Ingénierie des Systèmes d'Information
Vol. 30, No. 2, February, 2025, pp. 449-456
Journal homepage: http://iieta.org/journals/isi

Face Mask Recognition in a Street Camera Video Stream by Using Double CNN with Adaptive Particle Swarm Optimization

Rahul Bhandari[1], Anuj Kumar Jain[1], Gaganpreet Kaur[2], Meena Malik[1], Durgesh Nandan[3*], Om Prakash Suthar[4], Ram Kishun Lodhi[5]

[1] Department of CSE, Chandigarh University, Mohali 140413, India
[2] Department of CSE, Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab 140401, India
[3] School of CS & AI, SR University, Warangal 506371, India
[4] Department of Computer Engineering, Marwadi University, Rajkot 360003, India
[5] Department of Applied Science, Symbiosis Institute of Technology, Pune Campus, Symbiosis International (Deemed University), Pune 412115, India

Corresponding Author Email: durgeshnandano51@gmail.com

https://doi.org/10.18280/isi.300216

Received: 26 November 2024
Revised: 23 January 2025
Accepted: 17 February 2025
Available online: 27 February 2025

Keywords:
mask detection system, double CNN, deep learning, adaptive particle swarm optimization, computer vision

ABSTRACT

This research paper's primary goal is to create an application that makes it easier to identify persons in photos and determine whether or not they are wearing masks. When trying to identify faces in a picture or a video, this is crucial. The framework is composed of two networked CNN systems that use APSO and deep classical machine learning to recognize face masks. It is possible to identify those wearing masks with this strategy. OpenCV runs continuously via a webcam to detect faces in real-time. If there are two or more faces of the object, each face is enclosed by rectangles, and the rectangles indicate the coordinates of the face. Therefore, it opens the way for the immediate identification of correct and incorrect use of face masks in the stream. Based on this, we came up with two approaches to the problem and estimated the pertinence and possibly the time that may be necessary to accomplish the strategy. The first approach uses the face detector extracted from the synthetic datasets corresponding to the latest size and the masked image classifier. In the second approach, several of the restricted real photographs associated with them priori tags employed in identifying and detecting the facial positions are done with the help of an accurate object detection network.

1. INTRODUCTION

Algorithms of the mask detection platform are based on artificial neural networks that analyze whether or not the users wear masks. This application can interactively connect with the existing and new IP mask detection cameras to recognize those who are not wearing masks [1]. Users can invite faces and phone numbers into the app, and it will notify them when someone is not wearing a mask. There is also an option for when the camera does not identify a face or when the user sends a notification to the administrator. This system integrates IP and CCTV cameras, coupled with computer vision, and identifies people without masks. Whenever the mask sensor app detects a specific user who is not wearing a mask, a relevant AI alert with the person's photograph is immediately provided with a reminder of the necessity to wear a mask [2].

Given real-world deployment's real-time nature and constraints, we address this problem from two perspectives: productivity and effectiveness [3]. Performance can be understood as the achievement of planned outcomes or targeted aims. On the other hand, efficiency refers to utilizing inputs in a production process. The first pipeline is more focused on accuracy with a pre-trained face detector that extracts faces from the frames by passing them through an image classification program. An extensive synthetic dataset of 180,000 photos was used to train this classifier. The images were divided into three categories: non-use of the face mask, inappropriate use of the face mask, and proper use of the face mask [4, 5]. Conventional methods and contemporary Computer Vision architectures like Random Forest and Haar Cascade, DenseNet, and ResNet have been attempted.

The second pipeline is called YOLO, a real-time object detection architecture that aims to classify and detect things in real-time based on the current situation. This one trains the model for face-mask identification using the real face frames and the mask-wearing category. The dataset volume is relatively modest at 14,233 since labeling the bounding box for the dataset is laborious because there is no way to synthesize images.

To detect faces, this study presents the Double CNN architecture [6], which uses a two-step process: first, a lightweight image classifier determines whether people are wearing masks; second, an image classification of distinguishable faces based on mask wear is performed. A frame and the categorization result are shown around the face once the classification is complete. The algorithm works effectively with camera and video equipment such as closed-circuit television cameras for security infringement surveillance, encouraging mask wear, and guaranteeing worker safety. It is appropriate for identifying face masks in films and photos [7].

The following is a breakdown of the various sections that make up this study article: In the first section of this study, face mask detection is introduced. The author presents the findings of the study conducted by other researchers in section 2 of the paper, which is the literature review. Section 3 contains the proposed architecture as well as the former that was used for this investigation. In light of this, Section 4 presents the experimental analysis. Finally, Section 5 presents the research work's conclusion.

2. RELATED WORK

Initially, several researchers attempted to match faces using the grayscale and edge values of the database's face images [8]. This technique was predicated on a recognition model created using the experts' understanding of facial anatomy. The Viola-Jones Detector [9] was developed in response to the need for this model to be modified due to the demands of improved face identification in the last ten years. Despite significant challenges with the lighting and position of the face, K featured this detector since it enhanced real-time face detection.

At first, researchers tried to recognize faces by classifying the edges and grayscale values of facial images [8]. This approach used a recognition model built with an existing knowledge of building block facial structures. Still, with the advances in face recognition technology over this past decade, an improvement in such model generation was necessary and would be made possible by the introduction of Viola-Jones Detector [9]. Although large gains were already achieved in real-time face detection, challenges like varying light conditions and facial orientation still need to be addressed. The paper proposed a new face detection method based on deep learning to achieve better results across various facial image situations. Using sample data designated as "training data," deep learning is a sort of machine learning that generates predictions or judgments without the need for explicit coding instructions [10, 11]. Deep learning is a type of machine learning that provides predictions or decisions based on sample data called training data and does not require explicit coding instructions [10, 11].

Deep learning algorithms seek to discover rich internal features of data distributions by performing distributed representations learned through a cascade using a few-layered architecture, where the higher-level features are defined in lower levels. While built on machine learning and deep learning concepts, some limitations remain. To solve these problems, some libraries have been developed, such as OpenCV, TensorFlow, and PyTorch [12].

OpenCV is an efficient library for developing real-time computer vision applications with an emphasis on aspects like image capture, image processing, and video capture and analysis, some of which include face or object recognition. TensorFlow is an open-source software for data flow and differentiable computing that has found wide application in a range of tasks, including neural networks. Pytorch, another open-source artificial intelligence software developed on the Torch software, is used mainly in computer vision and language processing [13].

Another method presented by Ejaz et al [14] in 2019 is related to facial recognition in masked and unmasked conditions, as well as the implementation of Principal Component Analysis (PCA) [14]. However, the detection percentage decreases and reaches a value below 70% if the recognized face is masked [15]. In another study, Qin and Li [7] proposed a method for identifying masked and unmasked faces, categorizing them into three groups: recognized three categories of facial images; those with proper face masks, improper face masks, and no face masks [7, 16]. Their algorithm takes pictures, detects and isolates faces, and finally uses SRC_Net [17] to categorize people according to COVID-19 mask use. The developed framework emphasizes the interconnection of devices or actions to monitor patients and alert authorities [18]. A well-informed team using interconnected devices is formed to identify groups effectively. Patients are interviewed, and their previous movements are tracked. The proposed model leverages technology to track infected individuals, with robots and robotic technologies utilized to provide services to infected individuals [19, 20]. Additionally, Rupani et al. [21] proposed a robust image processing technique using IoT and FPGA to perform various filter operations. At the same time, Balamurugan et al. [22] introduced a Wrap-CNN with a voting scheme to address the recognition problem of multi-view objects and achieve improved recognition accuracy. In the study [23], a wearable (IoT)-based integrated healthcare system for initial detection with higher performance was created. The suggested integrated healthcare system makes use of wearable technology and deep learning methods to improve patient monitoring, diagnosis, and treatment. To detect those individuals who are afflicted at an early stage before the illness advances, the system makes use of the WIoT based on Artificial Intelligence technology. The proposed wearable health care system uses RNN-LSTM to monitor the data collected. It achieves an accuracy of 93% at epoch is 60 and batch size for RNN-LSTM is 100. While ASD detection with functional magnetic resonance imaging (fMRI) using parallel DCNN (PDCNN) was suggested in the study [24]. The PDCNN helps to acquire distinctive features with different filter kernels at parallel layers to describe the distinct local connectivity features of fMRI images and improve ASD detection accuracy [24]. Chaudhary et al. [25] proposed a novel way for evaluating MRI brain images that employs convolutional neural networks. By scanning through whole images using a convolutional neural network, attributes of equivalent spatial quality were discovered. It employed CNN model, a number of activation functions, including sigmoid, tanh, and rectified linear units (ReLU). Further the CNN model was divided into multiple tiers. The study [26] devised an innovative and integrated methodology for the accurate detection of COVID-19 using CT scans.

3. PROPOSED METHODOLOGY

The proposed framework for discerning and isolating masked and unmasked faces is shown in Figure 1. The framework comprises two main processes: face detection and face mask classification. The face detection involves identifying and detecting multiple faces from the dataset. Detected faces are batched and proceed to the face mask classification process. The face mask classification process employs a double CNN-based classifier to correctly classify all faces as masked or unmasked.

In the face detection process, the output of face detection consists of detected and grouped faces. This stage requires capturing each person's entire head to enable accurate classification of mask usage. The first step is to increase the height and width of the bounding boxes so as to capture the Region of Interest (ROI) with reduced interference from nearby faces. This will help in determining the effect on the detector's accuracy. The experiment tries with different expansion percentage of 10%, 25%, 50% for verifying the accuracy of detecting face. This experiment uses 25% ROI to detect the face. Next, the extended bounding boxes are cropped from the image to isolate the ROI for each detected face. These extracted faces are then resized and standardized.

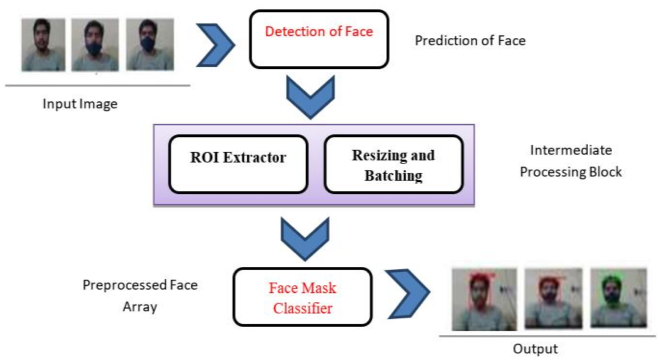


Figure 1. System architecture of the proposed method

The flowchart for detecting a face is shown in Figure 2. It consists of three main processes: face acquisition, grayscale conversion, and algorithm training. The goal of face acquisition is to collect a lot of face samples to build a strong dataset. In this process, a pre-trained model called "haarcascade_frontalface_default.xml", is used to detect faces. After the acquisition of the face, the grayscale conversion process of grayscale goes on. It aims to make an algorithm faster and more accurate. Finally, the algorithm training process is executed to train the algorithm to recognize faces in images using cascade functions, classifiers, and the Haar cascade. The four phases of the algorithm training are shown in Figure 3.

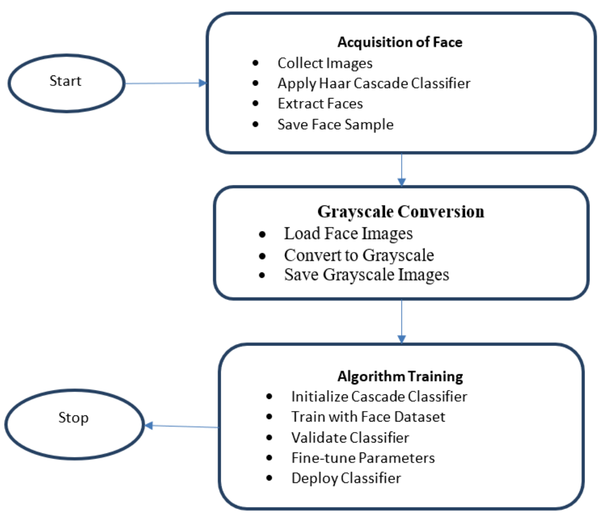


Figure 2. Flowchart for detecting a face

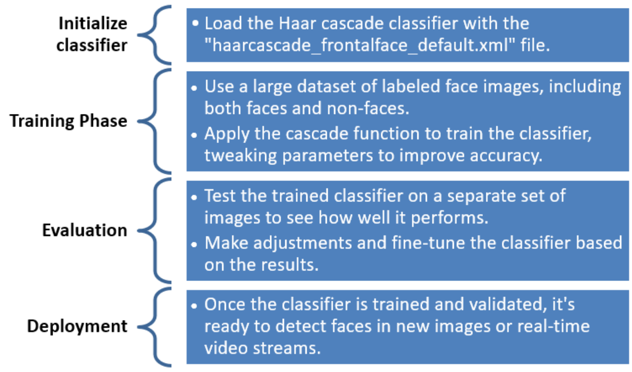


Figure 3. Phases of algorithm training

Python and OpenCV are used in the suggested study to develop face identification. The programming language used is Python, which is favored for its ease of use and adaptability. OpenCV, an open-source computer vision toolkit, provides the features and techniques to find and identify faces in photos and video streams. While OpenCV is feature-rich and easy to use, Python allows for high-accuracy face identification in a variety of related applications.

This method can be further extended for face recognition by providing a set of input images, allowing the system to verify the identity of the detected individual. The procedure advances to the following steps as soon as a face is successfully identified. The recognized faces are gathered together in this instance, and the ROI is represented by bounding boxes made all the way around the head. A face detection example is shown in Figure 4.

(a) Original image (b) Detected image

Figure 4. An example of the face detection image

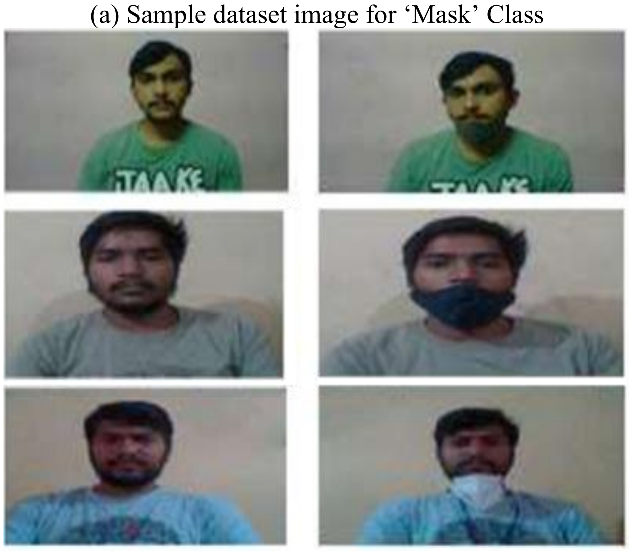
Figure 5. Sample mask images include non-masked, improper, and masked images

The dataset obtained in step 1-images of persons with appropriate, unsuitable, and no masks on their faces-is shown in Figure 5. Step 2 involves preparing the dataset to extract the region of interest from the previously mentioned photos. In addition, this step aims to include the entire head of each detected face.

Figure 6 illustrates these ROIs by presenting the heads of the only recently identified faces in the photos. These ROIs are feature-extracted and classified as masked or non-masked faces using a face mask classifier.

Figure 6. Some examples of the test images converted their respective ROIs

(a) Sample dataset image for ‘Mask’ Class



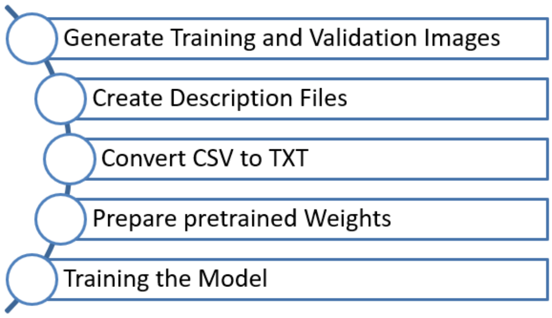
(b) Sample dataset images for ‘No-Mask’ class

Figure 7. Sample dataset images

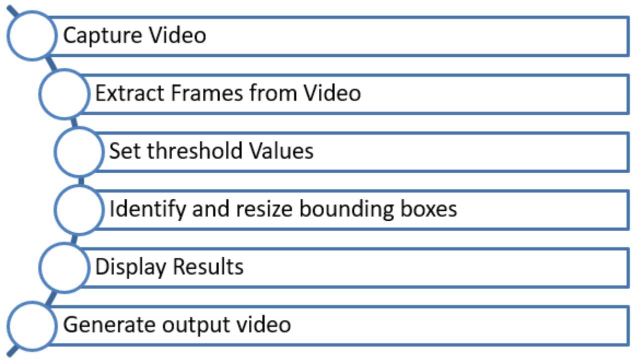
The last step of the proposed model includes a face mask classifier, which categorizes the ROI images given as input. This classifier is implemented using different CNN-based architectures such as MobileNetV2, DenseNet121, and NASNet. In this process, the internal processing block decides whether the images are masked or unmasked.

The suggested method is appropriately illustrated in Figure 7, which also shows how various classifiers may be applied to accurately categorize photos based on the mask wear regime: with, without, or with no compliance at all. This demonstrates how well the classifiers perceive different combinations of mask wear or non-wear. Figures 8(a) and 8(b) depict further technical details, explain how the model is trained using a double CNN. Figure 8(a) shows the outcome of the training phase, and Figure 8(b) shows the testing phase, where the OpenCV library is used to run the model in real-time via a webcam or video stream.

With reference to the aspect of identification of self-identity as part of the achievement of face masks, the process is easy and exact in the proposed system based on the multi-stage identification system which has attempted to use some of the most effective machine learning techniques and frameworks. Here is a detailed description of the stage of the developed system and its capability.



(a) Steps of training of double CNN



(b) Steps of testing on webcam or video using OpenCV

Figure 8. Training and testing the model

Stage 1: Face Detection

Hence while identifying the human faces in the initial stage by the help of OpenCV library, turn in to the next process in the system. Open Computer Vision is one of the popular open source computer vision libraries from where one can find number of techniques and algorithms to process the image and detect objects on the image. Similarly, using OpenCV ensures the assurance that the possibility of recognizing faces in given images is easy regardless of the lighting of the images and the background or movements of the faces in the images.

Stage 2: Mask Classification

Once faces are detected, the system proceeds to the second stage, which involves mask classification using a combination of state-of-the-art neural network architectures. The system goes to the second step of the process which is the mask classification after the faces have been detected, and this may utilize a number of the distinguished state-of-the-art neural network architectures.

1. NASNetMobile:

- Majority of these newly invented CNNs are still heavy and complex, hence time-consuming to execute on devices, say mobile phones; this led to the development of a light CNN called NASNetMobile or Neural Architecture Search Network Mobile.

- It is accuracy and the processing time, which also means that it will easily be developed to work on less powerful hardware such as mobile devices or in the cloud.

2. DenseNet121:

- DenseNet121 is widely popular because it has a dense connection, which means a fully connected feed-forward network in which each layer is connected to all the other layers.

- Moreover, this kind of architecture does not have the vanishing gradient issue, and the increase in feature reuse is good for the model.

DCNN with APSO Classifier:

- The term 'Double CNN' when used is often used to describe a situation where two Convolutional Neural Networks (CNNs) have been integrated in a way that will improve on the feature extraction, analysis or comparison. In my opinion, this is helpful for particular cases like image pairing, when we work with two sets at the same time, or with multi-modal data to combine their effective features into a single model, or modifying feature learning by utilizing one architecture within the task solved via the other architecture.

- APSO can optimize the feature set derived from the DCNN for the purpose of subsequent tasks. The DCNN extracts high level features from a given input data. Here, APSO chooses a subset of these features using a mechanism in which every particle represents a potential feature subset. To assess each particle's fitness, a lightweight feature selection algorithm is trained, cross-validated through a lightweight model. APSO fine tunes the selection process to try and achieve the best performance with the least computational complexity.

Performance and Capabilities

The framework developed here is the NASNetMobile, DenseNet121, and DCNN with APSO classifier, which finalizes the exceptional performance. It can differentiate whether a face in an image is wearing a mask or not in different situations.

To train the Double CNN, follow the steps outlined in Figure 8(a). First, organize your dataset by creating two folders: `test_images` and `train_images`. Place all test images in the `test_images` folder and all training images in the `train_images` folder. Next, create a `train.csv` file that describes your dataset. This file should include details about each image, such as the file name, label, and any other relevant metadata. Also, create a plain text file listing all the class names the model will recognize, with one class name per line. Then, convert the `train.csv` file to a .txt file in the specific format required for your training process. This step involves creating a new data frame with the necessary columns and values and then saving it as a .txt file. Ensure the pre-trained model weights are in the '.h5' format, as required by Keras. If the weights are in a different format, they will need to be converted. Now, unfreeze all the layers of the model to make them trainable. Begin training with a higher learning rate and gradually reduce it to fine-tune the model for better performance. Monitor the loss for both the validation and training datasets, and continue training until the loss is low and consistent. This indicates that the model is well-trained without overfitting.

To test the model using a webcam or video, as shown in Figure 8(b), follow these steps:

- Capture the video using any camera or select an existing video for testing.
- Extract frames from the video and input them into the model.
- Set threshold values and identify bounding boxes along with their scores.
- Resize the bounding boxes to their original size.
- Replace the bounding box outlines with shaded rectangles.
- Store all relevant information in a variable.
- After processing each frame, generate an output video containing the desired results.

4. EXPERIMENTAL ANALYSIS

Multiple databases serve as data sources for this analysis. Our dataset has been trained using three kinds of images: masked, unmasked, and images with improper masks. The face mask detection dataset was sourced from Kaggle [20]. Notably, images with improper masks are categorized as non-masked images. Table 1 presents the complete dataset we will be utilizing. It includes images of faces with proper mask usage and those with improper or no mask usage. Therefore, there are 9877 images and 2 classes in this dataset.

The results based on the proposed model are as follows: the proper classification of the dataset between mask and no-mask is shown in Figure 9. The system separately and accurately divides the images with properly worn masks into the mask class, while those without or with improperly worn masks fall into the no-mask class.

Table 1. Face mask classifier dataset

Class Name	Description	No. of Images
Mask	Proper mask used	5538
No-Mask	No mask or improper mask used	4339

Figure 9. Results of the mask and no-mask images using the proposed method

This makes the proposed system general and stable for use, making it extremely practical and implementable in different real-life settings such as commercial spaces, workplaces, and any other setting that requires everyone to wear a mask for health considerations.

This training model presents an integration of PSO with a focused adaptive mechanism called the 'strategically adaptive PSO' for the process of adaptive learning. It should be noted that depending on the distribution of particles, the group is arranged in a dynamic manner during the navigation process. There are two learning strategies used to actively adjust the search direction of two types of particles within each of the subgroups. The search can end when the global best solution value is reached or when it reaches a certain stop criterion. Subsequently, the particles are grouped into adaptive populations. In this stage, local particle density and distance to particles with higher local density are calculated and executed. Specific to the values obtained using particles' coordinates, hyperparameters of the model are set, and validation data becomes part of the model for prediction. Last but not least, the loss function of the dataset is checked to act as the fitness function for the particles.

This makes the proposed system general and stable for use, making it extremely practical and implementable in different real-life settings such as commercial spaces, workplaces, and any other setting that requires everyone to wear a mask for health considerations.

This training model presents an integration of PSO with a focused adaptive mechanism called the 'strategically adaptive PSO' for the process of adaptive learning. It should be noted that depending on the distribution of particles, the group is arranged in a dynamic manner during the navigation process. There are two learning strategies used to actively adjust the search direction of two types of particles within each of the subgroups. The search can end when the global best solution value is reached or when it reaches a certain stop criterion. Subsequently, the particles are grouped into adaptive populations. In this stage, local particle density and distance to particles with higher local density are calculated and executed. Specific to the values obtained using particles' coordinates, hyperparameters of the model are set, and validation data becomes part of the model for prediction. Last but not least, the loss function of the dataset is checked to act as the fitness function for the particles.

This dataset consists of two main classes for a face mask classifier: "Mask," which includes images of individuals wearing masks correctly, and "No-Mask," featuring images where individuals either do not wear a mask or wear it improperly. In total, there are 9,877 images, with a greater number representing proper mask usage.

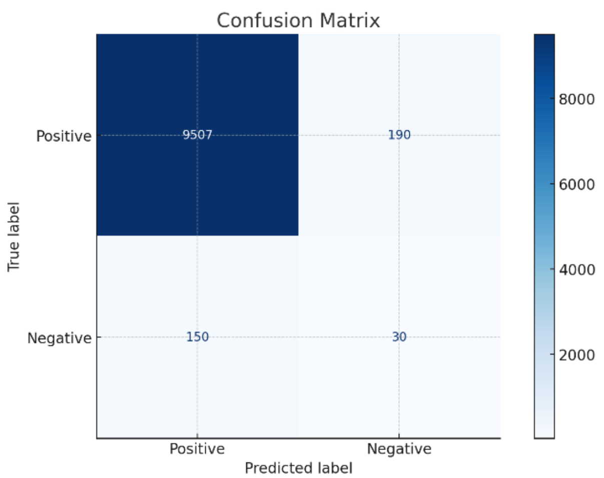


Figure 10. Confusion matrix of training the dataset

The classifier's performance in terms of precision and recall is high especially in positive cases shown in Figure 10. However, the Specificity which measures the model's ability to correctly identify negative cases is low as well. Such limitation might be attributed to the fact that the used dataset has fewer negative samples compared to positive ones. Despite the high Accuracy resulting from the large number of positive deviations, the model struggles with the cases classified as negative.

Table 2 summarizes the performance of three proposed face mask classifiers across different metrics. It includes training accuracy and loss values, validation accuracy and loss, as well as test accuracy. The results indicate that all classifiers achieved high training and validation accuracies, with NASA_Net_Mobile showing the highest training accuracy, while DCNN with APSO achieved the best test accuracy. The loss values are relatively low, suggesting effective learning across all models.

Table 2. Training result set for all three proposed face mask classifiers

Classifier Name	**Training Accuracy and Loss Value**		**Validation Accuracy and Loss Value**		**Test Accuracy**
	Accuracy (%)	Loss	Accuracy (%)	Loss	
NASA_Net_Mobile	98.91	0.00132	98.44	0.0182	98.26
Dense_Net_121	98.51	0.00157	98.72	0.0313	98.51
CNN [27]	98.00	0.05000	98.12	0.0560	97.39
DCNN with APSO	99.41	0.01520	99.36	0.0198	99.15

Table 3. Performance metrics result set for all three proposed face mask classifiers

Model	**Precision**	**Recall**	**F1-Score**
NASA_Net_Mobile	99.30	99.00	98.17
Dense_Net_121	98.72	98.16	98.42
CNN [27]	97.34	97.41	97.31
DCNN with APSO	98.15	98.05	98.54

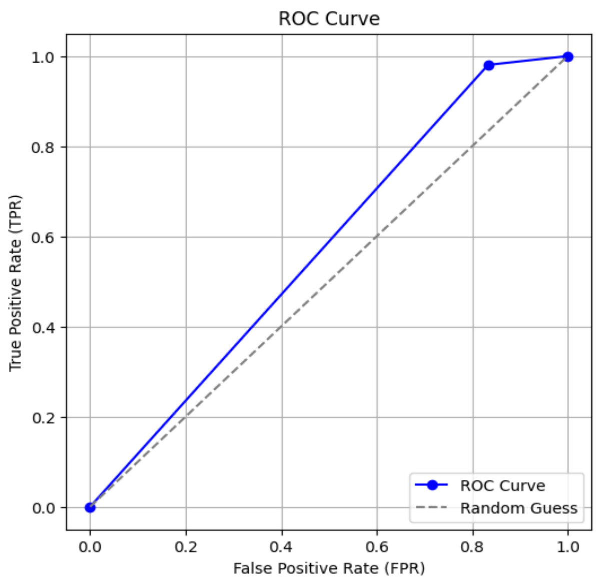


Figure 11. ROC curve of training dataset

Table 3 presents the performance metrics for four proposed face mask classifiers, focusing on precision, recall, and F1-score. NASA_Net_Mobile exhibits the highest precision at 99.30%, indicating its strong ability to minimize false positives. Dense_Net_121 follows closely with solid precision and recall values, resulting in a robust F1-score.CNN has the lowest metrics across the board, but still demonstrates reasonable performance. DCNN with APSO achieves a noteworthy F1-score of 99.16%, highlighting a balance between precision and recall. Overall, the metrics reflect that all classifiers perform well, with varying strengths in precision, recall, and overall effectiveness.

The curve in the Figure 11 is ROC (Receiver Operating Characteristic) curve which is used to measure the performance of a binary classifier. The low distance of the curve to the diagonal line indicates a low method of discrimination. This shape and this disposition with regard to the diagonal lead to the conclusion that the classification is slightly better than random.

5. CONCLUSIONS

This research proposes a mask detection technique involving several steps. Initially, a pre-trained human dataset is utilized, and the OpenCV library performs face detection to identify individuals wearing appropriate masks, inappropriate masks, and those without masks, thereby creating a dataset. Subsequently, various classifiers, including NASNet Mobile, DenseNet 121, and DCNN with APSO, are employed to differentiate between suitable and unsuitable mask usage based on insightful NASNet Mobile-based models. This approach ensures that the model can effectively avoid local optima when optimizing the hyperparameters of DCNN with APSO. Furthermore, a warning system is integrated, leveraging deep learning and machine operation technologies to detect and provide output regarding mask-wearing status solely based on real-time face detection using Double Convolution Neural Networks (DCNN) with APSO. The model, trained with DCNN with APSO, TensorFlow, and Keras algorithms, achieves a performance and accuracy rate of 98.15%, making it highly effective. Moreover, the device is designed using cost-effective materials, enabling widespread accessibility. Additionally, it features an alarm system that displays a red light when mask-wearing is not detected and emits a green light when masks are properly worn, further enhancing its usability and functionality with different configuration.

REFERENCES

[1] Fang, Y., Nie, Y., Penny, M. (2020). Transmission dynamics of the COVID-19 outbreak and effectiveness of government interventions: A data-driven analysis. Journal of Medical Virology, 92(6): 645-659. https://doi.org/10.1002/jmv.25750

[2] Liu, Y., Gayle, A.A., Wilder-Smith, A., Rocklöv, J. (2020). The reproductive number of COVID-19 is higher compared to SARS coronavirus. Journal of Travel Medicine, 27(2): 1-4. https://doi.org/10.1093/jtm/taaa021

[3] Howard, J., Huang, A., Li, Z., Tufekci, Z., Zdimal, V., van der Westhuizen, H.M., Rimoin, A.W. (2020). Face masks against COVID-19: An evidence review. Epidemiology and Infectious Diseases, https://doi.org/10.20944/preprints202004.0203.v1

[4] Feng, S., Shen, C., Xia, N., Song, W., Fan, M., Cowling, B.J. (2020). Rational use of face masks in the COVID-19 pandemic. The Lancet Respiratory Medicine, 8(5): 434-436. https://doi.org/10.1016/S2213-2600(20)30134-X

[5] Vinitha, V., Velantina, V. (2020). COVID-19 facemask detection with deep learning and computer vision. International Journal of Engineering Research & Technology, 7(8): 3127-3132.

[6] Chavda, A., Dsouza, J., Badgujar, S., Damani, A. (2021). Multi-stage CNN architecture for face mask detection. In 2021 6th International Conference for Convergence in Technology (I2CT), Maharashtra, India, pp. 1-8. https://doi.org/10.1109/I2CT51068.2021.9418207

[7] Qin, B., Li, D. (2020). Identifying facemask-wearing condition using image super-resolution with classification network to prevent COVID-19. Sensors, 20(18): 5236. https://doi.org/10.3390/s20185236

[8] Verma, S., Dhanak, M., Frankenfield, J. (2020). Visualizing the effectiveness of face masks in obstructing respiratory jets. Physics of Fluids, 32(6): 061708. https://doi.org/10.1063/5.0016018

[9] Wang, Z., Huang, B., Wang, G., Yi, P., Jiang, K. (2023). Masked face recognition dataset and application. IEEE Transactions on Biometrics, Behavior, and Identity Science, 5(2): 298-304. https://doi.org/10.1109/TBIOM.2023.3242085

[10] Woods, A. (2020). Britain faces an anxiety crisis as people return to work. BDaily News, https://bdaily.co.uk/articles/2020/06/22/britain-faces-an-anxiety-crisis-as-people-return-to-work.

[11] Kumar, A., Kaur, A., Kumar, M. (2019). Face detection techniques: A review. Artificial Intelligence Review, 52: 927-948. https://doi.org/10.1007/s10462-018-9650-2

[12] Meenpal, T., Balakrishnan, A., Verma, A. (2019). Facial mask detection using semantic segmentation. In 2019 4th International Conference on Computing, Communications and Security (ICCCS), Rome, Italy, pp. 1-5. https://doi.org/10.1109/CCCS.2019.8888092

[13] Rani, S., Acharjee, P.B., Kaswan, S., Athavale, V.A., Udhayamoorthi, M., Pant, K. (2022). A face spoof detection in artificial neural networks using concepts of machine learning. In 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 2022, pp. 1604-1607. https://doi.org/10.1109/ICACITE53722.2022.9823587

[14] Ejaz, M.S., Islam, M.R., Sifatullah, M., Sarker, A. (2019). Implementation of principal component analysis on masked and non-masked face recognition. In 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), Dhaka, Bangladesh, pp. 1-5. https://doi.org/10.1109/ICASERT.2019.8934543

[15] Zoph, B., Vasudevan, V., Shlens, J., Le, Q. (2018). Learning transferable architectures for scalable image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4510-4520

[16] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Greater Noida, India, pp. 4510-4520. https://doi.org/10.1109/CCAA.2018.8777675

[17] Kumar, S., Negi, A., Singh, J.N., Verma, H. (2018). A deep learning for brain tumor MRI images semantic

segmentation using FCN. In 2018 4th International Conference on Computing Communication and Automation (ICCCA), pp. 1-4.
[18] Fu, X., Qu, H. (2018). Research on semantic segmentation of high-resolution remote sensing image based on full convolutional neural network. In 2018 12th International Symposium on Antennas, Propagation and EM Theory (ISAPE), Hangzhou, China, pp. 1-4. https://doi.org/10.1109/ISAPE.2018.8634106
[19] Li, K., Ding, G., Wang, H. (2018). L-FCN: A lightweight fully convolutional network for biomedical semantic segmentation. In 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Madrid, Spain, pp. 2363-2367. https://doi.org/10.1109/BIBM.2018.8621265
[20] Karras, T. (2019). A style-based generator architecture for generative adversarial networks. arXiv preprint arXiv:1812.04948.
[21] Rupani, A., Jilani, S.A., Shibly, F.H.A., Choudhury, T. (2021). FPGA implementation of image processing filters using IoT. In Innovations in Cyber Physical Systems: Select Proceedings of ICICPS 2020, pp. 401-413. https://doi.org/10.1007/978-981-16-4149-7_36.
[22] Balamurugan, D., Aravinth, S.S., Reddy, P.C.S., Rupani, A., Manikandan, A. (2022). Multiview objects recognition using deep learning-based wrap-CNN with voting scheme. Neural Processing Letters, 54(3): 1495-1521. https://doi.org/10.1007/s11063-021-10679-4
[23] Kaur, G., Jain, A.K., Jain, N., Debnath, S.K., Bagchi, S., Kumar, A.S. (2024). Wearable intelligence system for health care: Synergizing smart devices and RNN-LSTM through IoT. In Computational Methods in Science and Technology, pp. 114-121.
[24] Dhamale, T.D., Bhandari, S.U., Harpale, V.K., Nandan, D. (2024). Autism spectrum disorder detection using parallel deep convolution neural network and generative adversarial networks. Traitement du Signal, 41(2): 643-652. https://doi.org/10.18280/ts.410208
[25] Chaudhary, S., Nehra, R., Kumar, K., Dahiya, S., Nandan, D., Kumar, S. (2024). Dementia Alzheimer's disease diagnosis using convolutional neural networks on two-dimensional MRI slices. Revue d' Intelligence Artificielle, 38(1): 81-91. https://doi.org/10.18280/ria.380109
[26] Dalal, S., Singh, J.P., Tiwari, A.K., Nandan, D. (2024). An automated computed tomography scan analysis framework for COVID-19 detection using machine learning. Traitement du Signal, 41(4): 1707-1726. https://doi.org/10.18280/ts.410406
[27] Kaur, G., Sinha, R., Tiwari, P.K., Yadav, S.K., Pandey, P., Raj, R., Rakhra, M. (2022). Face mask recognition system using CNN model. Neuroscience Informatics, 2(3): 100035. https://doi.org/10.1016/j.neuri.2021.100035