AI-based face mask detection system: a straightforward proposition to fight with Covid-19 situation

Ruchi Jayaswal 1 · Manish Dixit 1

Abstract
The whole world is suffering from a novel coronavirus, which has become an epidemic. According to a World Health Organization report, this is a communicable disease, i.e., it transfers from an infected person to a healthy person. Therefore, wearing a mask is the most important precaution to protect from COVID-19. This paper presented a deep learning-based approach to design a Face Mask Detection framework to predict whether a person is wearing a mask or not. The proposed method uses a Single Shot Multibox detector as a face detector model and a deep Inception V3 architecture (SSDIV3) to extract the pertinent features of images and discriminate them in mask and without masks labels. Optimizing the SSDIV3 approach using different modeling parameters is a genuine contribution of this work. In addition to this, the system is tested and analyzed on VGG16, VGG19, Xception, Mobilenet V2 models at different modeling parameters. Furthermore, two synthesized novel Face Mask Datasets are introduced containing diversified masks (2d_printed, 3d_printed, handkerchief, transparent, natural-looking mask appearance masks) and unmask images of humans collected in outdoor and indoor environments such as parks, homes, laboratories. The experiment outcomes demonstrate that the proposed system has achieved an accuracy of 98% on the synthesized benchmark datasets, which comparatively outperforms other state-of-art methods and datasets in a real-time environment.

Keywords Computer vision · Covid-19 · CNN models · Face mask detection · Deep neural network · Face mask dataset
1 Introduction

Novel coronavirus has become a pandemic affecting many countries globally. According to the W.H.O. [39] [36] report, more than 216 countries have been affected by a coronavirus. Other names for it include Coronavirus, Novel Coronavirus, and Covid-19. This novel virus was first identified in December 2019 in Wuhan, China, and has resulted in an ongoing pandemic. As of 2 October 2021, more than 23.5-million cases have been stated globally, resulting in more than 4,806,841 deaths as per WHO say [39]. COVID-19 is an infectious disease caused by a respiratory illness. The common symptoms include fever, cough, fatigue, shortness of breath, sore throat, headache, nasal congestion, etc.

The droplets of infection fall onto the surface, and people may get infected by touching these contaminated surfaces and touching their faces, and it becomes community spread. Generally, it takes 1–14 days to identify the symptom of the virus. When people come into contact with others who is not exhibiting indications in the early stages, the disease can spread before symptoms occur. Though, the virus can be transferred if individuals won’t follow the precautionary measure. However, some vaccines were discovered, like Covishield, Phizer, Covaxin, Sputnik, etc., to prevent this virus, but there is still a need to take precautions after taking vaccine doses. WHO has recommended people should wear masks and follow social distancing rules in public places to minimize the chances of this deadly infection spreading [38]. These preventive measures can help in avoiding getting infected. Figure 1 represents the global situation w.r.t the number of cases increases without taking precautions or safety measures while decreasing cases if safety measures have been taken. As per the current situation seen, the third wave of covid-19 has been spread globally. Therefore, WHO or governments endorse people must wear a mask and maintain social distancing even after taking both shots of vaccine [36]. With a mask’s help, this coronavirus is not transferred to another person while others suffer from cold and cough problems. Moreover, many service providers allow customers to enter and use the service only if they wear masks.

Face mask detection application is not only restricted to avoid contamination from Covid 19, but it also enables the safety aspects for different areas or industries. There are multiple places where facemask can help people be safe and avoid any harm because of their work. Face mask detection application is beneficial and can be a great protection system for these sets of hazardous industries like Coal mines, chemical factories, leather industries, Gas plants, production industries for species or similar products, garbage dumping yards, etc. These hazardous industries released blasted toxins,
polluting materials, huge dust particles; therefore, to protect the workers from respiratory infections, the face mask is helpful in these areas. Further, this application is also useful in the hospital where transmissible infection risk is very high. It has also been observed that wearing a face mask comes under compliance policies. This application may help identify the non-wearable mask and help to improve compliance for the particular areas. Thus, the face mask detection application is a revolutionary change that may help for a greater cause in having a safe working environment and act as a precautionary measure for multiple contamination health issues. Face mask identification has become an important computer vision problem for assisting society, and basic research has been conducted in this area. However, it is still a hot topic due to this pandemic, raising the compulsory demand to wear face masks.

At present, limited face mask datasets are available on the internet. Due to this, the problem of face masks detection has not yet been adequately addressed. If people wear 3d printed masks, natural-looking appearance as human masks, then the detection system does not properly detect the mask or mask. Therefore, this turned up to be an astonishing problem for researchers. G.Jignesh [6] suggested a face mask detection using a deep learning model on Simulated Masked Face Dataset and achieved good accuracy, but a major drawback is artificially masked dataset is used, and a natural-looking appearance mask is not addressed. Similarly, Preeti et al. [25] and Shashi Yadav [40] gave their contribution on detecting face masks using MobilenetV2 model over Face mask Kaggle dataset [20]. Researchers have achieved maximum accuracy on artificially created and real images but do not consider diversified masks category images in the dataset.

As it concluded, two reasons were found in the existing Face Mask Detection model that brings the low achievement accuracy in real-time frames. First is the lack of relevant datasets with adequately masked faces, and second is that masks on the face induce a distortion that degrades the system’s performance. In this paper, the proposed methodology addresses these two issues and achieves good accuracy on synthesized and existing face mask datasets.

So far, comprehensive and inclusive diversified masks are lacking in the existing system. The presented article proposes an intelligent Face Mask Detection (FMD) system, which automatically predicts the various face masks in the real-time stream to fulfill existing work requirements. The system can produce a warning in a red boundary box on the screen when people do not wear masks. The proposed approach named SSDIV3 has been introduced in this paper for Face Mask Detection using deep neural networks. Moreover, the suggested model has been compared with existing deep learning models in terms of accuracy and execution time. The proposed work used various terminologies such as Tensorflow, Keras, Single Shot Detector, and Inception V3 architecture. SSDIV3 experiments competently differentiate images with left, right side, and frontal faces with masks and without masks. Face mask detection system refers to detecting whether a person wears a mask or not by marking a boundary box on the face [38]. This system has experimented on synthesized RealTime Face Mask Dataset (R.T.F.M.D) and RealTime Face Mask Dataset Version 2 (RTFMD-V2). Detecting a facemask in run-time becomes a challenging task, and this problem is addressed by applying deep learning techniques, which provide high accuracy in a different section. Later, the proposed study is compared with existing work on FMD and checks the system’s performance.

The key contributions of an extended and modernized version of the presented work are summarized as follows:

- The literature review of existing work on the FMD system has been presented along with its technicalities involvement, research findings, the dataset used, limitations and performance evaluation.
An intelligent FMD system is proposed to predict a person wearing a mask and no mask using deep neural networks.

Furthermore, self-synthesized face mask datasets are introduced that involve diversified human masks (3d and 2d printed, natural-looking mask appearance, N-95, surgical, handkerchief and transparent masks) and without masks faces. These visualized images are taken in different orientations, making this dataset more informative and affluent than the previously available face mask dataset, thereby outperforming the proposed system’s efficiency.

Subsequently, this article has extensively analyzed and compared the impact of various existing deep learning models and different modeling parameters for self-synthesized and existing face mask datasets.

Among all models, the suggested model, i.e., SSDIV3 deep learning approach, has achieved better performance in terms of system accuracy on synthesized and existing face mask datasets.

Lastly, the proposed model and datasets have been compared with existing face masks datasets and approaches concerning the accuracy of the framework.

1.1 Motivation and challenges

Face mask detection for 3d printed masks and natural-looking appearance masks become a new challenge for researchers in real-time. Because there is a lack of availability of a 3d printed, natural-looking human face mask dataset, no algorithms are designed to cover this issue in a real-time scenario. Thus, there is a need for a suitable FMD to predict mask faces in real-time frames. A reliable database and structured program are needed to make a robust FMD system in a pandemic scenario. The objective of this paper is to address these two problems. First, two new datasets are introduced, namely RTFMD and RTFMD-V2, and then a novel algorithm, namely SSDIV3, is proposed, which works very efficiently to train and validate the dataset. Furthermore, existing Face Mask datasets and the synthesized Face Masks datasets are compared with respect to accuracy and training time.

The organization of this paper is cut up into seven major sections as shown in Fig 2. Section 2: Related work will explain the existing work and its relating areas. Further, in Section 3, the proposed methodology of face mask detection will describe. Section 4: Material Required will cover in which the proposed dataset and existing dataset will summarize. Section 5: Experiment Setup, Performance measures, Experimental Outcomes, and Result analysis, Resultant images, and Comparative Analysis with the existing approach will cover, and the last Section 6: Conclusion will be discussed.

2 Related work

Mingjie Jiang et al. [20] introduced “RetinaFaceMask,” an approach for recognizing face masks that is both reliable and precise, followed by a novel “Context Attention Module” strategy for detecting masks on faces. The dataset of public face masks [20] was used. Furthermore, the authors propose a technique for removing cross-class objects that rejects both low confidence forecasts in addition to the union’s strong intersection. This method employed the pretrained deep learning models i.e. mobilenet and resnet to predict masks or not.
masks. When utilizing ResNet and RetinaFaceMask, the evaluation metrics i.e. Precision and Recall of the technique to predict masks are 82.30% and 89.12%, respectively, and 93.40% and 94.50% through MobileNet + proposed model.

Zhongyuan Wang et al. [37] proposed a mask face dataset for detecting face masks. The authors have proposed three significant Face Mask datasets, all three of them are freely available. This dataset is easily accessible to anyone, and it is available on Github. The “Masked Face Detection Dataset” (MFDD) is divided into two parts. Some samples are from related research, whereas others are crawled from the internet, and MFDD includes 24,771 mask images. The Real Mask Face Recognition Dataset is the second dataset (RMFRD). It has 5000 pictures of 525 people that wear face masks and 90,000 pictures of the same 525 persons without face masks. This is the massive real-world mask and no mask dataset, and the third one is the “Stimulated Mask Face Recognition Dataset” (SMFRD). They have applied face masks to previous public face datasets in this dataset. Moreover, five hundred thousand facial images of 1000 people have been carried in this repository.

The MAFA dataset, which contains more than thirty thousand internet face pictures and thirty-five thousand eight hundred six face masked images, was introduced by Shiming Ge et al. [12] to address the face detection problem. The pictures in this collection contain varied degrees of occlusion, and masks and orientations have overfilled at minimum one portion of each face. For face mask identification, LLE-CNNs have three primary modules also demonstrated. The first module combines two CNNs that have been pre-trained to take facial characteristics from input pictures and use them as a suitable descriptor. The Local Linear Embedding (LLE) approach is then used to turn these features vectors into matching-based attributes. The
terminologies were trained on a variety of regular synthetic, masked faces, and non-faces in the embedding module. Moreover, the verification module integrates classification and regression tasks to identify the candidate’s facial regions using a unified CNN. Surprisingly, the MAFA dataset outperforms six arts states from our proposed technique by at least 15.6%.

Jiankang Deng et al. [9] RetinaFace is a single stage face detector that performs pixel-by-pixel face detection and employs extra supervised and self-supervised multi-task learning. The paper is partitioned into five main sections. In the first module, he manually labels five facial landmarks on the WIDER Face dataset and observes an improvement in hard face detection when introducing an additional supervision signal. The second module includes a self-supervised mesh decoder branch that predicts pixel-by-pixel 3D shape face information. Subsequently, at the third stage of the WIDER face hard test set, RetinaFace improved the state-of-the-art average precision by 1.1%. RetinaFace enables state-of-the-art methods to enhance the outcomes of their face verifications i.e., TAR = 89.58% for FAR = 1e-6. Consequently, in the last step, the RetinaFace could operate real-time on a CPU core for a quality picture utilizing pretty light-weight backhaul networks.

Faen Zhang et al. [43] reported “AInnoFace,” a successful face detection method. To obtain remarkable outcomes, the method begins with RetinaNet and is then supplemented with a few tactics. They employed the loss function Intersection-over-Union for regression, two-steps classification and regression to detect the face. The authors used the max-out operation and the multi-scale testing approach to categorize the data. As an outcome, our method enhances face detection performance when utilizing the WIDER face dataset.

A. Bastanfard et al. [2] proposed a facial rejuvenation in images of adults. Geometric points have been explored, such as skin texture, differentiating between the aged and the young. Anthropometrics body measurement technique is used to collect data on body changes. Furthermore, these procedures are wrapped and mapped to any person’s characteristics to achieve more exact face rejuvenation. This approach is simple to use because its calculation is straightforward. Furthermore, the proposed approach’s time complexity is efficient. Moreover, the testing is performed on a single image rather than a group of images.

Azam Bastanfard and Hiroki Takahashi [3] proposed a web-based appearance system that predicts the stimulating effects of facial images having variant appearances. The anthropometric idea was used to locate the landmarks on the face, and the geometric landmarks changed with age. Furthermore, the face muscles change in response to variations in facial expression. Human hair was gathered, and after considering all parameters, an efficient procedure for facial aging was presented. The suggested methodology has good time complexity. Below Table 1 presented the literature review of past researches in face mask detection field.

### 3 Proposed methodology

This article proposes an SSDIV3 network to detect a face mask in a real-time environment. The system predicts whether people have worn a mask properly with the help of the proposed methodology. The flow diagram of the proposed approach is illustrated in Fig. 3. Broadly, the methodology is segmented into two phases: Training and Testing. The proposed work follows a four-step framework: image acquisition, image pre-processing, training phase, and testing phase.

The initial stage is to capture the real-time images using a camera. Here, the captured images represent appearance information in RGB format. In the next step, the images are pre-processed and then apply image augmentation technique to improve the system’s accuracy. Image datasets are then
## Table 1 Review of the existing face mask detection techniques

| Authors                | Technicalities involved                                      | Dataset Used                                      | Research findings                                                                 | Limitations                                                                 | Performance                  |
|------------------------|--------------------------------------------------------------|---------------------------------------------------|----------------------------------------------------------------------------------|------------------------------------------------------------------------------|------------------------------|
| Preeti Nagrath et al.  | MobileNetV2, Fine tuning, SSD                                | Medical Face Mask Dataset and Face Mask Dataset    | SSD used as a face detector and MobilenetV2 as a classifier.                      | More different datasets can be used to consider facial landmarks and facial part detection process. | Achieved accuracy of 92.64 and F1 score of 93% |
| Mingjie Jiang et al.   | RetinaFaceMask and MobileNet, RetinaFaceMask and ResNet     | Wider Face and Masked Faces are combined and made Face Mask Dataset. | Transfer Learning and attention mechanism are used to predict the mask images.  | Normal masks like 2d printed, N-95 masks are only considered.               | Achieved accuracy upto 93.49% |
| Sunil Singh et al.     | YOLOv3 and Faster-RCNN                                       | MAFA, Wider Face and manually collected face masks images | YOLOv3 produced better results compared with Faster-RCNN for face mask detection. | Sensitive for spatial location camera, artificial mask and two masks are considered. | Achieved precision of YOLOv3 and Faster R-CNN model is 55 and 62, respectively. |
| Susanto et al.         | YoloV4                                                       | Surgical and fabric face mask images               | YOLOv4 is used for face mask detection                                          | Different 3d masks and other 2d masks are not considered.                    | Average FPS is nearly about 11.1 |
| Shilpa Sethi et al.    | ResNet50, AlexNet and MobileNet                              | Self-synthesized unbiased face mask dataset       | ResNet50 is used as a proposed framework, and a transfer learning approach is applied. | It is not integrated into high-resolution video surveillance devices.         | Achieved accuracy upto 98.2%  |
| Arjya Das et al.       | CNN model                                                   | Face Mask Dataset and Medical Mask Dataset         | Pre-trained CNN model is applied in which two 2D convolution layers are used.   | Moving images didn’t predict mask or no mask accurately.                     | Achieved Accuracy on first dataset 1 is 95.77 and 94.58% on dataset 2.       |
| G K Jakir Hussain et al.| CNN model, (SVM and Symbolic Classifiers)                    | Taking 50 several image datasets                  | CNN model is applied to predict mask or no masks, further check human body temperature, and then use Arduino controller to interrupt the people. | Train the dataset tiny size dataset, which is 50 and not taken different types of masks. | Achieved accuracy using proposed CNN is 91.11% |
| Walid Hariri [15]      | VGG-16 model, Bag of features and Multilayer Perceptron(MLP)| Real World Masked Face Dataset                     | VGG-16 model is taken as feature extractor then the mapping of features is done by Bag of features. Further, the classification process is done by MLP classifier. | Mask dataset is simulated i.e. artificially created.                        | Achieved high accuracy of recognition of face mask images is 91.3% |
divided into training and validation batches of images. The third stage is to train the model over the synthesized (RTFMD and RTFMD-V2) dataset and existing face mask dataset (RFMD and Face mask dataset) [37] dataset. Briefly, detail about the self-synthesized and existing dataset have been discussed in sections 4.1 and 4.2. A deep transfer learning technique is applied in the InceptionV3 technique to gain the useful features of Face mask datasets. The features are learned using the selected model. Later, proposed method is fine-tuned according to the need of model, as well as the accuracies and trained time achieved for the validating and training sets.

At the testing stage, deploy Single Shot Multibox Detector (SSD), the model of face detector model predicts and assigns the face inside the frame and shows the face with the help of the boundary box. Additionally, the face mask trained model was acquired during the learning phase and deployed at the testing or run time phase. A Face Detection [1, 23] model is essential to observe faces. Finally, with the help of the obtained model, the system will predict ‘wearing mask’ or ‘no masked’ faces by simply drawing a box around the border. This paper aims to achieve higher accuracy of the FMD system in minimum time and be robust on all datasets. Different existing models are compared with the suggested approach, and relevant hyperparameters are modified to check the results. We have two models for the SSDIV3 approach, the Caffe model and the InceptionV3 model, for extracting the features and classifying the mask or no mask face image. The pseudo-code is presented in a step-wise manner, which shows the workflow of the proposed model.

The algorithm or pseudo-code of the FMD is described in a step-wise manner for the training and testing stage.

**Phase 1: A Training stage- Pre-processing and Training on datasets**

**Input:** Load Face Mask Datasets, picture size (a × b × c) in RGB format.

**Output:** Face mask detector model (.h5 format).

- Step 1. Load the images of face mask datasets from disk.
- Step 2: Pre-process the images, i.e., resizing normalization.
- Step 3: Apply the Image Augmentation method and then divide images into training and testing batches using function.
- Step 4. Apply a deep Inception V3 model.
- Step 5. Modifying the hyperparameters for the model.
- Step 6. Acquired and saved the FMD model (.h5) and checked metrics of trained model and computation time.

**Phase 2: A Testing stage- Deployment of face detector and face mask detector model**

**Input:** Load the images from the real-time image/video stream.

**Output:** Determine a person wearing “mask” or “no mask” marked by a boundary box.

- Step 1. Acquired images from the real-time video stream.
- Step 2: Pre-process the image i.e. resizing (a × b × c), normalization.
- Step 3. Apply the predefined face-detector model to easily frame faces in an image, i.e., Caffe prototype model.
  - If face is spotted from the load image or from run time video:
    - then, draw the rectangle box around the cropped face.
- Step 4. Apply obtained face mask detector model - .h5 and show evaluation metric;
  - If a human wear a face mask, show on the rectangular box ‘With a mask.’
  - Else show on the rectangular box “No mask”.
- Step 5: Repeat Steps 1 to 4 until q is pressed.
3.1 Pre-processing

Pre-processing of the dataset is the primary step of methodology in which the image dataset is resized according to the proposed model. The synthesized dataset is collected from different sources such as mobile phones and webcams; therefore, image resizing is essential for all images to gain similarity. Additionally, the existing dataset contains many repetitive images, corrupt images, so data cleansing is done manually and resizes them at a particular resolution. A brief description of the dataset will discuss in sections 4.1, 4.1.2, and 4.2. Furthermore, partition the dataset into train and test sets of images using splitting function into 80% and 20%. Pre-processing is a function that takes the dataset folder as input, loads all of the images.
from it, and resizes the files using the SSDIV3 model. The image files are transformed to a NumPy array after normalizing and resizing the list of photos for easier and quicker computation.

Furthermore, the Image Augmentation technique is applied to improve the learning model of the accuracy. A huge amount of data is needed for the deep learning approaches to perform training for better precision. In this projected framework, the SSDIV3 model is trained after the image augmentation of the dataset. RTFMD and RTFMD-V2 datasets are inadequate for training purposes, due to which augmentation technique is beneficial for achieving good accuracy. Rotation, Flipping, Shearing, Shifting, Brightness, and Zooming methods can be done in augmentation technique to increase the dataset’s size. This technique is applied to better results in both synthesized and exiting face mask datasets. For data/image augmentation, the image data creation function is employed, which also produces train and validation batches of images.

3.2 Face detector model

Face detection is one of the crucial steps of this architecture. The critical stage of the FMD system is to frame the faces from test images or detect faces in real-time. At the testing phase, when the person’s face comes in front of the surveillance camera, it becomes necessary to detect the face first and then check the person having a mask or without a mask. So, the Caffe model is used to detect the face. This model is based on Single Shot-Multibox Detector (SSD) [10], and it follows ResNet-10 architecture as its backbone. Caffe model comes in a deep neural network module that was introduced post OpenCV 3.3. Two files are loaded to detect the face

i) Caffemodel
ii) Prototxt files.

After detecting the face using the Caffe detector model using cv2.dnn.readNet("path/to/prototxtfile","path/to/caffe-weights"), the faces are detected by making rectangular boxes on the face. Using the Caffe detector, real-time frames faces are detected even when faces are in different orientations, i.e., right, left, bottom, and top.

3.3 Categorization of images through fine-tuned inception V3 approach

The Inception model was first summarized by GoogleLENet/Inception V1 [41], and it achieved a significant milestone in the field of image analysis and object detection. The Inception model is heavily revolutionized and goes deeper for finding relevant information from an image. Inception V3 [4, 35] is a popularly adopted image recognition model that incorporates a more simplified architecture than Inception and Inception V2. The basic architecture of Inception V3 is shown in Fig. 4. The Inception V3 model does the extraction of features and image classification from the dataset. This network architecture includes more improvements like factorized 7 × 7 convolutions, RMSProp optimizer, batch normalization, and label smoothing. In the image classification competition, this network was in second place in ILSVRC 2015 [12]. For fine-tuning, we create the model and fill it with ImageNet weight, freeze the foundation layers, build other FC softmax, defreeze the top layer, and then swap it out for the novel FC layer you may do the equivalent of top layer truncation.
The classification or categorizing layer is help to measure the functioning of a FC layer that depends on its expected probabilities. This layer’s many categories are classified using the softmax function [7]. The softmax function’s formula is provided in eq. (1). It compresses the CNN result into a z-score vector in the zero to one band, with total sums equaling 1 [7].

\[
S_n(z) = \frac{e^{z_n}}{\sum_{i=1}^{N} e^{z_i}}
\]  
(1)

Cross entropy (E) is then estimated as depicted in eq. (2)

\[
E = -S_{y_t} + \log(\sum_{i=1}^{N} e^{S_i})
\]  
(2)

Since CNN has the capability to extract knowledgable useful descriptors at a low cost of computation, as much as deepening the networks, we obtained high dimensional features. Apart from the SSDIV3 model, the system is also trained using Mobilenet_v2 [29], VGG-16 [31], VGG-19 [35], Xception [5] models with the different tune or modifying of hyperparameters and check the accuracy of the face mask dataset. These models are prepared for fine-tuning architecture [13]. The fine-tuning approach is crucial to enhance the accuracy and speed up the training images. Fine-tuning is one of the types of deep-transfer-learning that can surpass the feature extraction approach and constructs a new fully-connected head. Freeze
the base layers so that they will not be updated during future training rounds. On the other hand, the head layer weights will be tuned. Analyze the accuracies and time of all the models and check the performance of the FMD.

3.3.1 Optimizer

An optimizer plays an important job in training a deep learning model to diminish the error rate. The optimization technique helps to improve the performance and speed of training a model. Optimizers are the enhanced class, which is comprised of more information to train a model. There are numerous optimizers like SGD [28], AdaGrad [11], RMSProp [11], ADAM [21], ADADELTA [42], Nesterov [34], Ftrl [24] which help to optimize the results. Each has its pros and cons. ADAM optimizer is used with the proposed methodology. The rest are all compared and analyzed in the result section Table 4.

Adaptive Moment Estimation (Adam): Nowadays, this optimizer is showing great interest and works well in deep learning tasks. It is the amalgamation of both momentum and RMSprop algorithms. The updating process looks like RMSprop, except a smooth gradient variant is employed instead of raw stochastic gradients. Adam also includes a bias correction. It is a simple method to implement and requires less memory [21]. Equations 3, 4, 5, and 6 explain the gradient’s updating and include a bias correction method.

\[
gd = \frac{1}{m} \nabla_{\theta} \sum_{i} L \left( f \left( x^{(i)}; \theta \right), y^{(i)} \right)
\]

\[
m = \beta_1 m + (1 - \beta_1) \gd
\]

\[
s = \beta_2 + (1 - \beta_2) \gd^T \gd
\]

\[
\theta = \theta - \epsilon_k \frac{sm}{\sqrt{s + \epsilon_2}}
\]

x(i) and y(i) represent the training dataset and the training labels, respectively, and m is the mini-batch size of the dataset. Equations 3 and 4 describe SGD update the model parameters \( \theta \) in the negative direction of the gradient (gd). The gradient of the loss L is estimated concerning model parameters \( \theta \). Whereas learning rate \( \epsilon_k \) decide the size of the step.

4 Materials required

4.1 Synthesized face mask dataset

The proposed methodology has worked on novel datasets named as new Real-Time Face Mask Dataset (RTFMD) and Real-Time Face Mask Dataset Version 2 (RTFMD-V2) dataset to determine whether the person wears a mask or no mask. Both datasets have been collected through webcam and mobile camera under varying illumination effects in laboratories, parks,
offices, and homes. RTFMD dataset contains 2d printed masks, daily wear masks, whereas the RTFMD-V2 dataset covers all different masks such as 3d_printed masks, natural-looking masks appearance, and also includes all images of the RTFMD dataset. Synthesizing the real-time datasets is a genuine contribution of the work. Moreover, all details related to these datasets are given in further subsections. The pictorial layout of the synthesized RTFMD datasets is given below in Fig. 5.

On sincere analysis of each image in the existing face mask dataset, it has been observed that images present in the dataset consist of various inconsistencies, which are briefly following:

- **Variety of masks**: Considering all types of masks is a significant concern for each dataset. The variety of masks used in the trained model determines the real-time accuracy because people wear different masks. In the existing dataset, either artificially created masks or normal 2d_printed masks are present. This type of dataset may drop the test accuracy and may create a hindrance to recognizing other masks.

- **Viewpoint variation**: The variation in viewpoints has also been seen in images. Some images show only cropped faces with a mask or no masks in the dataset. Due to this, training accuracy achieved good, but test accuracy may fall. Images captured near the camera may or may not represent mask or no mask more precisely.

- **Limitation of face mask dataset**: A limited number of existing face mask datasets are available on the internet. Face mask datasets are collected from Kaggle medical face mask repository. Several researchers worked on this application using their self-synthesized dataset, but none included all categories of masks. Corrupt, noisy images and other varieties of images are present in the dataset. Due to this, researchers have to do a lot of work at the pre-processing stage.

The earlier face mask datasets were created on artificial masks that were not represented in real-world scenarios and produced inaccurate output during testing. There are various limitations in the existing dataset, for instance, not using distinguish masks, lacking viewpoint variations. Hence, we have proposed two synthesized face mask datasets, namely RTFMD and

---

**Fig. 5** The layout of the proposed RTFMD dataset of both version
RTFMD-V2, collected in outdoor and indoor environments, described thoroughly in further subsections.

4.1.1 RTFMD

This dataset contains a total of 620 images having 330 with masks and 290 without masks pictures. They are all collected from a webcam, mobile camera, and the camera’s distance is 2 ft to 7 ft. An arbitrary collection of images has been taken with no gender specialty. In this dataset, different types of masks people have worn to train with distinguishing masks. RTFMD has three categories of images. One person is present in one picture with a mask, while the second, one person is present in one picture with no mask, and the last is two or three people present in the photo either with a mask or without a mask. All images have contained various illumination effects, poses (including side-poses and frontal side). The main characteristics of the RTFMD dataset are discussed below, making this dataset unique compared to other datasets.

- RTFMD dataset is proposed because different types of masks are used to cover the face, like a handkerchief, cloth mask, surgical mask, N-95 mask, etc.
- Images are captured under the varying occlusion, illumination effect, and various poses to check the robustness of the proposed method.
- Real-time images are captured to make the database, and also, one and more than one person is present in a single image.
- Half-covered faces of people are also taken so that system can also distinguish such kinds of images.
- Mask and non-mask images of a person are captured to make this dataset.

4.1.2 RTFMD-V2

RTFMD-V2 is an updated version of the RTFMD dataset. There are two kinds of images present in this dataset, i.e., mask and no mask. Additionally, some more images are included in the mask category, which are 3d_mask, natural-looking appearance mask, skin color mask pictures present. This kind of dataset is not available on any website. Therefore, there is a need to make such type of dataset in which natural-looking mask appearance like human facial appearance, 3d printed facial mask which has natural looking as the same appearance as human face categories are contained. For making the RTFMD-V2 dataset, female-male appearance masks, transparent masks, and clown printed masks have been taken. A total of 1181 images are present (881-mask and 300-without mask) without augmentation reside in the RTFMD-V2 dataset. The updated version of the dataset involved all images from the RTFMD dataset and the new 3d_printed mask, natural-looking like a human face mask. These images are captured through a mobile phone, webcam, and the camera’s distance from a person is 2 ft to 7 ft. Distinct maks are used in this dataset, as shown in Fig. 6, and sample files of the dataset are given in Fig. 7. The main features of the RTFMD-V2 dataset are discussed below, making this dataset unique compared to other datasets.
RTFMD-V2 dataset is proposed because different types of masks cover the face like 3d printed masks, transparent masks, natural-looking appearance masks similar to human facial appearance, handkerchief, surgical mask, etc.

Images are collected with changing occlusion, lighting effect, and varied postures to test the robustness of the proposed technique

This dataset is collected at various places such as parks, colleges, laboratories, and homes with different lighting effects such as sunlight, tube light, and led lights.

Real-time images are captured to make the database, and also, one and more than one person are present in a single image.

Sample of face mask images such as 3d printed mask, skin color, human appearance masks for RTFMD-V2 is shown below in Fig. 8.

4.2 Existing face mask dataset

There are only a few datasets accessible for the recognition of face masks. The majority of them are artificially generated, which either does not address the real world precisely or is full of noise and incorrect labels. Some researchers have used the synthesized face mask dataset for research, and some use existing open-source datasets from the Kaggle dataset library. Face Mask Dataset and Medical Mask Dataset are available at the Kaggle repository, and these were gathered utilizing the dataset given by the Masked Face Recognition Dataset and application [37]. There are 7000 images in Face Mask Dataset [20] taken from Kaggle, in which 3420 without masks and 3580 with masks images are present. However, these datasets have no skin color masks, human appearance masks, or transparent masks.
4.2.1 Real world masked face dataset

This dataset was taken from the internet named Real World Masked Face Dataset (RFMD) [37], containing two images, masked and without mask faces. It has included 5000 masked faces of 525 people and 90,000 without mask faces of people. These images have crawled from the internet and collected at train stoppages etc. But we have taken only 3000 images, in which 1500 images have mask faces, and 1500 have unmasked faces. Figure 9 shows the sample images of the RFMD dataset.

4.2.2 Face mask dataset

The Face Mask Dataset [20] includes 7959 images with having a mask or no mask labeled on their faces. Although, this dataset is a combination of Wider Face and Masked Faces datasets. Wider Face is a collection of 32,203 pictures featuring 939,703 ordinary faces in different lighting occlusion and poses. MAFA comprises 30,811 normal face images and 34,806 masked images. However, some of the faces are covered by hands or other objects rather than actual masks. Noisy and corrupt images are present in the dataset. Here, Fig. 10 displays some images of the face mask dataset.

5 Experiment setup, performance measures, outcomes and result analysis

In this section, the experimental setup, performance measures, outcomes and result analysis are discussed. Different models at different hyperparameter values are considered to measure the
accuracies of the method. Hyperparameter values are fine-tuned to evaluate the performance of the FMD. A hyperparameter is a parameter whose value can be set before the learning process begins for optimal. The setting of hyperparameters values is compassionate because it varies with the new application [26, 27]. Hyperparameter tuning is the problem of selecting a set of optimal hyperparameters for a deep learning tactics [26]. In this paper, Learning_rate (α), Batch_size (m), Epochs (n), Activation functions (AF), Optimizers are considered to evaluate the performance of the system. At different values of hyperparameter, the results are varied. The experimental outcomes are found on the synthesized dataset and past Face Mask Datasets.

5.1 Experimental setup

In this experiment, we have worked on a 16 GB Ram, i-7 intel core processor with 4 GB NVIDIA GPU 1650ti processor supported laptop, Ubuntu-OS is used. The implementation work is done on the Visual Studio software by installing all necessary deep neural networks packages. The dataset is split into a training and validation set with 80% and 20% of dataset images. We have also trained some models without a G.P.U environment, but it takes longer to train. Thus, efficiency degrades. Hence, the proposed work switched to a G.P.U environment, decreased computation time, and trained the dataset in less time.

5.2 Performance measures

This system employed the Precision (PR), Recall (RC), Accuracy (AC) [17–19], and elapsed time as metrics, and they are formulated as follows. The terminology used in the metrics are T.P. - True positive; T.N. - True Negative; F.P. - False Positive; and F.N. - False Negative classes.

5.2.1 Precision

It is stated to estimate the correctness and evaluate the ratio of relevant outcomes from the correctly recognized results in the test window [22].

\[
PR = \frac{\text{T.P.}}{\text{T.P.} + \text{F.P.}}
\]  

(7)

5.2.2 Recall

It is described the ratio of the number of relevant query responses to the actual number of images resides in the dataset.
RC = \frac{T.P.}{T.P. + F.N.} \quad (8)

5.2.3 Accuracy

It can be estimated as the percentage of accurately predicted classes to the all testing class of the dataset [14].

\text{Accuracy} = \frac{T.P. + T.N.}{T.P. + T.N. + F.P. + F.N.} \quad (9)

5.2.4 Elapsed time

The total training time is calculated as the difference between generating models’ start and total end training time.

Elapsed training time = \text{End training time} - \text{Start training time} \quad (10)

5.3 Experiment outcomes

5.3.1 Experimentation with synthesized dataset (RTFMD)

The accuracy and total trained time of existing models and proposed model are summarized in Table 2. Here, the no. of epochs to train the model is 30, the number of dense layers of all models is 256, the AF is Sigmoid, and the ADAM optimizer is used for training. The learning rate in CNN algorithms is the crucial parameter to tune. Picking the learning rate is an important role and also selecting the batch size plays a major role at the time of the learning

| S.no | Models   | Learning Rate (α) | Mini-Batch Size (m) | .0001 | .0001 | .0001 | .001 | .001 | .001 | .01 | .01 | .01 | .01 |
|------|----------|-------------------|---------------------|-------|-------|-------|------|------|------|-----|-----|-----|-----|
|      |          |                   |                     | 16    | 32    | 64    | 16   | 32   | 64   | 16  | 32  | 64  | 16  |
| 1    | Mobilenet| 92.32             | 92.45               | 93.63 | 92.54 | 90.14 | 93.86 | 90.28 | 92.32 | 91.88 |
|      |          | 152.69            | 154.69              | 146.71| 153.77| 151.52| 146.4 | 150.22| 145.76| 146.87|
| 2    | VGG16    | 69.34             | 71.52               | 67.38 | 84.47 | 85.89 | 80.09 | 86.35 | 79.16 | 79.99 |
|      |          | 163.21            | 165.22              | 152.68| 163.69| 155.02| 152.18| 163.49| 155.86| 152.18|
| 3    | VGG19    | 74.59             | 71.46               | 64.88 | 83.78 | 82.35 | 78.61 | 81.26 | 84.64 | 81.98 |
|      |          | 170.63            | 159                | 156.0 | 172.7 | 160.65| 184.62| 171.80| 161.92| 160.2 |
| 4    | Xception | 93.45             | 94.12               | 93.41 | 95.82 | 96.76 | 96.43 | 98.57 | 94.24 | 96.32 |
|      |          | 296.77            | 281.66              | 280.49| 290.05| 281.29| 274.59| 282.61| 276.43| 269.62|
| 5    | SSDIV3   | 95.54             | 95.48               | 94.21 | 97.76 | 98.02 | 96.32 | 96.48 | 96.67 | 97.51 |
|      |          | 274.58            | 267.89              | 267.13| 270.76| 266.6 | 272.93| 276.27| 276.35| 271.09|

Table 2 Accuracies and Training time of models are shown by modifying α and m hyperparameters
Table 3  Accuracies and Training time of models by modifying A.F., D.L. and n hyperparameters

| Models       | AF=Softmax, DL=256 and n=30 | DL=512, AF=Sigmoid and n=30 | n=50, DL=256 and AF=Sigmoid |
|--------------|------------------------------|------------------------------|-----------------------------|
|              | Accuracy | Time (ms) | Accuracy | Time | Accuracy | Time |
| MobilenetV2  | 92.32    | 145.46    | 92.46    | 144  | 92.25    | 227.69 |
| VGG16        | 80.67    | 158.17    | 84.25    | 148.02 | 90.06  | 253  |
| VGG19        | 81.33    | 156.11    | 81.08    | 158  | 84.65    | 258.39 |
| Xception     | 93.24    | 271.66    | 97.04    | 274.09 | 95.05  | 454.48 |
| SSDIV3       | 95.88    | 260.18    | 98.45    | 261.94 | 97.25  | 432.42 |

process. Table 2 shows the accuracy and time changes by changing the training model’s learning rate and batch size value. The average accuracy of the SSDIV3 model outperforms on the RTFMD dataset. More analysis on the RTFMD dataset has been done by tuning the models’ parameters, and their results have been presented in tabular format (Table 3 and Table 4). The comparative analysis of other models with proposed SSDIV3 on our generated dataset RTFMD is presented in Tables 2, 3, and 4. The experimentation has been performed on the synthesized Face Mask benchmark datasets.

These graphs in Fig. 11 show the training accuracy, loss and validation accuracy, and loss of models. Apart from the SSDIV3 model, Xception, MobileNet, VGG19 models are also used to train the dataset, determine the accuracy, and conclude that the SSDIV3 model has worked better than other models. Overfitting problem has occurred in Xception, MobileNet V2 models, i.e., models are trained more, but SSDIV3 model did not face this problem. VGG19 has not provided good train and validation accuracy. After obtaining the results, it can conclude that the proposed model and Xception provide better accuracy, but SSDIV3 is taking less computation time than the Xception model. Hence, for the face mask detection system, we choose the SSDIV3 model for this proposed work.

Table 4  Accuracies and Training time of models by modifying Optimizers

| Models       | S.G.D | RMSProp | AdaGrad | Adam | AdaDelta | NAdam | Ftrl |
|--------------|-------|---------|---------|------|----------|-------|------|
| Mobilenet_V2 | 91.45192.2 | 93.17 | 90.32 | 94.45 | 84.45 | 96.25 | 89.25 |
| VGG16        | 60.45 | 84.13 | 184.17 | 187.74 | 146.4 | 185.08 | 183.3 | 185.3 |
| VGG19        | 250.5 | 227.97 | 227.55 | 152.1 | 228.41 | 229.91 | 227.6 |
| Xception     | 60.32 | 77.01 | 57.14 | 78.98 | 38.85 | 79.74 | 52.15 |
| SSDIV3       | 244.7 | 243.94 | 244.51 | 181.46 | 243.54 | 244.8 | 243.7 |
|              | 90.41 | 94.43 | 89.20 | 95.82 | 71.51 | 93.58 | 52.22 |
|              | 427   | 429    | 431.26 | 274.5 | 426.09 | 428.4 | 430.25 |
| SSDIV3       | 96.05 | 92.54 | 92.69 | 98.45 | 70.62 | 95.92 | 76.86 |
|              | 365.3 | 354.60 | 350.30 | 272.9 | 351.66 | 354.34 | 343.28 |
The aforementioned details in Table 3 show the performance of the model, if change subsequent hyperparameters activation function (AF), number of dense layer (DL), and number of epochs(n). The performance of training is varied if the value of hyperparameters is changed. The activation function helps to keep the relevant information and quash the irrelevant information. Sigmoid is used for binary classification purposes and is suitable for the output layer, while Softmax function is applicable for multi-classification applications. In this table, we will see the differences in results. Dense layer neurons can be changed in the hidden layers but can not change the neurons of the dense layer in the layers. Epochs play a significant role in the performance of the training model. If the number of epochs is too much, then the model will have faced an overfitting problem. The system’s performance in terms of validation accuracy for each iteration is to determine whether it over-fit the training data. The learning rate and batch size are 0.001 and 32, respectively used in Table 3. It can be seen that the proposed works well by taking the sigmoid activation function and 512 dense layers at batch size 32 therefore, these hyperparameters will be used in the subsequent face mask dataset in this article.

Various optimizers are shown in next aforementioned in Table 4 and will analyze the performance of our models. The comparison result of five optimizers on different models is shown in Fig. 12 graph. The AF - Sigmoid, learning rate – 0.001, number of epochs – 30 and batch size – 32 are used in Table 4 for evaluating the results. ADAM and NADAM perform
well on our dataset as compared with their optimizers. AdaDelta and Ftrl optimizers are not given compromising results. SGD, AdaGrad, RmsProp provide better results as compared with AdaDelta and Ftrl optimizers. Overall, the proposed and Xception models offer better accuracy, but the Xception model takes more computation time than the SSDIV3 model. Thus, the ADAM optimizer is considered for further experiments in this article.

### 5.3.2 Experimentation with synthesized dataset (RTFMD-V2)

The proposed network produces outstanding results for synthesized face mask datasets (RTFMD-V2) with different batch size tuning, as shown in Table 5. The previous tables show that the system has experimented with best-updated hyperparameters, considering only those parameter values for the RTFMD-V2 dataset. The outcomes are evaluated at learning rate $10^{-4}$.

| Model used    | Batch size | Train | Validation | Training time (In milliseconds) |
|--------------|------------|-------|------------|---------------------------------|
|              |            | Train_Accuracy | Train_Loss | Val_Accuracy | Val_Loss |        |
| MobileNet    | 16         | 0.95   | 0.14       | 0.94  | 0.21       | 765.16  |
|              | 32         | 0.96   | 0.08       | 0.95  | 0.12       | 766.24  |
|              | 64         | 0.96   | 0.08       | 0.96  | 0.094      | 767.23  |
| VGG16        | 16         | 0.87   | 0.31       | 0.84  | 0.31       | 873.92  |
|              | 32         | 0.85   | 0.299      | 0.85  | 0.304      | 865.82  |
|              | 64         | 0.83   | 0.34       | 0.84  | 0.33       | 863.92  |
| VGG19        | 16         | 0.88   | 0.35       | 0.85  | 0.35       | 940.12  |
|              | 32         | 0.86   | 0.32       | 0.863 | 0.30       | 932.67  |
|              | 64         | 0.832  | 0.34       | 0.86  | 0.32       | 930.48  |
| Xception     | 16         | 0.97   | 0.07       | 0.97  | 0.08       | 1798.54 |
|              | 32         | 0.98   | 0.08       | 0.9812| 0.06       | 1758.26 |
|              | 64         | 0.96   | 0.08       | 0.98  | 0.072      | 1734.77 |
| SSDIV3(Proposed) | 16     | 0.98   | 0.064      | 0.97  | 0.107      | 1694.29 |
|              | 32         | 0.99   | 0.03       | 0.98  | 0.06       | 1503.29 |
|              | 64         | 0.97   | 0.06       | 0.97  | 0.097      | 1497.63 |

The bar-graph depicts the difference of optimizers in mentioned models.
3, the activation function-sigmoid, dense layer-512, the ADAM optimizer, and each model is trained over 50 epochs (because the dataset is large compare with RTFMD). In the context of achieving higher accuracy for each case, it is found that the proposed model produces up to 99% training accuracy and 98% validation accuracy for batch size 32. As shown below in Table 6, the training and validation accuracy is high for each model approx at batch size 32. The training and validation loss is low for other models at batch size 32. This table also interpolates the training time of each model, which shows higher computational timing for smaller batch sizes and lower computational time for larger batch sizes.

Upon analyzing outcomes, it is found that the proposed model offers higher validation accuracy at batch size 32 with minimum loss compares to other models. Therefore, a comparison of validation accuracies and losses of all models at batch size 32 is shown in Fig. 13a. A bar graph shows the validation accuracy and loss of different models and depicts that the proposed methodology works excellent on the synthesized dataset. The processing time to generate the accuracies and losses of each model is shown in Fig. 13b. The bar graph represents elapsed time to evaluate the model at batch size 32 and trained over 50 epochs. It can conclude that both graphs illustrate the trade-off between each model’s accuracy and time taken.

A comparison of training and validation trade-offs for batch size 32 of each model over the synthesized RTFMD-V2 dataset is depicted in Fig. 14, which shows the proposed model’s used deep learning approach and has achieved good performance. Figure 14a shows the accuracies and loss trade-off for training and validation over 50 epochs of the SSDIV3 model, which incurs smooth training and achieves preferably good validation accuracy up to 98% and minimum loss during the training and validation phase. Similarly, Fig. 14b–e, shows the accuracies and losses of the trade-off for training and validation phases for other models.

![Table 6](image)

| Synthesized dataset | Image size | Precision | Recall | Accuracy(%) | Loss | Time elapsed |
|---------------------|------------|-----------|--------|-------------|------|--------------|
| RTFMD               | 620        | 96.93     | 97.23  | 98.45       | 0.09 | 261.94       |
| RTFMD-V2            | 1180       | 98.12     | 97.45  | 98.75       | 0.05 | 1503.29      |

![Fig. 13](image)

Fig. 13 Comparative analysis among different models (a) Accuracy (b) elapsed time
The aforementioned details in Table 6 clearly mention the final experimental outcomes of the proposed evaluation over self-synthesized face mask datasets for different modeling parameters. In terms of achieving higher accuracy for each case from above Table 2 and 5, it has been seen that batch size 32 produces good results. Therefore, this table depicts the final results for both synthesized datasets with batch size 32, which can be used for further comparative analysis. The precision and recall over the RTFMD dataset using the proposed work produce 96.93% and 97.23%, respectively, for image size 620 at batch size 32. Similarly, accuracy and loss are found up to 98.45% and 0.09, respectively. The proposed model is trained over 30 epochs for the RTFMD dataset because, after 30 iterations, the result is constant, and an overfitting problem will occur later.

On the other hand, the proposed model produces 98.12%, 97.45% precision, and recalls at batch size 32 on the RTFMD-V2 dataset. Additionally, accuracy and loss are 98.75% and 0.05, respectively. The accuracy is computed over 50 iterations. The model stops training because no progress has been found in validation loss after 50 epochs, which saves the model from overfitting issues.
5.3.3 Experimentation with existing face mask dataset

Several researchers experimented with various modeling parameters on the face mask dataset. It has been found from the above sections that our proposed model works efficiently as compared with other methods. Various parameters have been taken to analyze the system’s performance using the proposed work. As seen in previous Tables 2, 3, 4 and 5, the presented model has performed well at 1e-3 Learning rate, Sigmoid activation function, 512 dense layers, and ADAM optimizer. Hence, the system is evaluated using the proposed model with the same optimized parameters on the existing face mask dataset. Table 7 illustrates the precision, recall, accuracy, loss, and total elapsed time are computed for RFMD and Face Mask Dataset (taken from Kaggle) at 16,32,64 batch size. After the assessment, it is observed that the proposed model works well at batch size 32 for both datasets. The model produces a good accuracy score with minimum loss.

5.3.4 Comparative analysis

The proposed system is evaluated on four datasets that have produced decent outcomes for different cases of modeling parameters. Bar graph Fig. 15 shows the precision, recall, and accuracy comparison for the synthesized and existing face mask datasets. We examined the graph that the presented model trained with different batch sizes provides variant accuracy for all datasets. As we take a smaller dataset, a small batch size can produce good results, but if we take a large dataset, then a high batch size should be taken for better performance in the context of accuracy. In the RTFMD dataset, the total image size is 620, so 16 batch size is enough to produce good results. While, on the RTFMD-V2 dataset, the system evaluates excellent outcomes at batch sizes 32 and 64. Similarly, the RFMD and Face_mask datasets have large face mask datasets; hence, the proposed work computes good results at batch sizes 32 and 64.

Below, Fig. 15 depicts the comparison of the evaluated parameters for standard existing and self-synthesized datasets in the context of precision, recall, and accuracy with three possible batch sizes, i.e., 16,32,64.

Furthermore, Fig. 16 illustrates the loss that occurs during training the model at different batch sizes for existing and synthesized face mask datasets. The impact of hyper-parameters while training is clearly presented. Loss value shows how much the value of loss incurs during training the proposed model on datasets. We have computed the validation_loss for the proposed model over datasets.

Upon analysing the above bar graph, it is found that the proposed model with three different batch sizes estimates the loss in percentage. It can be concluded as with larger dataset with a

| Name_of Dataset | Image Size | Batch size | Precision(%) | Recall(%) | Accuracy(%) | Loss | Time elapsed(ms) |
|-----------------|------------|------------|--------------|-----------|-------------|------|------------------|
| RFMD            | 3000       | 16         | 94.24        | 96.48     | 95.64       | 0.18 | 4124.32          |
|                 |            | 32         | 96.85        | 97.45     | 97.85       | 0.09 | 4022.45          |
|                 |            | 64         | 96.78        | 96.81     | 96.98       | 0.10 | 4006.29          |
| Face_Mask       | 7000       | 16         | 91.48        | 93.72     | 92.68       | 0.35 | 11,038.65        |
| Dataset         |            | 32         | 95.12        | 96.25     | 95.85       | 0.12 | 10,125.39        |
|                 |            | 64         | 93.55        | 96.48     | 95.90       | 0.19 | 10,012.26        |

Table 7 Evaluated results on the Existing dataset using the proposed methodology
large batch size produces a minimum loss. On the other hand, a smaller dataset with a small
batch size computes less loss. The face mask dataset has the most extensive dataset (7000
images), generating high loss at batch size 16. While RTFMD dataset has a smaller number of
images (620 images) and the model produces minimal loss at batch size 16. Additionally, we
also found that smaller batches spend a longer training time than larger batches.

5.4 Result analysis

After finding the outcomes with various models at different modeling parameters, there are
also some following observations, which are listed below:

Analysis 1 Compared to other models, the proposed model performs well for the Synthesized
and Existing Face Mask Datasets.
**Explanation** On analyzing the above experimental outcomes from Tables 2, 5, and 7, it can be seen that the proposed SSDIV3 model works outperform as compare with other models. This model achieved higher accuracy, up to 99% on the synthesized datasets, and minimal loss was up to 0.05. On the other hand, the proposed work achieved up to 97% accuracy and loss up to 0.09 for the existing dataset. It can also be noted in Tables 2, 5, and 7 that the accuracy achieved from the Xception model is approximately the same as a proposed model in some cases but the training time required to train this model is more. However, the SSDIV3 model provides excellent accuracy with a minimum loss, and it also takes less time to train the proposed model than other models, as shown in Figs. 13, 15, and 16.

From Table 8, it can infer that the accuracy, and F1 score are high of proposed model compared with other state-of-the-art models. The precedence of these parameters to choose the proposed model is accuracy > F1 score > time required > number of parameters > model size. The main focus of this paper is to achieve the best accuracy and F1 score with less time required. However, MobileNetV2 is a lightweight model among all models in terms of several parameters, elapsed time, and model size, but the accuracy and f1-score are not good; therefore, they will not be considered. Moreover, VGG16 and VGG19 models generate fewer (trainable and non-trainable) parameters after mobilenetv2 but low accuracy. Consequently, the Xception and Inception models have taken approximately the same parameters. The model size of the inception model is less bulky than the Xception model along with the accuracy, and the F1 score of SSDIV3 is better than the Xception model. The total computation time required to execute the process is less in the proposed model than in the Xception. Thus, SSDIV3 is the best suitable approach for the FMD system among all models. Henceforth, the proposed model produced good results on the synthesized and existing face mask datasets.

**Analysis 2** The network is trained on different modeling parameters over the synthesized RTFMD dataset and finds the best parameters for other datasets.

**Explanation** The system is trained with different models and found SSDIV3 is the best among other models. Likewise, the network also trains the model at different modeling parameters such as learning rate, batch size (as discussed previously), dense layer, activation function, and optimizers. Activation functions control weights and bias of neural networks, and without using it, a model behaves like a linear regression model. Sigmoid and softmax functions are used over the RTFMD dataset, and we have concluded that sigmoid works better than softmax functions, as shown in Table 3. It is because the sigmoid function is used to classify binary categories images, whereas the softmax function is used for a multiclassification purpose. Dense layer neurons can be changed in the hidden layers but can not change the neurons of the dense layer in the layers. The system performs well upon analyzing Table 2 by changing the

| Models      | Accuracy | Time Required(ms) | F1 Score | Total number of parameters | Model size |
|-------------|----------|-------------------|----------|---------------------------|------------|
| MobileNetV2 | 96%      | 766.24            | 0.96     | 2 Million                 | 20 MB      |
| VGG16       | 85%      | 865.82            | 0.85     | 14 Million                | 64 MB      |
| VGG19       | 86%      | 932.67            | 0.86     | 21 Million                | 83.8 MB    |
| Xception    | 98%      | 1774.1            | 0.98     | 21.9 Million              | 98 MB      |
| SSDIV3      | 99%      | 1607.3            | 0.99     | 22.3 Million              | 96 MB      |
layer from 256 to 512 with updating softmax to sigmoid activation function as shown in Fig. 17a.

Lastly, an optimizer plays a crucial role in training a deep learning model to diminish the error rate. For this system, various optimizers have been taken to check the performance of the proposed work over the RTFMD dataset, as seen in Table 4. Figure 5 shows that the ADAM optimizer works well for all models on the synthesized dataset. The proposed model achieved up to 97% after applying ADAM optimizer, which is highest among other optimizers. Additionally, the total computation time is minimal compared to other optimizers on the RTFMD dataset, as depicted in Fig. 17b. Due to these variations, the overall proposed algorithm complexity is good in accuracy and time terms as compared with other algorithms.

Therefore, we have taken the best parameters such as Sigmoid as activation function, Dense layer is set to be 512, and ADAM optimizer is used for RTFMD-V2, RFMD, Face Mask dataset for evaluating the metrics of the proposed model.

**Analysis 3** The network is trained on different learning rates over the synthesized RTFMD dataset, but the 1e-3 learning rate has been taken for all datasets.

**Explanation** The crucial parameter to tune is a deep learning algorithm’s learning rate (LR). It affects the most to measure the performance of the training algorithm. Using an appropriate scale to pick a learning rate plays an important role. In Table 2, we have used three learning rate values to train the model, and observed results are varied as the LR is changed. Upon analyzing the results, it has been noted that using 1e-4 learning produces outperforming accuracy with minimal loss. Henceforth, further training for the remaining datasets has been done with LR = 1e-3 for better outcomes.

**Analysis 4** Training time is the longest for the network with a mini-batch size.

**Explanation** The batch size reflects the number of samples transmitted through the network before updating the internal parameter. The batch size is neither too long nor too small; it should be somewhere in between. Also, it is based on the dataset size. It also impacts the accuracy with which the error gradient is estimated while neural networks are trained. It has
been observed that the networks with smaller batch sizes require a longer time to train. This is due to the lower batch size generated in a greater number of training instances. The network accomplishes this by evaluating each sample and using backpropagation to update internal parameters. Assessing a huge number of small samples takes more time to train. Hence, the network to train the model spends more time processing the small batches. On the other hand, a network with a greater batch size creates fewer data and takes less time to compute. However, choosing a batch size depends on the size of the dataset if the main concern is finding the system’s accuracy.

Figure 18a, b illustrates the total training time using the suggested model for distinguishing batch sizes in milliseconds. On observing RTFMD experimental findings, it has been seen that the network presented with 16 batch size entail less computation time because of the small dataset size. However, on RTFMD-V2 and other existing datasets, batch size 32 and 64 take less training time than batch size 16 because all have a more significant number of images. Moreover, the augmentation technique doubled these images in each dataset at run time.

**Analysis 5** The proposed system trained on self-synthesized face mask datasets provides better results than the existing face mask datasets.

**Explanation** The proposed work evaluated for both types of the dataset, i.e., synthesized and existing face mask dataset with various modeling parameter cases, generates precise validation accuracy, precision, recall, and loss from Tables 6 and 7. Figure 15 depicts the precision, recall, and accuracy comparison over datasets, and Fig. 16 shows the loss comparison over face mask datasets. Although, it is observed that our projected work computed over synthesized face mask dataset (RTFMD and RTFMD-V2) provides much better results than the existing face mask dataset (RFMD and Face Mask Dataset-Kaggle). This is because the masks are artificially created, and the dataset is full of noise and does not have all categories of masks in the dataset. So, these datasets do not represent the natural masks. Because of these deficiencies, the suggested system produces some false positives at the time of the validation process and run-time testing. On the contrary, the synthesized face mask datasets have been collected in real-time scenarios by considering these vulnerabilities and capturing human faces with masks and without masks. Henceforth, the designed model with a synthesized face mask

![Figure 18](image-url)
dataset provides higher validation accuracy and test accuracy at run time with fewer false positives images during the validation process than existing face mask datasets.

5.5 Resultant images in real-time

The testing is performed in a real-time stream using the proposed model trained over the synthesized and the existing face mask datasets. The below figures show the resultant images, i.e., a person wearing a mask or without a mask in run time. The proposed trained model over the existing face mask dataset produces results depicts in Fig. 19. However, some false positive instances in the resultant images are caused by discrepancies in the dataset, as shown in Fig. 19b. If the mask covers only the mouth, then the system monitors and detect the human put the mask on face; though, it should not be as shown in Fig. 19b. Consequently, this is a false positive, and the error rate is high using the trained model over the existing face mask dataset in a real-time stream. Besides, the existing datasets do not contain skin tone masks, so when a person wears 3d printed or skin tone masks, the system predicts wrong in the real-time stream. However, these deficiencies are removed in our when the proposed model is trained over the synthesized dataset.

The outcomes over synthesized face mask datasets generated scores much more proficient during the testing phase and provided accurate classification, as shown in Figs. 20 and 21. The model is evaluated over the RTFMD dataset, and the resulting instances with a live camera over the testing phase are shown in Fig. 20. The first image illustrates that a person is without a mask; the system correctly predicts and labels with no mask, as depicted in Fig. 20a. Figure 20b shows that the only mouth partially covers a person’s face with a mask, and the system detects accurately with no mask. Likewise, if a person is covered with a mask properly, the system predicts with mask in Fig. 20c. Additionally, if a person’s face is slightly tilted with a right or left wear mask, the system correctly detects with mask in Fig. 20d, e. Figure 20(f) presented the multiperson face covered with a mask in a single frame and detected correctly with the mask.

Furthermore, the model is trained over RTFMD-V2, and the resultant live images during the run time are shown in Fig. 21. A person’s face is slightly tilted and wears skin tone masks, and another person wears 2d_printed masks in a single frame; the system shows the accurate
result and labeled with the mask on the face as depicted in Fig. 21a. Figure 21b presents that one person is without the mask and the other is with a natural-looking mask appearance, and the system predicts well and annotates with without mask and with mask respectively. Lastly, Fig. 21c represents the multiperson face are covered with natural-looking masks appearance, and the system labeled them with the mask.

5.6 Comparative assessment with existing methods

Furthermore, the projected system is compared with other existing approaches, as illustrated in Table 9. Here, Jiang et al. [20] have presented a two-model RetinaFaceMask with MobileNet

![Fig. 21 Resultant images for RTFMD-V2 dataset. (a) Multiperson covered with natural-looking masks, and 2d_printed mask (b) Multiface covered with a natural-looking mask and no mask (c) Multiperson covered with natural-looking masks appearance](image-url)
and ResNet models for predicting masks and without masks. Preeti Nagrath et al. [25] take advantage of the MobileNet model with a single-shot detector to identify the mask and no mask. In addition to these, another face mask detection technique by Arjya Das [8] is computed by CNN pre-trained model. Consequently, the proposed model used an SSD face detector and a deep Inception V3 network to bind the pre-processing technique to encode face mask detection. The presented algorithm is robust in real-time because the system can predict any masks, either 3d_printed, natural-looking appearance masks or simple masks with high accuracy in less time. Table 9 concludes that the suggested model, trained over both face mask datasets, encompasses more accurate outcomes in real-time than existing approaches.

### 6 Conclusion

Motivated by modern advances in deep neural networks, we successfully employ the pre-trained Caffe model and deep Inception V3 design (SSDIV3) model with different modeling parameters to identify the person having a mask and no mask in public places, megastores, etc. Moreover, this article presents two synthesized face mask datasets, masks images (considered varieties of masks) and no masks images. The research is mainly intended to protect everyone from this pandemic era. The synthesized (RTFMD and RTFMD-V2) and existing face mask datasets have been taken for evaluation purposes. The experiments found that our proposed model performs remarkably well to categorize the face Mask and No Mask. The investigated outcomes show that the system proposed has attained a detection average accuracy of up to 96% on the existing face mask dataset and 98.43% on the synthesized face mask datasets, which remarkably perform well from other existing FMD systems. The SSDIV3 model presumably assists the concerned authorities in this major pandemic scenario, which has spread throughout the world.

The drawback of the system is that it follows the pipeline approach, which means one module is dependent on the other module. If the face detection module does not properly work, then the detection of the face mask becomes a failure. On the other hand, the illumination and occlusion may affect the result outcomes; thus, further research will cover these issues of the present system. The futuristic scope is that the other researchers can use our synthesized datasets for more applications such as face recognition, face spoofing, and facial landmarks detection in a surveillance system. If anyone wears a mask and there is a need to recognize the person’s face because of security purposes, this model and dataset can be considered. Moreover, the suggested model can be applied in human activity recognition, face spoofing,

### Table 9 Comparison of proposed model and dataset with existing approaches and dataset

| Methods               | Accuracy (%) | Previous Face Mask Datasets | Synthesized Face Mask Datasets |
|-----------------------|--------------|----------------------------|-------------------------------|
|                       |              | Face Mask Dataset [20]      | Medical Face Mask dataset [25]|
|                       |              | 92.48                      | 93.02                         | 93.25 | 95.6  
| Preeti Nagrath et al. [25] | 93.49        | 93.29                      | 95.45 | 96.45 |
| Xingi et al. [20]     | 95.77        | 94.58                      | 93.45 | 94.76 |
| Arjya Das et al. [8]  | 95.96        | 96.82                      | 98.45 | 98.75 |
| Proposed model        | 95.96        | 96.82                      | 98.45 | 98.75 |
facial landmarks detection, face recognition and object recognition tasks because the SSDIV3 model extracts pertinent features from the images and videos.

Acknowledgments The authors would like to thank all people involved in preparing this valuable dataset, and there is no funding requirement in this paper.

Declarations

Competing interests None of the authors of this paper have a financial or personal relationship with other people or organizations that could inappropriately influence or bias the content of the paper. It is to specifically state that “No Competing interests are at stake and there is No Conflict of Interest” with other people or organizations that could inappropriately influence or bias the content of the paper.

References

1. Ansari M, Singh DK (2021) Human detection techniques for real time surveillance: A comprehensive survey. Multimed Tools Appl 80(6):8759–8808
2. Bastanfard A, Bastanfard O, Takahashi H, Nakajima M (2004) Toward anthropometrics simulation of face rejuvenation and skin cosmetic. Comput Animation Virt Worlds 15(3–4):347–352
3. Bastanfard A, Takahashi H, Nakajima M (2004) Toward E-appearance of human face and hair by age, expression and rejuvenation. In: 2004 International conference on Cyberworlds. IEEE, pp 306–311
4. Bazi Y, Al Rahhal MM, Alhichri H, Alajlan N (2019) Simple yet effective fine-tuning of deep CNNs using an auxiliary classification loss for remote sensing scene classification. Remote Sens 11(24):2908. https://doi.org/10.3390/rs11242908
5. Chollet F (2017) Xception: deep learning with depthwise separable convolutions. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 1800-1807. https://doi.org/10.1109/CVPR.2017.195
6. Chowdary GJ, Punn NS, Sonbhadra SK, Agarwal S (2020) Face mask detection using transfer learning of inceptionv3. In: International Conference on Big Data Analytics. Springer, Cham, pp 81–90
7. Dalgleish T, Williams JMG, Golden AMJ, Perkins N, Barrett LF, Barnard PJ, Au Yeung C, Murphy V, Edward R, Tchanturia K, Watkins E (2007) Reduced Specificity of autographical memory and depression. J Exp Psychol Gen 136(1):23–42
8. Das A, Ansari MW, Basak R (2020) Covid-19 face mask detection using TensorFlow, Keras and OpenCV. In: 2020 IEEE 17th India Council International Conference (INDICON). IEEE, pp 1–5
9. Deng J, Guo J, Zhou Y et al (2019) Retinaface: Single-stage dense face localisation in the wild. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 5203–5212
10. Duan Y, Zhou L, Wu Y (2016) Facial Expression Recognition Based on Convolution Neural Network. In: 2nd International Conference on Computer Engineering, Information Science & Application Technology (ICCIA 2017). Atlantis press, pp 4077–4081 Jul 13
11. Duchi JC, Bartlett PL, Wainwright MJ (2012) Randomized smoothing for (parallel) stochastic optimization. In: 2012 IEEE 51st IEEE Conference on Decision and Control (CDC), pp 5442–5444. https://doi.org/10.1109/CDC.2012.6426698
12. Ge S, Li J, Ye Q, Luo Z (2017) Detecting masked faces in the wild with LLE-CNNs. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 426–434. https://doi.org/10.1109/CVPR.2017.53
13. Guo Y, Shi H, Kumar A, Grauman K, Simunic T, Feris RS (2019) SpotTune: transfer learning through adaptive fine-tuning. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 4800–4809
14. Gupta S, Thakur K, Kumar M (2021) 2D-human face recognition using SIFT and SURF descriptors of face’s feature regions. Vis Comput 37(3):447–456
15. Hariri W (2021) Efficient masked face recognition method during the covid-19 pandemic. SIViP:1–8
16. Hussain, GKJ, Priya R, Rajarajeswari S, Prasanth P, Niyazudddeen N (2021) The face mask detection technology for Image analysis in the covid-19 surveillance system. J Phys Conf Ser 1916(1):012084. IOP Publishing
17. Jayaswal R, Dixit M (2020) Comparative analysis of human face recognition by traditional methods and deep learning in real-time environment. In: 2020 IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT). IEEE, pp 66–71
18. Jayaswal R, Dixit M (2021) Detection of hidden facial surface masking in stored and real time captured images: a deep learning perspective in Covid time. Traitement du Sig 38(6):1875–1885
19. Jayaswal R, Jha J (2017) A hybrid approach for image retrieval using visual descriptors. In: 2017 International Conference on Computing, Communication and Automation (ICCCA). IEEE, pp 1125–1130
20. Jiang M, Fan X (2020) RetinaMask: A Face Mask detector. arXiv preprint arXiv:2005.03950.
21. Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980
22. Kumar M, Chhabra P, Garg NK (2018) An efficient content based image retrieval system using BayesNet and K-NN. Multimed Tools Appl 77(16):21557–21570
23. Kumar A, Kaur A, Kumar M (2019) Face detection techniques: a review. Artif Intell Rev 52(2):927–948
24. Mahan HB, Holt G, Sculley D, Young M, Ebner D, Grady J, Nie L et al (2013) Ad click prediction: a view from the trenches. In: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 1222–1230
25. Nagrath P, Jain R, Madan A, Arora R, Kattaria P, Hemanth J (2021) SSDMNV2: a real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2. Sustain Cities Soc 66:102692
26. Paul S, Kurin V, Whiteson S (2019) Fast hyperparameter tuning for policy gradient methods. Advances in Neural Information Processing Systems 32
27. Probst P, Boulesteix AL, Bischl B (2019) Tunability: importance of Hyperparameters of machine learning algorithms. J Mach Learn Res 20(53):1–32
28. Ruder S (2016) An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747
29. Sandler M, Howard A, Zhu M, Zhmoginov A, et al. (2018). Mobilenetv2: inverted residuals and linear bottlenecks. In proceedings of the IEEE conference on computer vision and pattern recognition: 4510-20.
30. Sethi S, Kathuria M, Kaushik T (2021) Face mask detection using deep learning: an approach to reduce risk of coronavirus spread. J Biomed Inform 120:103848
31. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556:1–14
32. Singh S, Ahuja U, Kumar M, Kumar K, Sachdeva M (2021) Face mask detection using YOLOv3 and faster R-CNN models: COVID-19 environment. Multimed Tools Appl 80(13):19753–19768
33. Susanto S, Putra FA, Analia R, Suciningtyas IKLN (2020) The Face Mask Detection For Preventing the Spread of COVID-19 at Politeknik Negeri Batam. In: 2020 3rd International Conference on Applied Engineering (ICAE). IEEE, pp 1–5
34. Szegedy C et al (2015) Going deeper with convolutions. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 1–9. https://doi.org/10.1109/CVPR.2015.7298594
35. World Health Organization, Coronavirus disease (COVID-19) advice for the public (2020) https://www.who.int/emergencies/diseases/novel-coronavirus-2019/advice-for-public. Accessed June 10, 2020.
36. Worldometer, COVID-19 Coronavirus Pandemic, (2020) https://www.worldometers.info/corona-virus/. Accessed June 10, 2020.
37. Yadav S (2020) Deep learning based safe social distancing and face mask detection in public areas for covid-19 safety guidelines adherence. Int J Res Appl Sci Engin Technol 8(7):1368–1375
38. Zeiler MD (2012) Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701
39. Zhang F, Fan X, Ai G et al (2019). Accurate face detection for high performance. arXiv preprint arXiv:1905.01585:1–9

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.