



Electronic Government, an International Journal

ISSN online: 1740-7508 - ISSN print: 1740-7494

<https://www.inderscience.com/eg>

Innovative approach of masked facial recognition

Om Pradyumana Gupta, Arun Prakash Agrawal, Om Pal

DOI: [10.1504/EG.2025.10063683](https://doi.org/10.1504/EG.2025.10063683)

Article History:

Received:	19 May 2023
Last revised:	27 January 2024
Accepted:	12 March 2024
Published online:	03 December 2024

Innovative approach of masked facial recognition

Om Pradyumana Gupta*

Department of Computer Science and Engineering,
Sharda University,
Greater Noida, India
Email: op.gupta@nic.in
*Corresponding author

Arun Prakash Agrawal

Department of Computer Science and Applications,
Sharda University,
Greater Noida, India
Email: arun.agrawal@sharda.ac.in

Om Pal

Department of Computer Science,
University of Delhi,
New Delhi, India
Email: opal@cs.du.ac.in

Abstract: Not only diseases such as COVID-19 but also professions across healthcare, construction and manufacturing require the usage of facemasks. Currently, obstacles that are present in the available masked facial recognition techniques lack a focus on Indian specific databases which poses serious challenges in surveillance. In this paper, the proposed model, which achieved 96.7% accuracy, works in two-stages. The input goes through the pre-processing stage, which is based on an image cropping approach with a multi-task cascaded convolutional neural network applied over it. Output moves to the convolutional neural network stage where the image is trained on two models, convolutional block attention module and modified face net architecture. Special techniques such as channel attention, spatial attention, batch normalisation, leaky rectification and face embedding vector normalisation, triplet selection are applied to reach an optimal level of accuracy. The robotic process automation engine using Bayesian optimisation is also put in place.

Keywords: image processing; face recognition; artificial intelligence; Python; masked face detection; deep learning; neural network; COVID-19; facial landmark points; feature extraction.

Reference to this paper should be made as follows: Gupta, O.P., Agrawal, A.P. and Pal, O. (2025) 'Innovative approach of masked facial recognition', *Electronic Government*, Vol. 21, No. 1, pp.35–67.

Biographical notes: Om Pradyumana Gupta is pursuing his PhD in the field of Artificial Intelligence, Computer Vision and Deep Learning from Sharda University, Greater Noida, India. Earlier, he has received degree of MTech in

Information Technology and MCA. Presently, he is working as a Scientist-E in National Informatics Centre, Ministry of Electronics and Information Technology, Govt. of India. He is involved in a number of projects of national level. Gov.in Secure Intranet, Dididhan Dashboard, Electronics Transaction Aggregation and Analysis Layer (eTaal) are among few of them. His area of interest includes artificial intelligence, computer vision and deep learning.

Arun Prakash Agrawal is currently working as a Professor with the Department of Computer Science and Application and Engineering at Sharda University, Greater Noida, India. He received his PhD and Master's in Computer Science and Engineering from Guru Gobind Singh Indraprastha University, New Delhi. He was the gold medallist of his batch. Prior to his current assignment, he has served many academic institutions including Amity University, Noida. He has also taught short-term courses at Swinburne University of Technology, Melbourne, Australia and Amity Global Business School, Singapore. His research interests include machine learning, AI and soft computing.

Om Pal received his BE in Computer Science and Engineering from Dr. B.R. Ambedkar University, Agra, MBA (O&M) from IGNOU, MS (Research) in field of Cryptography from IIT Bombay and PhD in field of Cyber Security from Jamia Millia Islamia, New Delhi. Presently, he is working as an Associate Professor in Department of Computer Science, University of Delhi (Central University), North Campus, Delhi. His area of interest includes cryptography, information security, cyber security, network security, post-quantum cryptography, quantum computing, blockchain technology, algorithms analysis and design, cyber law, artificial intelligence for cyber security, machine learning for cyber security, theory of automata, operation research and mathematics.

1 Introduction

Masked facial recognition has, thus, gained notable significance since COVID-19. However, its scope is not just limited to the people wearing masks at large, but also extends to the realm of surveillance.

India witnessed a surge in facial masks because of diseases such as COVID-19, which can be a threat in the future as well. Front line workers such as doctors, nurses, policemen, government officials, etc. are the ones who face the situation first. However, with growing digital penetration across the country, face mask identification is gaining popularity. These include surveillance, monitoring entering official premises, and even across smart cities. Therefore, the main objective of drafting this research paper is to develop a masked facial identification model that can be developed with focus on Indian face mask identification.

Previously, object detection was accomplished by conventional non-neurological methods, which had a number of drawbacks, including: these traditional algorithms use multi-sliding windows that slide over the entire image, which may result in the generation of redundant region proposals and therefore makes the process very complex. Additionally, these windows were designed manually and fixed in nature. However, after the development of deep convolutional neural networks (DCNN), which has a deep structure with different layers (convolutional layers, pooling layers), feature extractors in traditional ML detectors were primarily hand-crafted, implying that feature extraction has

been done by a domain expertise, deep learning-based methods for object detection fared better than traditional algorithms because of their automatic extraction of low-to highly complex features, enhanced precision, DL-based algorithms exhibit remarkable performance on big training datasets due to their increased speed. To obtain greater accuracy, data augmentation techniques are typically employed to artificially enlarge small datasets.

However, due to:

- 1 the variety and shapes of masks
- 2 camera pixel variation
- 3 blockage degree variation (e.g., illumination, rotation, resolution, angle of view, etc.)
- 4 restricted dataset, face mask identification with image processing algorithms is still difficult.

People wear a variety of fashionable face masks these days. Certain masks feature features like lips, noses, or chins, while others feature intricate designs in an array of hues. It is therefore difficult for a face mask recognition system to identify a person wearing their customised mask because the majority of the work that has already been done is mostly focused on finding surgical masks. It can cause confusion in the system between the mask and the genuine face, leading to more false positives.

Deep learning techniques are built on top of a standard convolutional neural network (CNN), which results in significant performance gains. Additionally, many CNN architectures are now commonly utilised as the network backbone for face mask detection algorithms, such as AlexNet (Krizhevsky et al., 2017), VGGNet (Simonyan and Zisserman, 2014), GoogLeNet (Szegedy et al., 2015), ResNet (He et al., 2016), DenseNet (Huang et al., 2017) and MobileNet, etc.

Despite the relatively high accuracy produced by ordinary CNN models, each model has its own set of drawbacks. Because of their smaller designs, AlexNet and VGGNet find it difficult to grasp intricate feature correlations in faces that are obscured. Due to its emphasis on bigger receptive fields, GoogleNet and Inception Series may miss small cues in masked regions. ResNet may be able to propagate occlusion-related errors due to its residual connections. DenseNet: veiled facial patterns may not be fully adapted to because to its extensive connectivity.

Furthermore, disparities in databases can result in the creation of biases throughout the model's training process. In order to make the model more consistent, an internal database containing masked Indian faces was incorporated.

To overcome these challenges, the proposed model works in a two-stage fashion. The input goes through the first stage of pre-processing, which is based on an image cropping approach with the multi-task cascaded convolutional neural network (MTCNN) applied over it. The output from this stage moves to the CNN stage. At this point, the image is trained on two models, namely convolutional block attention module (CBAM) and modified FaceNet architecture (MFNA). Special techniques such as channel attention, spatial attention, batch normalisation, leaky rectification and face embedding vector normalisation, triplet selection are applied to reach an optimal level of accuracy. The robotic process algorithm (RPA) engine using Bayesian optimisation for hyper-parameter tuning is put in place to identify the explicit optimal values for the hybrid parameters in the model and subsequently achieve maximum level of precision.

The importance of CBAM seldom do face recognition algorithms based on images enhance their features during training; instead, many of them rely on common network architectures as their foundation. SphereFace (Liu et al., 2017) employs a 64-layer residual-style architecture, whereas ArcFace (Deng et al., 2018) uses ResNet-50 (He et al., 2016) and ResNet-101 as its foundational architectures. Although the number of channels in the feature map produced by ordinary CNN architectures is frequently large, this invariably leads to information duplication among the channels and even increases the danger of over-fitting. The outcome is still unsatisfactory even though there are regularisation strategies like dropout (Hinton et al., 2012) to prevent it. Moreover, convolution kernels handle various facial features in an identical manner.

The mechanism of attention, which identifies the most crucial component of an input signal, has demonstrated remarkable efficacy in the classification of images (Hu et al., 2017; Wang et al., 2017a; Woo et al., 2018), saliency detections (Kuen et al., 2016; Jian et al., 2015) and semantic segmentations (Fu et al., 2019; Wang et al., 2017b), effective for the aforementioned issues.

Owing to the growing necessity of face masks, a face mask detection system can be implemented in an actual setting to ascertain whether or not an individual is donning a mask. The following are some busy public areas where mask usage is required. It is imperative to establish an effective face mask detection system to keep an eye out for mask-wearing infractions.

- 1 Hospitals: in order to prevent infections during surgery, doctors in this critical area used to prefer wearing masks. However, in order to lower the rate of coronavirus infections, hospitals now need to install surveillance cameras to detect face masks, protecting both patients and medical staff.
- 2 Exam rooms: a face mask detection system can be deployed to prevent crowding because many students currently have to wait in line to complete their formalities before entering the exam room.
- 3 Airports and stations: not wearing a face mask is completely forbidden in one of the most restricted locations, which is airports and stations. There, the face mask system can be used to make sure that personnel and tourists are adhering to all safety regulations.
- 4 Shopping centres: to protect patrons and staff, face mask detection systems can be deployed in crowded areas such as shopping centres, supermarkets, and movie theatres.
- 5 Workplaces: following work from home, a number of businesses and educational institutions are reopening and enforcing mask wear as a requirement for everyone. As a result, a face mask detection system is necessary rather than manually verifying whether or not each individual is donning a face mask.
- 6 Social events: in order to lower the spread of disease, face mask recognition systems should be deployed at social events including weddings, rallies, and protests when wearing a mask is required.
- 7 Crime investigation: identifying suspects or witnesses by examining CCTV footage featuring masked individuals.

- 8 Border security: simplifying passport checks at airports and other border crossings for individuals wearing masks.
- 9 Automatic payments: 'face pay' technologies are being implemented in retail settings to enable safe and contactless checkout.

The remaining part of this paper is divided into three more sections, namely, methodology, results and discussions, and conclusion with future scope. In the methodology section, the working of the algorithm is demonstrated. Results and discussions section mainly consists of the outcome that the model has achieved when put in real life use cases system. The last section of conclusion and future scope encapsulates findings of this research along with the future scope of the proposed model.

2 Literature review

In current digitally oriented environment, we have witnessed a growing interest in theories and algorithms for facial recognition. The scope of research has been broadened, after the COVID epidemic, to incorporate aspects related to masked faces. Numerous ideas have evolved over time in order to fathom and solve tasks of masked face recognition, elucidated in research papers. Some of the works are highlighted in this section.

A major obstacle in the domain of facial recognition is caused by the problems such as occlusion, complex illumination, and density. In this regard, authors Zhong et al., Wu et al., Qiang et al. and Peibei et al. in their work describe neural network models based on attention mechanism and loss function to recognise faces with masks in complex environments. At the feature fusion stage, an attention mechanism is introduced to get better feature utilisation. Its effects were studied to improve the deep network models, followed by exploring the influence of various attention mechanisms bounding box loss functions. The experimental results declare CBAM significantly better amongst the attention mechanisms and CIoU the better loss function. The mAP of the model suggested in the research reached 90.96% on the validation set by utilising YOLOv5s as the benchmark model and gathering 7,958 photographs of people wearing masks and a large number of images of people without masks as a dataset. This is much better than the conventional deep learning method (Wang et al., 2022).

Hidden facial parts while wearing masks give rise to complications in the facial recognition process. Author Walid Hariri primarily focuses on solving this issue by proposing an efficient technique based on the removal of occlusion from the input image and further applying deep learning-based features. The model removes the masked region of the face and thereafter applies three DCNN namely, VGG-16, AlexNet, and ResNet-50 and extracts deep features from regions around the forehead and the eyes. The resulting features' bag is applied as an input to the last convolutional layer for quantification of features. For classification, multilayer perceptron (MLP) is applied. Maximum accuracy is not more than 91.3% (Hariri, 2022).

Authors Gagandeep et al., Ritesh et al., Puneet et al., Srijan et al., Prabhash et al., Rohit et al., Anshu et al. and Manik et al. use a few fundamental ML machine learning tools such as Scikit-Learn, Keras, OpenCV and TensorFlow with the aim of achieving the objective of detection of a person who is wearing a mask. This technique is capable of

successfully identifying the face from an image as well as a video. Subsequently, it determines if the face is with or without a mask. Optimal parameters involved in the CNN model are investigated to recognise presence of masks without generating over-fitting (Kaur et al., 2022).

Authors Hong et al., Rita et al., Su et al., Yanbing et al., Wei et al. and Giovanni et al. proposed an interesting approach of masked face detection using the power of human posture. The model uses Openpose to recognise the human skeleton and identify the facial region. Openpose along with the help of human posture recognition is used to perform reduction in spatial region and background filtering in the original input. Images clipped to the alleged face mask covered region are used to train CNN. This method improved recognition accuracy while significantly reducing the complexity of neural networks. Using photos taken in public spaces at various times of the day and from various perspectives, the system has been tested in a wide range of scenarios. In general, the system attains a recognition accuracy of 95.8% during the day and 94.6% during the night (Lin et al., 2021).

Authors Bingshu et al., Yong et al. and C.L. et al. presented a two-stage-approach for masked facial recognition. Using a broad learning system and the Faster_RCNN transfer model, the first stage aims to identify as many candidate wearing mask locations as feasible. Meanwhile, the second stage verifies the authenticity of the facial masks through a variety of methods. A two-class model is trained in order to implement it. Additionally, this paper suggests 7,804 realistic photos as part of a dataset for wearing mask detection (WMD). The suggested strategy outperforms the compared approaches, achieving an overall accuracy of 91.13% for the complex scenario and 97.32% for the basic scene (Wang et al., 2021).

Authors Hongxia et al., Zijian et al., Guanyu et al., Xindong et al., Haifang et al. and Gang et al. proposes a large margin cosine loss (MFCosface) based algorithm for recognition of masked faces. Due to limited availability of dataset, an image generation algorithm is applied to recognise the key facial features. After the detection of face from an image, it is sent into a multi-task cascaded convolutional network for the detection and recognition of important features of the face. According to the positional information, a mask template is selected for coverage generating the corresponding masked facial image. The model uses the loss function of a large margin cosine loss for the purpose of training the model. It is able to map all the sample features in the feature space. An Att-inception module is put into place to allow more attention to the non-masked region. To increase the weight of the unoccluded area, this module, amalgamates the Inception-Resnet module with the CBAM. Hence, enlarging its contribution (Deng et al., 2021).

Since the COVID-19 pandemic, people at large wear masks for protection. Detection of unmasked people should be done automatically; otherwise it is a complete waste of resources. Authors Sahand et al., Haniyeh et al. and Ali et al. in their paper present two models for real-time recognition of unmasked and masked faces. The first model applies an object detection method to identify and classify the faces with and without face masks. The second one uses a You Only Look Once (YOLO) face detector which helps to identify the masked and unmasked faces. These are then classified into categories using the CNN architecture (Abbasi et al., 2021).

Authors Alanoud et al., Ahmad et al. and Firas et al. have enunciated a deep learning model for accurate detection of people wearing face mask. The technique uses multi-graph convolutional networks (MGCN) which includes several convolutional filters

to produces a multi-graph structure, including many convolutional filters. A 4D facet tensor is utilised as a function of the input. To capture numerous facial expressions, a convergence layer is put in place. The image is thus transformed with the help of this multi-graph version into graph rows and column eigenvalues. The model includes other layers such as max pooling, dropout, and softmax as well (Alguzo et al., 2021).

The work published by Sanghyun et al., Jongchan et al., Joon et al. and In So et al. is referred to in order to understand the working and functionality of CBAM to consistently enhance the classification and detection performance of various methods and techniques. They propose using the max-pooled features in addition to the average-pooled features, which will yield finer attention than SE, and validate the CBAM through extensive tests on ImageNet-1K, MS COCO detection, and VOC 2007 detection datasets (Woo et al., 2018).

While the Coronavirus, COVID-19, unfurled the traditional techniques of facial recognition, such as manual inspection, and posed challenges. To address the point at issue, authors Naeem et al., Ali et al., Mustansar et al., Abdulmajeed et al. and Sami et al. proposed a robust system capable of masked and unmasked facial recognition using the DeepMasknet framework. They even developed a unified and diversified dataset to compute the performance of various masked and unmasked face recognition techniques. However, the dataset developed is limited in size. The model's architecture includes six convolutional layers and four fully connected layers, resulting in a total of ten learned layers. Initially, the model categorises masked and unmasked faces and then detects unique persons in each category individually. The performance of the DeepMasknet framework is compared to that of eight other comprehensive deep learning models, including Darknet 19, Resnet 18, Darknet 53, VGG 19, and Shufflenet, among others, in the interest of assessing the effectiveness of the DeepMasknet technique presented in the paper (Ullah et al., 2022).

With the focus on masked facial recognition, authors Yande et al., Kun et al., Yonggang et al. and Li et al. proposed a CBAM integrated with a cropping-based approach. Different cases are taken into consideration in the paper, where CBAM is applied to the regions around the eyes and optimal cropping levels are explored for each case. For the purpose of training the model, two application scenarios are used wherein masked faces are utilised to train the model to recognise unmasked faces and vice-versa. The authors, through their research, found high performance of attention mechanism in the case of occlusion face recognition (Li et al., 2021).

Facial recognition is a key requirement in the space of security. Almost every organisation has places with restricted entry due to reasons of sensitivity and confidentiality. Authors F. Ennaama, K. Benhida, and Essalki have presented an original approach to face recognition systems. It is primarily based on previously used skin detection techniques. The proposed approach yields result even when a major part of the face is covered, i.e., masked facial recognition. At the pre-processing stage, the model recognises the non-occluded along with the occluded regions of the face and then determines the face by identifying the skin colour. A protective face mask is applied on the image, followed by the application of Eigenfaces technology to enhance the functionality. A total of 43 photographs are used, which puts a limitation on the overall usability of the model (Ennaama et al., 2022).

The faces captured through camera/video are analysed by applying an AI technique on them. Author O.P. Gupta in his research work augmented the available datasets and

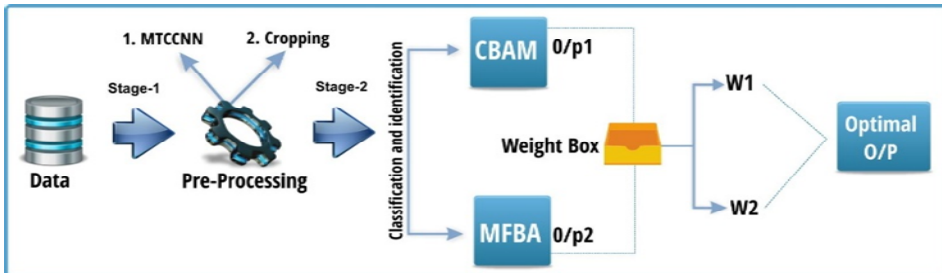
performed masked face detection and recognition using the deep learning algorithms – a combination of models, namely VGG-16, FaceNet, MLP, and support vector machine (SVM) classifier in cascade. The proposed model is a five-layered approach wherein various facial landmark points are recognised over which the facial extraction layer extracts deep facial features. The classification layer identifies the individual's face with the help of SVM and MLP. Weights are assigned to the two pipelines to generate an optimised output.

3 Methodology

When the face is covered with a mask, leaving only primary facial features visible for detection, it poses an obstacle to accurately identifying the person. This predicament extends to all the facets of masked facial recognition. There are a very few standard face algorithm systems that work well with this system, but the accuracy level of any such algorithm is contingent on the nature of the dataset it has been trained and tested on.

Figure 1 describes the workflow of the algorithm proposed in this paper. Our methodology comprises of three phases, namely, dataset creation, pre-processing, and model construction. In the first phase, we first addressed the issue of the lack of datasets available. We have constructed an in-house dataset and augmented it, and the other available datasets, using numerous distinctive techniques. It was imperative to solve the issue of exiguous datasets before proceeding towards the pre-processing and model construction stages since the efficiency and applicability of the model output chiefly depend on the size and scope of the dataset used. In the second phase, the model goes through the phase of pre-processing, where the images are transformed into understandable and usable forms from raw data through an iterative process. In the third phase, which is the model construction phase, we build the model algorithm that is trained and then tested on real time images. We will go through each of the three phases separately with the purpose of imparting better model understanding.

Figure 1 Model architecture (see online version for colours)



3.1 Pre-processing stage

Manipulation or dropping of data holds a great significance value to ensure and magnify the performance of any model. The data entering a model has to be transformed to be successfully read and understood by the algorithm. In the pre-processing stage, image is reengineered with the help of multi-task cascaded convolutional networks (MTCNN) and

image cropping. This task is performed so as to increase the quality of data and extract its full potential to elicit an efficient and accurate output.

3.1.1 Multi-task cascaded convolutional networks

There exists a plethora of cases where the images received for facial detection does not just contain the face of the person to be detected and identified. In such a case, detecting the face from the image is of vital importance. This brings us to the use of MTCNN, which is capable of detecting face and facial landmarks from an input image.

MTCNN is composed of three neural networks connected in a cascade. The algorithm follows a methodical strategy wherein it locates the image, reads it, creates an object, feeds the image through a detector to get the resultant box, and plots the box on the image. In a nutshell, MTCNN returns the pixel values, or the coordinates, where the face is detected in the image. Advanced extractions locate other facial landmarks, called 'keypoints', such as eyes, mouth, and nose.

3.1.1.1 Face classification

A binary classification is applied, which takes the help of cross entropy loss (Kaur et al., 2022):

$$L_i^{\text{det}} = -(y_i^{\text{det}} \log(P_i) + (1 - y_i^{\text{det}})(1 - \log(P_i))) \quad (1)$$

where y_i^{det} is the grand truth label and P_i is the probability produced by the network. Also, $y_i^{\text{det}} \in \{0, 1\}$.

For tasks involving bounding box regression and facial landmark localisation, the MTCNN employs the cross-entropy loss. It is a metric for the degree of difference between two probability distributions. When it comes to MTCNN, the ground truth probability of a face being present at a given place is represented by the second probability distribution, whilst the predicted probability is represented by the first. The total of the negative log-likelihoods of the ground truth labels for every pixel in the image is the cross-entropy loss.

3.1.1.2 Bounding box regression

Euclidean loss is employed to calculate an offset between the nearest ground truth and the candidate, for every candidate window with the learning objective of a regression problem (Kaur et al., 2022).

$$L_i^{\text{box}} = \|\hat{y}_i^{\text{box}} - y_i^{\text{box}}\|_2^2 \quad (2)$$

where $\hat{y}_i^{\text{box}}$ is called the target which is obtained from the network and y_i^{box} is known as the grand truth coordinate.

3.1.1.3 Facial landmark localisation

With Euclidean distance as the loss function, facial landmarks localisation is calculated as a regression problem (Kaur et al., 2022):

$$L_i^{landmark} = \|\hat{y}_i^{landmark} - y_i^{landmark}\|_2^2 \quad (3)$$

3.1.2 Image cropping

A large number of real-time factors have ascendancy over masked facial recognition systems such as occlusions, expressions, and noises. The complexity of the model to handle facial recognition expands to large amounts while factoring in unpredictable elements like occlusion shape, occluded facial region, occlusion position, and occluded region colour. To solve this problem, a model that is robust to a wide range of conditions must be implemented.

The variation in shapes, sizes, colours, and types of face masks used by people makes unambiguous detection almost impossible. It is hard to precisely detect face masks since they have different shapes and orientations. Several regions on the sides of the face mask that are informative for facial expression recognition classification are repudiated. We have put forward a novice way and narrowed down the region of the face with a mask by taking inspiration from the lightweight attentional convolutional network proposed in Minaee et al. (2021) and the visualisation technique proposed in Zeiler and Fergus (2014). An unmasked/masked binary classification is trained, which is even used for the generation of attention heatmaps.

The network of spatial transformer is exploited as an attention mechanism to emphasise significant regions of the face. The input from block $n - 1$ is warped to the output of block n , where both, block n and block $n - 1$, refer to the adjacent blocks. These blocks are composed of convolutional layers, the max-pooling layers, and the rectified linear unit activation function. For generation of heatmaps, which has focus on the facial region around the mask, a square patch of size $N \times N$ is zeroed out. It starts from the top-left corner of an image, and then prediction is made with the help of the model trained of varied images.

The facial expression identification classifier uses patches that aids in prediction of masked and unmasked regions. These patches are considered important to classify masked and unmasked faces. An attention heatmap, S_n , is obtained around the region of mask on the face. This marks the most vital space while differentiating masked and unmasked faces. The attention heatmap is up sampled and the result is fed to the sigmoid function. This process converts every pixel in the S_n in the range of $[0, 1]$. Subsequently, an attention weight map for the regions surrounding the mask on face is obtained. This attention weight map, H_n , when subtracted from 1 gives a reverse attention weight map, H_v . This is denoted as below:

$$H_v = 1 - \text{Sigmoid}(H_n) = 1 - \text{Sigmoid}(\text{up sample}(S_n)) \quad (4)$$

3.2 Model construction-test and train

Masked facial recognition model development and deployment is a multi-stage process involving a series of inter-connected activities. Following data collection, data augmentation, and data preparation, the next phase in the data pipeline appertains to

formulation of an adroit model capable of handling all the nuances in the masked facial recognition sphere. The algorithms in the model assist in recognising the patterns in data to draw conclusions that emulate the expertise of the human brain. In essence, the construction of a masked facial recognition model follows three steps:

a Modelling:

The first step is to develop the complex layers of an algorithm which interprets data successfully to take an informed decision precise enough to define it as a surrogate of human expertise. Figure 2 describes the high-level architecture of our model which interacts with two algorithms, namely MFNA and CBAM, simultaneously and monitors the efficiency of the output of the two algorithms with the help of a weight box. The best of the two results defines and fabrics the final model output.

b Modelling training:

The second step engages in training the masked facial recognition model developed thus far. At this stage, it is essential that the model is trained, in iterative loops, on abundance of data to ensure superlative level of accuracy. The outcome of this stage should reflect that the model is behaving as per expectations and desires.

c Inference

The third and final step is that of inference, wherein the masked facial recognition model is deployed in real-world cases. Based on the available data, the model routinely infers logical conclusions.

3.2.1 Convolutional block attention module

The fundamental shift in culture to embrace the new touchless facial detection module has given rise to an approach that conjoins theory and method: CBAM. The rich representation power of CNNs, has metamorphosed the vision tasks performance. With improvements in the three factors of networks – width, depth, and cardinality – CNNs yield better results. The proposed model is ten layers deep. When the model is predominantly trained on a specific image type, but the real-time datasets contain a small set of images which are slightly dissimilar from the images used to train the model, the problem of covariate shifting arises, which is taken care of by using batch normalisation in the model. The problem of dying rectified linear (ReLU) is manoeuvred by using a leaky Relu activation function.

A significant advancement in attention, which guides where to focus, along with refinement in the representation of interest, was investigated and added to the proposed architectural design of the CBAM. The attention mechanism of the CBAM model focuses only on the key features and suppresses the non-essential features. The module is designed to extract informative features by emphasising on two principal dimensions, which are, channel axes and spatial axes. The channel and attention modules are sequentially applied to the branches, to effectively accomplish extraction of essential features. They comprehend the ‘what’ and the ‘where’ to attend to the channel and spatial attention, respectively, as represented in Figure 3.

Figure 2 High level architecture of the model (see online version for colours)

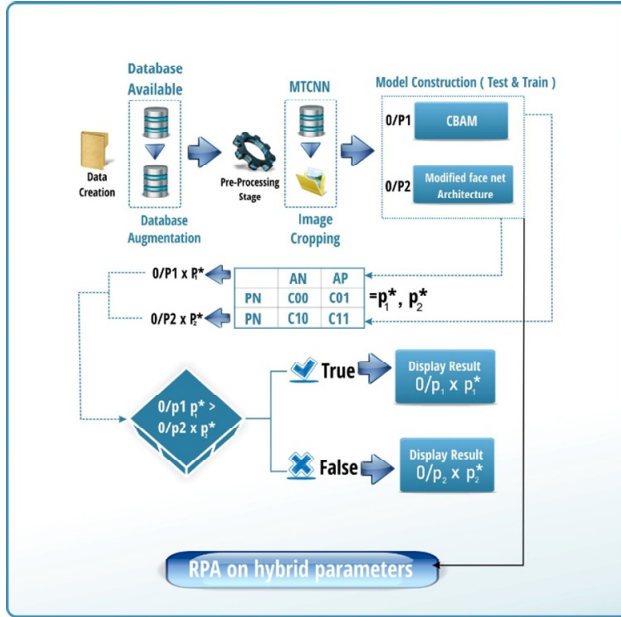
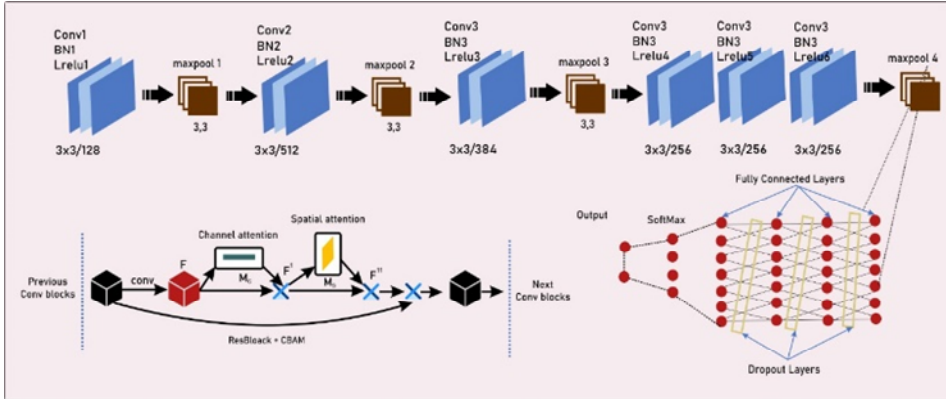


Figure 3 CBAM architecture (see online version for colours)



3.2.1.1 CBAM architecture details

To generate feature maps, convolutional operations are performed on the image data by convolutional layer, resulting in an output feature map, which can be formulated as below (Ullah et al., 2022):

$$h_{ij}^m = \sum ((w^m \times x) + b_m) \quad (5)$$

where m^{th} layer is represented by m , value of the feature is represented by h , pixel coordinates are denoted by (i, j) , convolution kernel of the current layer is represented by

w^m , bias is represented by b_m . To improve the output of the model, we have applied leaky ReLU instead of ReLU. This is done in order to specify an immensely small linear component of x as the ReLU activation function instead of declaring it 0 for those input values (x) that are negative. The resulting activation function is denoted as follows:

$$f(x) = \max(0.01 \times x, x) \quad (6)$$

The technique returns x for a positive input, returns a very small value (1/100 times of x), if the input is negative. As a result, negative values are also recorded in the output. For the purpose of minimising the spatial size, procedures of down-sampling are performed utilising the layers of max-pooling.

$$f(x) = \max(x_1, x_2, x_3, x_4, \dots, x_k) \quad (7)$$

where $f(x)$ is a feature vector which is optimised. The down sampling procedure of max-pooling utilises kernel (m) and stride (s) for the extraction of largest value from the current neighbourhood on an image of size $h \times w$. This is based on the kernel size.

Figure 4 The two sequential channel and spatial sub-modules of CBAM (see online version for colours)

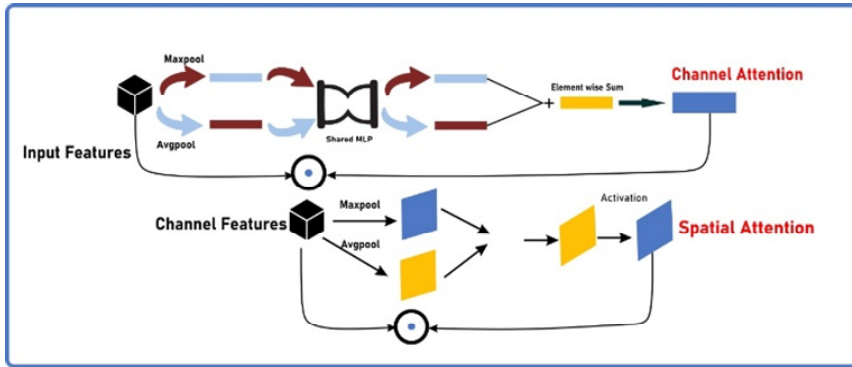


Figure 4 depicts the two modules of CBAM. Convolutional block attention mechanism aims to concentrate on more feature maps, which are influential, from channel and spatial dimensions (Woo et al., 2018). For an input feature map $G \in R^{C \times H \times W}$, let $M_c \in R^{C \times 1 \times 1}$ denote a map of channel attention, and $M_s \in R^{1 \times H \times W}$ denote a map of spatial attention. The process of feature refining is represented as follows:

$$\begin{aligned} G' &= M_c \otimes G, \\ G'' &= M_s \otimes G' \end{aligned} \quad (8)$$

where $\otimes$ implies multiplication element-wise.

G is the input feature map, with dimensions C (channels) $\times H$ (height) $\times W$ (width).

M_c is the channel attention map, obtained by applying a CNN to F and reducing its dimensions to $C \times 1 \times 1$. This map assigns weights to each channel, indicating their relative importance.

$\otimes$ is the element-wise multiplication operator.

M_s is the spatial attention map, obtained by applying another CNN to the output of the first attention module G' and reducing its dimensions to $1 \times H \times W$. This map assigns weights to each spatial location, highlighting informative regions.

During multiplication, the attention values are broadcasted (copied) accordingly: channel attention values are broadcasted along the spatial dimension, and vice versa. G'' is the final refined output.

- Module of channel attention: the inter-channel relationship of features is brought into focus by the channel attention module. The global average-pooling and max-pooling operations are utilised for the aggregation of a feature map's spatial information, in the CBAM model. This generates the 2 spatial information descriptors: $G_c \text{ avg} \in R^{c \times 1 \times 1}$ and $G_c \text{ max} \in R^{c \times 1 \times 1}$. These two descriptors are fed into a shared multi-layer perceptron (MLP). The final channel attention map: $M_c \in R^{c \times 1 \times 1}$ is generated by 1 layer which is hidden.

$$\begin{aligned} M_c &= \sigma(MLP(Avg \ Pool(F)) + MLP(Max \ Pool(G))) \\ &= \sigma(W_1(W_0(G_{avg}^c)) + W_1(W_0(G_{max}^c))); \end{aligned} \quad (9)$$

where σ denotes a sigmoid function, $W_0 \in R_{c/r \times c}$, $W_1 \in R_{c \times c/r}$.

- Module of spatial attention: the inter-sparse relationship of features is brought into focus by the spatial attention module. The aim is to explain 'where' is the more influential feature among the entire feature map. Like in the channel attention module, explained earlier, the global average-pooling and max pooling operations are adopted to generate the information descriptors. These two operations are along with the channel axis, and generate two 2D maps, namely, $G_s \text{ avg} \in R^{1 \times H \times W}$ and $G_s \text{ max} \in R^{1 \times H \times W}$. These two maps are the concatenated together. With the adoption of a convolution layer, the final spatial attention map is generated, which is $M_s \in R^{1 \times H \times W}$. The following is the computation process for the spatial attention map:

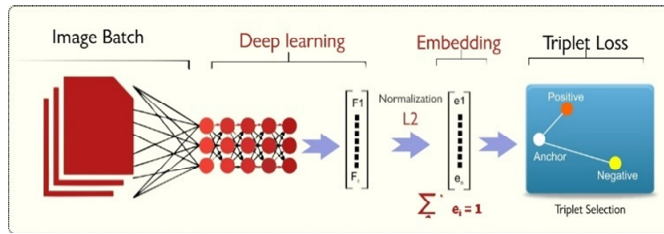
$$\begin{aligned} M_s &= \sigma(f \otimes ([Avg \ Pool(G); Max \ Pool(G)])) \\ &= \sigma(f \otimes ([G_{avg}^s; G_{max}^s])) \end{aligned} \quad (10)$$

where σ denotes a sigmoid function, $\otimes$ represents the process of convolution, and f is a convolution kernel.

3.2.2 Modified face net architecture

The MFNA is a neural network method for the recognition, verification, and clustering of faces. The neural network is 22 layers deep, which works in order to train the output to be a 128-dimensional embedding. The triplet selection loss function is applied in the last layer instead of a typical triplet loss function.

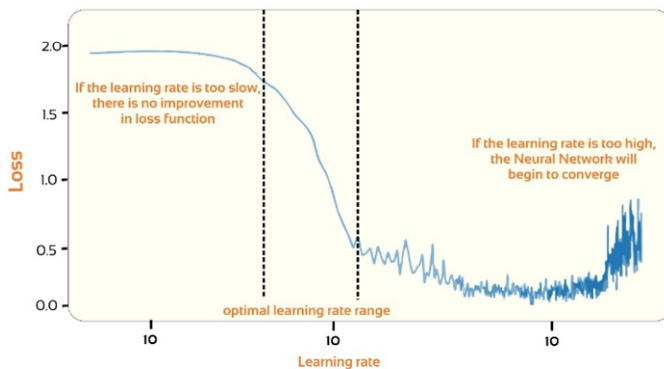
This system of facial recognition was proposed in a paper published by the researchers at Google in 2015 and was able to produce remarkable results with various famous benchmark datasets. The approach has the potency to generate superlative quality face mapping with the assistance of input images. The triplet loss function is applied as a loss function for the purpose of training the algorithm. We have applied a triplet selection for a better outcome in our model. Figure 5 represents the architecture of the FaceNet model.

Figure 5 FaceNet architecture (see online version for colours)

3.2.2.1 FaceNet architecture details

FaceNet architecture involves end-to-end learning. The underlying architecture used by the model is either the inception network or ZF-Net. The number of parameters is decreased by the addition of numerous 1-1 convolutions in the model. The FaceNet architecture generates an output with the image embedded $f(x)$. On the embedded output images, L2 normalisation is performed. The resultant embeddings are passed into the triplet loss function for the calculation of loss. With the use of triplet loss function, we aim to reduce the squared distance between two image embeddings (independent of image condition and pose) for the identities which are same. We also wish to increase the squared distance between two image embeddings for the identities which are different. The goal of incorporating the triplet loss function into the architecture is to enforce a margin between faces of the identities which are different.

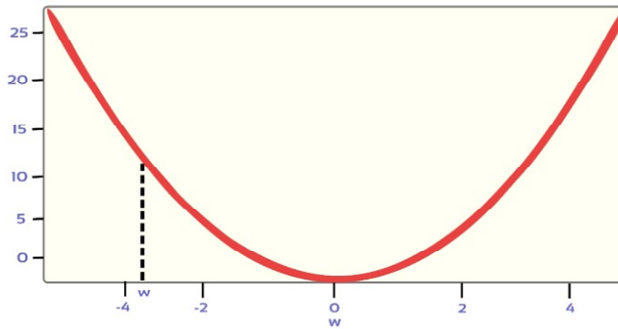
- *Learning rate*: the change between iterations is weighed with the help of a learning rate in a neural network. It is a salient feature of the model, and it is of paramount significance to determine its optimal value. Figure 6 is a graphical representation of the learning rate which depicts that large swings between the weights may be constituted by a large learning rate and a low learning rate, although considered better, will constitute more iteration to converge.

Figure 6 Graphical representation of optimal position of learning rate (see online version for colours)

The learning rate hyperparameter is a backpropagation step which is a representation of the importance of change in weight after re-calibration. To understand the process

involved in determining the optimal value of the learning rate, let us consider a loss function with just one weight as shown in the graphical representation in Figure 7.

Figure 7 Loss function graph with one weight (see online version for colours)

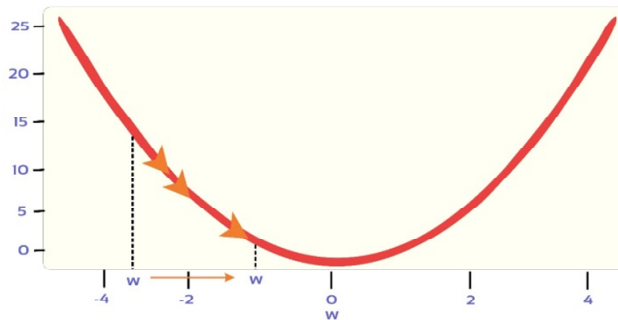


To minimise the loss, ideally, we need to slide the current 'w' towards in minimum. Figure 8 depicts a graph with the loss function sliding towards minimum. The corresponding function is:

$$w'_i = w_i - \frac{\delta L}{\delta w_i} \quad (11)$$

where w_i is the current weight, the function gradient is denoted by $\frac{\delta L}{\delta w_i}$ (for this case of 1-D, it is the first derivate of the loss function). When the tangent segment's steep is negative, the then first derivate, T , will have a negative value. Due to this, a minus is put in the middle of the two terms (for example, the weight should ideally move in the right direction, if the steep is negative). The procedure of optimisation is known as gradient descent (Li et al., 2021).

Figure 8 Graph of loss function towards minimum (see online version for colours)



After addition of the learning rate, the resultant formula is as follows:

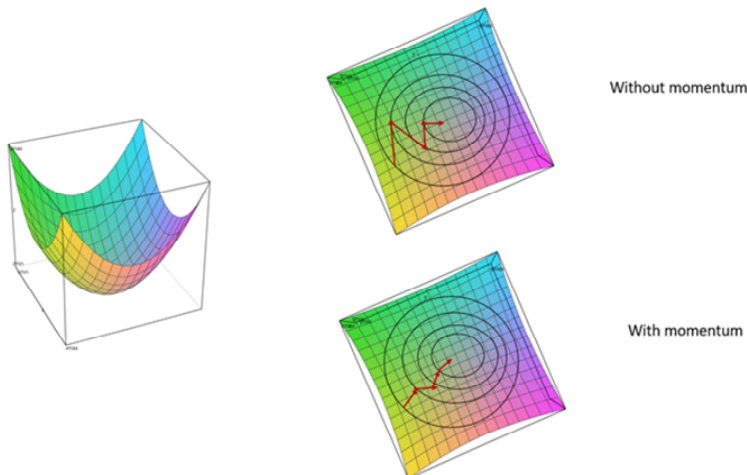
$$w'_i = w_i - \gamma \frac{\delta L}{\delta w_i} \quad (12)$$

The gamma added in the above formula is the learning rate, and it informs the algorithm the importance of impact of a gradient on weight. A small value of gamma converges the neural network very slowly, and the problem of ‘vanishing gradient’ may be incurred. On the other hand, if gamma has a large value, there is a risk of missing the minimum, which will lead to the problem of an ‘exploding gradient’. We strategically started from a value of 0.1, and then exponentially reduced it. For initial iterations, the loss function’s value started to decrease. This is the signal for the movement of weight in the correct direction. By applying RPA in a range of 0.001, 0.01, 0.1, and 0.2. It was determined that 0.01 is the learning rate’s optimal value.

- *Momentum*: momentum is analogous to the mass of a ball that is rolling down the surface of the loss function. The heavier the ball, the quicker the ball falls. But if it’s too heavy, it can get stuck or overshoot the target.

Momentum is a technique that is applied during the phase of backpropagation. The process can consume a lot of time and may affect the algorithm’s efficiency. One of the possible solutions is to track down the previous directions (i.e., the loss function gradients with respect to the weights). This should be kept as an embedded information. It escalates the convergence speed in terms of past re-calibration’s embedded memory (the algorithm is able to comprehend the weight of previous direction and proceeds in the same direction directly for the next propagation) and not from learning rate’s perspective. It can be visualised if the projection of 2-weight loss function (specifically, a paraboloid) is considered:

Figure 9 Loss function with and without momentum (see online version for colours)



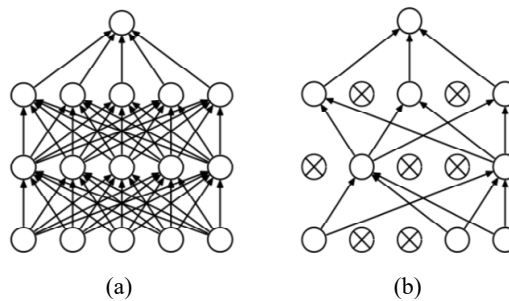
In Figure 9, we can clearly visualise that the addition of a momentum hyperparameter results in a faster descending phase. This happens because the model is keeping a track of past gradient directions.

If it is decided to go for high momentum values, it means that it will heavily consider previous directions: this may result in an algorithm which is incredibly fast learning. But, in such a case, there is generally a high risk of missing some correct

‘deviations’. Hence, it is suggested that it should start with low values and then increased subsequently.

- *Dropout rate:* to incorporate regularisation in neural networks, dropout layers are added into the model. The techniques functions in a manner such that some unnecessary and redundant nodes are removed from the neural network so that the neural network is not too heavy and is not overwhelmed by information that does not contribute to the efficiency of the output. It is generally implemented in the model’s training phase itself. The dropout rate defines the input to drop fraction as a precaution for overfitting. At each training stage, the algorithm decides the nodes to keep, each with probability o (known as ‘keep probability’) or drop the nodes with probability $1-o$ (known as ‘drop probability’). Recommended range: 0.2–0.5. Figure 10 represents graphically neural networks with and without dropout layers (Li et al., 2021).

Figure 10 Graphical representation of neural networks with and without applying dropout, (a) standard neural net, (b) after applying dropout



- *Activation function:* the weighed sum in the neural network is passed through the activation function. The purpose of applying an activation function is to get a significant resultant output, known as the probability vector or an output of 0 and 1. The prime activation functions are Sigmoid (SoftMax function is the variant of such function, generally applied in multiclass classifications – it returns an output as a probability vector with sum equal to 1), ReLU, Tanh, and leaky rectified linear activation function (LReLU). Figure 11 depicts the activation functions in graphical format. The activation functions can be located anywhere in the neural networks considering efficiency and velocity. For example, the ReLU and LReLU functions are quick from the perspective of training, whereas the Sigmoid function takes more time and is complex. The results of RPA applied on the activation functions resulted in that LReLU should be used for the hidden layers and Sigmoid should be applied in the last layer to yield better results.
- *Batch size:* sometimes, if the neural networks are fed with mammoth amounts of data, the results are counterproductive and inefficient for practical use. Hitherto, feeding the neural networks with small samples of data, called batches, is a good policy. In such a case, the algorithm trains itself on samples of same data batch size. The desired size range is 32 or higher, the sample batch should not be too large in size to avoid the risk of an over generalised model (Li et al., 2021). By applying RPA, it was determined that the optimal value of minibatch size is 20.

- *Epochs*: it is a representation of the number of times the algorithm trains itself on the entire dataset available. It is important to note that epochs are not iterations. Iterations correspond to number of batches which are required to complete an epoch. It is dependent on the kind of data and task defined by the model. There are various approaches to applying epochs to the model, like staggering the condition to stop epochs when the error approaches zero or starting with low no. of epochs and gradually increasing it by keeping track of some accuracy evaluation metrics (Li et al., 2021). We applied RPA for the epoch values of 8, 16, 32 and determined that 32 gave the optimal results.
- *Stride and padding*: stride is the rate at which the kernel passes over an input image and Padding depicts the addition of layers of 0, so that the kernel passes over the edge of the input images. On applying RPA to this hyper parameter, it was found that the optimal value for stride is 1 (Li et al., 2021).
- *Number of layers in the model*: complexity in the neural networks is defined by the number of layers in it. A problem of overfitting occurs when there are too many layers since the model is allowed to learn too much information from the training dataset. The model's learning ability is limited when there are very few layers, leading to the problem of underfitting. We have tried the CNNs with two, four, six, and ten layers, and the ten-layer model has performed well (Li et al., 2021).
- *Number of units per layer*: the transformation information performed by a layer is held with the units in a layer. This information is driven by the number of features, for the first layer. In all the subsequent layers, the number of units is dependent on the choice of contracting or expanding the representation from the previous layer. The goal is to minimise the information which is lost between the layers. We have tried the unit values in the range of 8, 16, 32, 64 and 64 units have worked well for our model (Li et al., 2021).

Figure 11 Activation functions – ReLU, Sigmoid, Tanh, and LReLU (see online version for colours)

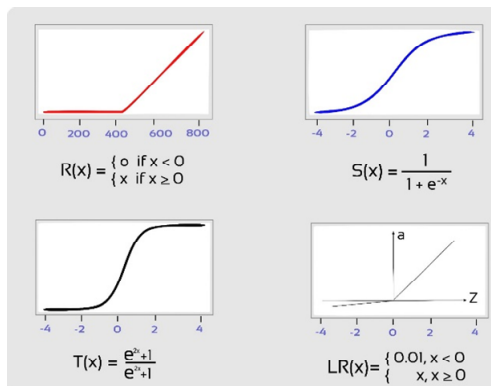


Table 1 The range and optimal values of hyperparameters

<i>Hyperparameters</i>	<i>Range</i>	<i>Best value</i>
Learning rate	[0.001, 0.01, 0.1, 0.2]	0.01
Activation function	[Sigmoid, ReLU, tanH, LReLU]	LReLU
Batch size	[5, 10, 20]	20
Epochs	[8, 16, 32]	32
Stride	[1,2]	1
No. of layers in the model	[3, 4, 6, 10]	10
No. of units per layer	[8, 16, 32, 64]	64

3.3 Assigning weights to the modules

The comparisons between alternatives for which the parameters are simple, priorities have to be assigned, clear reasoning for the decision has to be provided, and logical fallacies need to be taken into account. This brings us to the last phase of the model which deals predominantly with assigning weights to the output of CBAM and MFNA. This is done to reflect the competencies and the output of the two models for the respective set of input images and finally select the outcome with better efficiency.

In this study, we propose a technique with the ability to reckon with multi attributes and gauge the better result of the two models. During the training phase, every time an input image is processed by the two models, namely CBAM and MFNA, the accuracy of the output is compared with the input. The result is stored and categorised into the following 4 categories in the form of a matrix as depicted in Table 2.

- C_{00} : Predict negative – actual negative.
- C_{01} : Predict negative – actual positive.
- C_{10} : Predict positive – actual negative.
- C_{11} : Predict positive – actual positive.

Table 2 Result matrix

	<i>Actual negative</i>	<i>Actual positive</i>
Predict negative	C_{00}	C_{01}
Predict positive	C_{10}	C_{11}

To explore the better outcome of the two models, the optimal threshold for making decisions is determined by calculating the value of p^* for the respective models:

The output of CBAM is:

$$p_1^* = \frac{C_{10} - C_{00}}{C_{10} - C_{00} + C_{01} - C_{11}} \quad (13)$$

The output of MFNA is:

$$p_2^* = \frac{C_{10} - C_{00}}{C_{10} - C_{00} + C_{01} - C_{11}} \quad (14)$$

The final selection is made with the help of an if-else loop:

- If $p_1^* > p_2^*$, the output from CBAM is selected as the final output of the model.
- Or else, the output from MFNA is selected as the final output of the model.

3.3.1 Triplet loss

Triplet loss aims to make the embedding of the anchor and the positive closer in the embedding space than the embedding of the anchor and the negative. This means embeddings of similar faces should be clustered together, while those of different faces should be further apart.

- Anchor (a): the reference image you want to compare others to.
- Positive (p): an image of the same identity as the anchor (a ‘matching’ image).
- Negative (n): an image of a different identity from the anchor (a ‘non-matching’ image).

An image’s embedding are represented by $f(x)$ such as $X \in R$. The embedding is in the form of a vector (128 size). It is normalised in the following manner (Abbasi et al., 2021):

$$\|f(x)\|_2^2 = 1 \quad (15)$$

Now, we need to assure that the positive image of a person is very close to the anchor image of the person when compared with the negative image of the person. By positive image, we mean the image of the same person and by negative image, we mean image of some other person. This is done by:

$$\|f(x_i^a) - f(x_i^p)\|_2^2 + \alpha < \|f(x_i^a) - f(x_i^n)\|_2^2 \quad \forall (f(x_i^a), f(x_i^p), f(x_i^n)) \quad (16)$$

where α denotes the margin. To differentiate between the positive pairs and the negative pairs, this margin is enforced. The image space is represented by T . Therefore, the loss function, L , is calculated as below:

$$\sum_i^N \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right] \quad (17)$$

While training a model, if such triplets are chosen that may easily satisfy above mentioned property, then it would not help in an acceptable training of such algorithm. Hence, it is always necessary to have triplets which are able to violate the property mentioned above.

3.3.2 Triplet selection

In furtherance, to ensure quicker learning of the model, triplets that violet the above property are required. Keeping the view in consideration, for a given x_i^a we are required to choose triplets in a manner, such that $\|f(x_i^a) - f(x_i^n)\|_2^2$ is minimum and

$\|f(x_i^a) - f(x_i^p)\|_2^2$ is maximum. It is generally computationally expensive to produce triplets bases the entire training dataset. Triplet selection is used to generate triplets by selecting hard positive x_i^p and hard negative x_i^p with the help of maximum and minimum on a mini batch size (Abbasi et al., 2021).

3.4 *Robotic process automation using Bayesian optimisation for auto-hyperparameter tuning*

The selection of hyperparameters for machine learning algorithms can have a significant impact on algorithm performance, making it a laborious yet essential process to tune. Important phases in the machine learning process, such as feature engineering and result interpretation, are slowed down by manual tuning. While grid and random search are passive methods, they waste time analysing unproductive regions of the search space, which makes them take a long time to complete. More and more automated techniques are being used to tune hyperparameters with the goal of reducing search time and requiring no human labour after the first setup.

Results indicate that Bayesian optimisation, a model-based technique for determining a function's minimum, can outperform random search in the test set while requiring fewer iterations. Furthermore, a variety of Python modules are now available that simplify the process of implementing Bayesian hyperparameter tweaking.

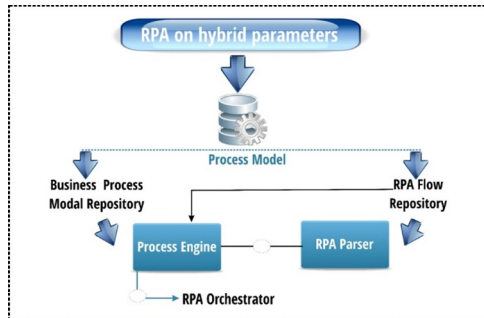
Hyperopt library is used to implement Bayesian optimisation using the TPE option.

Figure 12 represents the functional components along with the additional RPA related elements built to define the process of finding the value of hybrid parameters. The process modeller, a component of the robotic process automation system, enables the seamless visualisation and creation of the model process and robotic process automation workflows. The same is stored in their respective repositories, adapted from the traditional business process management system. The business process model repository contains definitions of business processes that are enriched by the applicability of RPA functionality. The robotic process automation flow repository is made of a sequence of end-to-end repetitive and recurring activities with the perspective of becoming automated. It intrinsically employs flowcharts of the series of activities and tasks to be automated. Furthermore, we can also define how and whether the workflow has to be implemented in an unattended or attended manner. Automating individual tasks using RPA demands detailed technical instructions, giving rise to complexity and the development of extensive process models. This is solved by drawing inspiration from the layered approach of van Nuffel and de Backer (Ullah et al., 2022). Sub-process elements are implemented in the process modeller, where the concept is broken down into three layers. The first layer defines the processing level. It usually consists of primary tasks and interactions. The level of activity and the list of important tasks are represented by second layer. The third layer poses a level of task and consists atomic instructions on the performance of an activity in particular.

The process engine executes and orchestrates the well-defined process in a manner similar to the traditional business process management system process engine. The tasks of the RPA are processed by the RPA Parser. It transforms them into a format specific to robotic process automation, which is subsequently added as an input to the RPA orchestrator for implementation and execution of activities in an environment that is

appropriately specified by the respective process variables. In the case of a non-RPA task, the bot's execution is paused until the completion of the intervening non-RPA activities by the RPA orchestrator.

Figure 12 Diagram representing the RPA work-flow (see online version for colours)



The value of the hyperparameters controls the learning algorithm of the masked facial recognition model. They are the ‘top-level’ parameters, as suggested by the prefix ‘hyper’ and they determine the value of the model parameters that result from the learning process algorithm. These elements are not updated by the model in accordance with the optimisation strategy but need manual intervention. We propose the application of RPA to comprehend the optimal values of the respective parameters. To drive the finest output, human intervention is essential to conclusively define the optimal value of the hyperparameters in connection with each other. We need the perfect combination of all the hyperparameters that work together in the finest form of conjunction to build a model that has the capability to detect faces that are covered with masks. To accomplish the rationale of the quintessential combination of hyperparameters, we have empirically incorporated robotic process automation for each combination of the following parameters:

4 Results and discussion

4.1 Dataset

Facial recognition algorithms have a profound influence on a diverse range of applications, such as security, defence, authentication, access control, healthcare, forensics, and transportation. The algorithm can be pragmatically useful only if it has better computational power and is trained on large amounts of data. The importance of training data cannot be emphasised enough. Quality training data forms a constitutional element in the work of the model. It is a process where human intervention is desired to examine and build the data for the model's learning procedures (Hariri, 2022).

In most of the algorithms already developed, the fundamental problems are not caused by the model but by the scarcity of relevant datasets. A number of times, the facial recognition models experience difficulty due to poor resolution, low-quality images, a suboptimal view angle, and poor illumination. This brings us to the importance of training the model with massive amounts of data to incorporate periodic learning of the

model accommodating such challenges. To achieve the best accurate results, we have built our own repository by taking our in-house dataset, Kaggle dataset, and others into consideration. For the purpose of creating a vast pertinent repository of data, we have augmented the datasets available. With such large datasets, this database takes a significant step towards closing the data gap prevalent in the domain of masked facial recognition.

Both Kaggle and MDMFR suffer from challenges such as lack of diversity with respect to Indian faces. Moreover, the in-house dataset adds more variations in mask types and wearing style commonly seen in India (scarves, bandanas, and other makeshift masks).

We have used the following datasets:

- In-house dataset: this dataset comprises of images of Indian people from different regions. This is deliberately done to improve the accuracy of the results, when put in test with real life use cases. There are three classes of images:
 - a Images of people with mask.
 - b Images of people without mask.
 - c Images of people wearing mask inappropriately.
- Every class has three separate masked and unmasked images. The dataset has 260 images of people wearing face masks incorrectly. Such types of images have been deliberately added to train the model and invigorate its capability of detecting people wearing masks inappropriately and accommodate a real-life use case into the model's training. Figure 13 represents some of such images. Also, the dataset encompasses images with numerous levels of resolution as well. There are a total of 360 pictures of people wearing mask and pictures of people not wearing mask.

Figure 13 Images of people wearing masks inaccurately (see online version for colours)



- Publicly available Kaggle dataset: dataset consists of 45,890 images. It comprises of three non-identical classes of images. The first type of images are of people wearing masks, the second type of images are of people non wearing masks, and the third type of images are of people who have worn masks in an incorrect manner. The dataset comprises 17,226 images of people with masks, 3,969 without masks, and 2,900 images of people wearing masks in an inappropriate manner.
- Other masked facial datasets are available: This dataset has been taken from the paper published in 2021 (Wang et al., 2018a) and is named masked detection mask

face recognition (MDMFR). It comprises of samples of 5,548 male and 457 female images. All the pictures have two configurations: with mask and without mask. There are a total of 3,174 masked images and 2831 without mask faces. Also, the types of mask used are 2,892 surgical masks, 121 cloth masks, and 164 N95/KN95 masks.

Table 3 Database details

<i>Dataset</i>	<i>No. of identities</i>	<i>No. of images</i>	<i>After augmentation</i>	<i>Availability</i>
In-house dataset	26	360		Privately
Kaggle dataset	4,798	45,890		Publicly
MDMFR	2,002	6,005		Publicly

We augment the above-mentioned datasets by adding several different masks, making the model train better. These include green surgical masks, white N95 masks, blue surgical masks, black cloth, and masks white KN-95 masks, amongst other coloured masks available in the market.

4.2 Evaluation matrices

To appraise the degree to which the proposed model is successful in delivering the desired results, we have used four parameter metrics, namely, accuracy, precision, recall, and F1-Score. Following are the computations of the same:

$$\text{Accuracy} = \frac{(TN + TP)}{TS} \quad (18)$$

$$\text{Precision} = \frac{TP}{(TP + FP)} \quad (19)$$

$$\text{Recall} = \frac{TP}{(TP + FN)} \quad (20)$$

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{(\text{Precision} + \text{Recall})} \quad (21)$$

where TS , TN , FN , TP , and FP are total samples, true negative, false negative, true positive, and false positive respectively.

In the above equations, the images that are labelled true but after prediction gave false results are referred to as true negative (TN) values. Similarly, the images that are labelled as false but after prediction gave true results are referred to as false negative (FN) values. The images that are labelled as true and after prediction by the model, gave true results are referred to as true positive (TP) values. The images that are labelled as false and gave false results after prediction are referred to as false positive (FP) values.

Since the classes are balanced, accuracy is a good measure to begin with. The measure of positive predicted values is given by precision. While the ability to find all positive samples was given to a classifier with the help of recall, the measure of test accuracy was defined by the F1-Score. In a balanced dataset, these evaluation metrics yield the best results, and this is the reason why we choose these evaluation metrics.

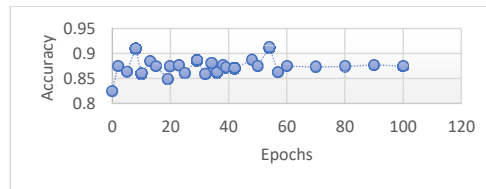
4.3 Experimental setup

A laptop empowered with the Intel processor i7-8750H (4.1 Giga Hertz), 16 Giga byte of random access memory with 1050ti max-Q with four Giga byte of video random access Memory was used for the purpose of performing experimental trials. To develop and implement the different experimental trials, software, namely, Jupyter Notebook with Python 3.9 Kernel software was made use of to carry out the research proposed in this paper. The graphs were plotted using the library matplotlib.

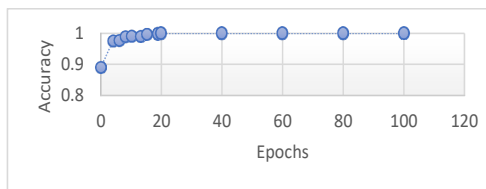
To accomplish the implementation of the research, 80% of the dataset was designated for the training of the model and the rest 20% was exploited for testing. A number of experiments were performed in series for performance evaluation of the model proposed in this research.

Robotic process automation technique was used to establish the best values for the hybrid parameters in the proposed model. By performing RPA, the learning rate was set to 0.01 with a batch size of 20. The learning rate value was set at 0.1 and gradually decreased to find the optimal value of 0.01 through RPA. Likewise, after analysing the range of all the hybrid parameters through RPA, epochs was set to 32 and stride was set to 1. Through RPA, we were able to ascertain the best activation function, which is Leaky ReLU. By the same logic, we determined the pre-eminent choice of the number of layers in the model and the number of units per layer, which are 10 and 64, respectively.

Figure 14 (a) Validation of accuracy curve (without data augmentation) and (b) Training of accuracy curve (without data augmentation) (see online version for colours)



(a)



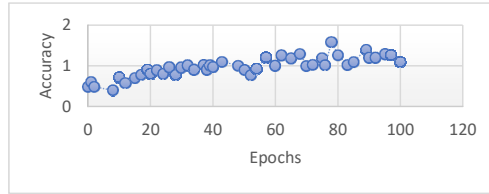
(b)

4.4 Data augmentation results

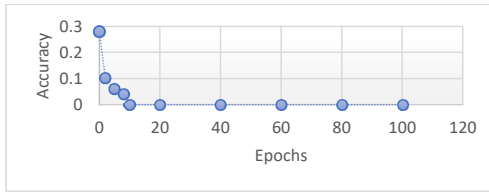
Data augmentation was initiated keeping in view the lack of availability of relevant datasets. After applying RPA on the various hybrid parameters of the model, it was observed that the results improved after the application of data augmentation on the respective datasets. Figure 14(a), Figure 14(b), Figure 15(a), and Figure 15(b) represent the results of the proposed model without data augmentation. The model shows a great decline in growth terms, when it is trained on 100 epochs, as compared to the results after

data augmentation. It was observed that till 60 epochs, the model struggled to learn features in validation accuracy without applying data augmentation, as depicted in Figure 1. The training accuracy equals 92.64% after applying data augmentation, resulting in an increase of 5.13% from the results when data augmentation was not applied. Figure 16(a), Figure 16(b), Figure 17(a) and Figure 17(b) show remarkable improvement in the results after applying data augmentation to the model.

Figure 15 (a) Training of loss curve (without data augmentation) and (b) Validation of loss curve (without data augmentation) (see online version for colours)

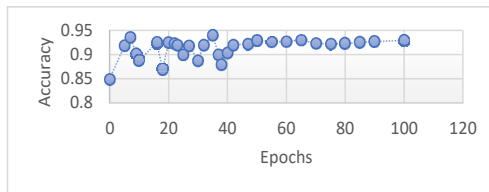


(a)

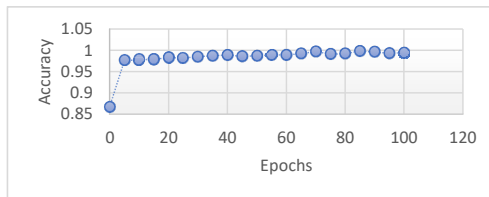


(b)

Figure 16 (a) Validation of accuracy curve (with data augmentation) and (b) Training of accuracy curve (with data augmentation) (see online version for colours)

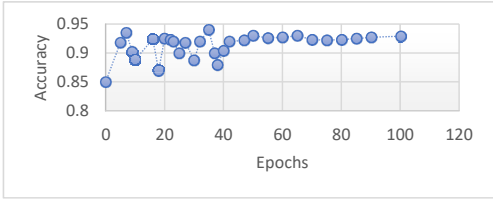


(a)

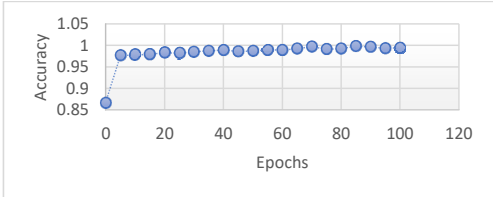


(b)

Figure 17 (a) Validation of loss curve (with data augmentation) and (b) Training of loss curve (with data augmentation) (see online version for colours)



(a)



(b)

4.5 Face mask detection

First purpose of our model deals with detecting the masked and unmasked faces. 80% of the available dataset, which accounts for 16,528 masked and 5,648 unmasked, and 2,528 inappropriate wearing of masks, was utilised for the purpose of training the model and the remaining 20%, which accounts for 4,132 masked and 7,060 unmasked, and 3,160 inappropriate wearing of masks, was utilised for testing the model results. The experimental settings, as described under the experimental setup section, remained the same for both the test and train datasets. The result of our model is depicted in Table 4. From the results, it can be concluded that the proposed model displayed optimal results with respect to precision, accuracy, F1-Score, and recall. Our model is successfully able to draw discriminating features which help in accurately classifying the masked and unmasked faces.

Table 4 Identification result with different type and colour of facemasks

	Accuracy
Facemask colour – blue	0.99
Facemask colour – white	0.99
Facemask colour – black	0.96
Facemask colour – others	0.97
Facemask type – surgical	0.99
Facemask type – cloth	0.97
Facemask type – N95 or KN95	0.99

4.6 Performance evaluation on masked facial recognition

The second purpose of our model is to recognise different people from the facial images with masks and subsequently determine the performance of the proposed model. Masked facial images and determine the performance and capability of the proposed model. For this experiment, we have considered three datasets: in-house dataset, Kaggle dataset, and MDMFR dataset. Eighty percent of the dataset is utilised for the purpose of training the model and the rest 20% comes into picture while validating the proposed model. The experimental set up is same as mentioned under the experimental section of this paper. Our proposed methodology was successfully able to achieve an accuracy of 96.7%, precision of 97.7%, recall 97.7%, and F1-Score 97.5% for recognising people from masked facial images. The results are well depicted in Table 5. The ability of the model to extract discriminative and robust features from the limited visible portion of the face, due to the applicability of mask, makes the results more alluring.

Table 5 Results of the model proposed in the paper

	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>
Non-masked	0.99	0.99	0.99	1.00
Masked	0.97	0.97	0.98	0.98
Wearing masks inappropriately	0.96	0.97	0.98	0.98

4.7 Masked facial recognition with other models

The purpose of the experiment is to compare the outcome of our proposed model with other existing models. In this regard, we have compared our model with DenseNet201, ShuffleNet, ResNet18, MobileNetv2, DarkNet 19, DarkNet 53, VGG19, Alexnet, CNN, VGG-16 and FCN, VGG-16 CNN, and ResNet 50 and YOLO v2. Table 6 compares the accuracy achieved in various models.

Table 6 Masked facial recognition comparison with other models

<i>Method</i>	<i>Accuracy</i>
DenseNet201	88.89
ShuffleNet	88.57
ResNet18	84.20
MobileNetv2	86.76
DarkNet 19	90.00
DarkNet 53	89.10
ResNet 50 and YOLO v2	81.00
Alexnet	88.89
CNN	96.00
VGG-16 and FCN	93.80
VGG19	90.91
VGG-16 CNN	96.00
Our proposed model	96.7

4.8 Cross dataset validation

We have used three datasets for experimental purposes of this paper. The idea is to comprehend the generality of our proposed model by detecting face masks over cross-dataset scenarios. This will also yield results which can potentiate our model's applicability in real-life use cases. Following six scenarios were evaluated for the purpose of cross-dataset validation:

- Trained on our in-house dataset and tested on the publicly available Kaggle dataset.
- Trained on the publicly available dataset and tested on MDMFR.
- Training on MDMFR and tested on our in-house dataset.
- Trained on the publicly available Kaggle dataset and tested on our in-house dataset.
- Trained on MDMFR and tested on the publicly available Kaggle dataset.
- Trained on our in-house dataset and tested on MDMFR.

Table 7 depicts results of the various scenarios considered for the cross-dataset validation. The precise results are a clear indication that the proposed model can be applied to real life use cases.

Table 7 Cross dataset validation results

<i>Training dataset</i>	<i>Testing dataset</i>	<i>Accuracy</i>	<i>Precision</i>	<i>F1-Score</i>	<i>Recall</i>
In-house dataset	Kaggle dataset	0.97	0.97	0.98	0.98
Kaggle dataset	MDMFR	0.96	0.98	0.97	0.97
MDMFR	In-house dataset	0.97	0.97	0.98	0.98
Other dataset	In-house dataset	0.97	0.99	0.98	0.97

5 Conclusions and future scope

The paper has proposed a model for accurate and reliable face mask detection along with facial recognition especially focusing on increasing the database size of Indian citizens that will help in not only cases like diseases such as COVID-19 (which can be a threat in the future as well) but also for professions across healthcare, construction and manufacturing, etc. that require the usage of face masks. Professionals such as doctors, nurses, policemen, government officials, construction workers, etc. are the ones who regularly use face masks. Currently, stumbling blocks that are present in the available masked facial recognition techniques lack a focus on Indian specific database. These limitations pose serious challenges in surveillance, monitoring, and even development of Smart Cities in India. With this, an in-house database was developed which included three different classes of images:

- 1 images of people with mask
- 2 images of people without mask
- 3 images of people wearing mask inappropriately.

The application of this dataset also helped in determining the efficiency and performance of the model proposed in this paper. With an accuracy of 96.7% for masked facial recognition, it can be concluded that the proposed model works well when put in a real-life environment and yields superior results when compared with other existing contemporary technologies. The experimental results of cross-validation datasets verify the robustness of our proposed model with respect to both face mask detection and masked facial recognition under conditions as diverse as variations in skin tone, lightning conditions, age, face angles, gender, types of masks, and occlusions. The results of our model have shown explicit generalisation ability for the detection of face mask and recognition of masked faces.

The results of the model were compared with existing models such as DenseNet201, ShuffleNet, ResNet18, MobileNetv2, DarkNet 19, DarkNet 53, VGG19, Alexnet, CNN, VGG-16 and FCN, VGG-16 CNN, and ResNet 50 and YOLO v2 and outperform the models on the developed dataset.

Although the model performs with high accuracies. In the future, we intend to study the implementation of the model in various situations with an even larger and more extensive dataset. It will cover all postures and environments. We will make room for identifying the results on a video stream covering at least the basic video frame rate of 24 frames per second (FPS). Furthermore, the algorithm will be improvised to incorporate the practical application of identifying and recognising masked faces with the help of a dataset that includes a set of videos as well. We will also bring emotions; positive, negative, or neutral – to the bargain and evaluate the result of the model proposed in the paper.

References

- Abbasi, S., Abdi, H. and Ahmadi, A. (2021) ‘A face-mask detection approach based on YOLO applied for a new collected dataset’, in *2021 26th International Computer Conference, Computer Society of Iran (CSICC)*, IEEE, March, pp.1–6, DOI: 10.1109/CSICC52343.2021.9420599.
- Alguzo, A., Alzu’bi, A. and Albalas, F. (2021) ‘Masked face detection using multi-graph convolutional networks’, in *2021 12th International Conference on Information and Communication Systems (ICICS)*, IEEE, May, pp.385–391, DOI: 10.1109/ICICS52457.2021.9464553.
- Deng, H., Feng, Z., Qian, G., Lv, X., Li, H. and Li, G. (2021) ‘MFCosface: a masked-face recognition algorithm based on large margin cosine loss’, *Applied Sciences*, August, Vol. 11, No. 16, p.7310, DOI: 10.3390/app11167310.
- Deng, J., Guo, J., Yang, J., Xue, N., Kotsia, I. and Zafeiriou, S. (2018) *ArcFace: Additive Angular Margin Loss for Deep Face Recognition*, January, DOI: 10.1109/TPAMI.2021.3087709.
- Ennaama, F., Benhida, K. and Essalki, S. (2022) ‘Proposed approach to improve facial recognition techniques for occluded faces by Covid-19 mask protection’, *IFAC-PapersOnLine*, Vol. 55, No. 12, pp.456–461, DOI: 10.1016/j.ifacol.2022.07.354.
- Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z. and Lu, H. (2019) ‘Dual attention network for scene segmentation’, *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June, DOI: 10.1109/CVPR.2019.00326.
- Hariri, W. (2022) ‘Efficient masked face recognition method during the COVID-19 pandemic’, *Signal Image Video Process*, April, Vol. 16, No. 3, pp.605–612, DOI: 10.1007/s11760-021-02050-w.

- He, K., Zhang, X., Ren, S. and Sun, J. (2016) 'Deep residual learning for image recognition', in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, June, pp.770–778, DOI: 10.1109/CVPR.2016.90.
- Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R.R. (2012) *Improving Neural Networks by Preventing Co-Adaptation of Feature Detectors*, DOI: 10.48550.
- Hu, J., Shen, L., Albanie, S., Sun, G. and Wu, E. (2017) *Squeeze-and-Excitation Networks*, DOI: 10.48550/arXiv.1709.01507.
- Huang, G., Liu, Z., Van Der Maaten, L. and Weinberger, K.Q. (2017) 'Densely connected convolutional networks', in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, July, pp.2261–2269, DOI: 10.1109/CVPR.2017.243.
- Jian, M., Lam, K-M., Dong, J. and Shen, L. (2015) 'Visual-patch-attention-aware saliency detection', *IEEE Trans. Cybern.*, August, Vol. 45, No. 8, pp. 1575–1586, DOI: 10.1109/TCYB.2014.2356200.
- Kaur, G. et al. (2022) 'Face mask recognition system using CNN model', *Neuroscience Informatics*, September, Vol. 2, No. 3, p.100035, DOI: 10.1016/j.neuri.2021.100035.
- Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2017) 'ImageNet classification with deep convolutional neural networks', *Commun. ACM*, May, Vol. 60, No. 6, pp.84–90, DOI: 10.1145/3065386.
- Kuen, J., Wang, Z. and Wang, G. (2016) *Recurrent Attentional Networks for Saliency Detection*, DOI: 10.48550/arXiv.1604.03227.
- Li, Y., Guo, K., Lu, Y. and Liu, L. (2021) 'Cropping and attention based approach for masked face recognition', *Applied Intelligence*, May, Vol. 51, No. 5, pp.3012–3025, DOI: 10.1007/s10489-020-02100-9.
- Lin, H., Tse, R., Tang, S-K., Chen, Y., Ke, W. and Pau, G. (2021) 'Near-realtime face mask wearing recognition based on deep learning', in *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, IEEE, January, pp.1–7, DOI: 10.1109/CCNC49032.2021.9369493.
- Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B. and Song, L. (2017) *SphereFace: Deep Hypersphere Embedding for Face Recognition*, DOI: 10.48550/arXiv.1704.08063.
- Minaee, S., Minaei, M. and Abdolrashidi, A. (2021) 'Deep-emotion: facial expression recognition using attentional convolutional network', *Sensors*, April, Vol. 21, No. 9, p.3046, DOI: 10.3390/s21093046.
- Simonyan, K. and Zisserman, A. (2014) *Very Deep Convolutional Networks for Large-Scale Image Recognition*, DOI: 10.48550/arXiv.1409.1556.
- Szegedy, C. et al. (2015) 'Going deeper with convolutions', in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, June, pp.1–9, DOI: 10.1109/CVPR.2015.7298594.
- Ullah, N., Javed, A., Ali Ghazanfar, M., Alsufyani, A. and Bourouis, S. (2022) 'A novel DeepMaskNet model for face mask detection and masked facial recognition', *Journal of King Saud University – Computer and Information Sciences*, November, Vol. 34, No. 10, pp.9905–9914, DOI: 10.1016/j.jksuci.2021.12.017.
- Wang, B., Zhao, Y. and Chen, C.L.P. (2021) 'Hybrid transfer learning and broad learning system for wearing mask detection in the COVID-19 era', *IEEE Trans. Instrum. Meas.*, Vol. 70, pp.1–12, DOI: 10.1109/TIM.2021.3069844.
- Wang, F. et al. (2017a) *Residual Attention Network for Image Classification*, DOI: 10.48550/arXiv.1704.06904.
- Wang, X., Girshick, R., Gupta, A. and He, K. (2017b) *Non-Local Neural Networks*, DOI: 10.48550/arXiv.1711.07971.
- Wang, Z., Sun, W., Zhu, Q. and Shi, P. (2022) 'Face mask-wearing detection model based on loss function and attention mechanism', *Comput. Intell. Neurosci.*, July, Vol. 2022, pp.1–10, DOI: 10.1155/2022/2452291.

- Woo, S., Park, J., Lee, J.-Y. and Kweon, I.S. (2018) *CBAM: Convolutional Block Attention Module*, DOI: 10.48550/arXiv.1807.06521.
- Woo, S., Park, J., Lee, J.-Y. and Kweon, I.S. (2018) *CBAM: Convolutional Block Attention Module*, pp.3–19, DOI: 10.1007/978-3-030-01234-2_1.
- Zeiler, M.D. and Fergus, R. (2014) *Visualizing and Understanding Convolutional Networks*, pp.818–833, DOI: 10.1007/978-3-319-10590-1_53.

Appendix

<i>Abbreviation</i>	<i>Definition</i>	<i>Abbreviation</i>	<i>Definition</i>
COVID-19	Coronavirus disease	MDMFR	Masked detection mask face recognition
MTCNN	Multi-task cascaded convolutional neural network	RELU	Rectified linear activation function
CBAM	Convolutional block attention module	BPMS	Business process management systems
MFNA	Modified face net architecture	LRELU	Leaky rectified linear activation function
CIOU	Complete intersection over union	TS	Total samples
CNN	Convolutional neural networks	TN	True negative
VGG-16	Visual Geometry Group-16	FN	False negative
MLP	Multilayer perceptron	TP	True positive
RCNN	Region-based convolutional neural network	FP	False positive
BLS	Broad learning system	FCN	Fully convolutional networks
MFCOSFACE	Margin cosine loss	FPS	Frames per second
YOLO	You only look once	SVM	Support vector machine
MGCN	Multi-graph convolutional networks		