid time. *Traitement du Signal* 2021;38(6):1875–85. <https://doi.org/10.18280/ts.380632>.
- [44] Jayaswal R, Dixit M. AI-based face mask detection system: a straightforward proposition to fight with Covid-19 situation. *Multimed Tools Appl* 2022. <https://doi.org/10.1007/s11042-022-13697-z>.
- [45] Jayaswal R. A face mask detection system : an approach to fight with COVID-19 scenario. *Concurr Comput* 2022;34(28):1–19. <https://doi.org/10.1002/cpe.7394>.
- [46] SJ, He K, Zhang X, Ren S. Deep residual learning for image recognition. *IEEE Xplore* 2015;45(8):770–8. <https://doi.org/10.1109/CVPR.2016.90>.
- [47] Jangra A. "Face Mask Detection ~12K Images Dataset," Kaggle repository. <https://www.kaggle.com/datasets/ashishjangra27/face-mask-12k-images-dataset>; 2020.
- [48] Gurav O. "Face Mask Detection Dataset 7553 Images," Kaggle repository. <https://www.kaggle.com/datasets/omkargurav/face-mask-dataset>; 2020.
- [49] Kumar V. "Face Mask Detection," Kaggle repository. <https://www.kaggle.com/datasets/vijaykumar1799/face-mask-detection>; 2021.
- [50] Kaushik M. "facedatahybrid," Kaggle repository. <https://www.kaggle.com/datasets/mrviswamitrikaushik/facedatahybrid>; 2021.