INTERNATIONAL JOURNAL OF
Telecommunications
AIR DEFENSE COLLEGE

Improving Smart Infrastructure Monitoring in Response to Frequent Pandemics

Citation: Mohamed, N.; Abd El Salam, A.; Mohamed, M.; ElNaghi , E.
Inter. Jour. of Telecommunications, ***IJT'*****2025**, *Vol. 05, Issue 01, pp. 1-14, 2025.*

Editor-in-Chief: Youssef Fayed.

Received: 10/01/2025.

Accepted: date 12/03/2025.
Published: date 13/03/2025.

Publisher's Note: The International Journal of Telecommunications, IJT, stays neutral regard-ing jurisdictional claims in published maps and institutional affiliations.

Mohamed, Nourhan Osama*[1], Ali, khaled Abd El Salam[2], Mohamed, Ahmed Magdy[3], and ElNaghi, Bassem ELhady[4]

* Correspondence: Electrical Engineering Department, Faculty of Engineering, Suez Canal University, Ismailia, Egypt. nourhanu4@gmail.com
[2] Electrical Engineering Department, Faculty of Engineering, Suez Canal University, Ismailia, Egypt. khaled.abdelsalam@eng.suez.edu.eg
[3] Electrical Engineering Department, Faculty of Engineering, Suez Canal University, Ismailia, Egypt. ahmed.m.1986@ieee.org.
[4] Electrical Engineering Department, Faculty of Engineering, Suez Canal University, Ismailia, Egypt. basem_elhady@eng.suez.edu.eg

Abstract: Airborne illnesses like chickenpox, influenza, and COVID-19 pose a major risk to public health since COVID-19 has killed about 7 million people. Wearing face masks has therefore become mandatory and significant. To prevent the spread of certain illnesses, particularly in healthcare institutions such as hospitals. This study introduces a scalable deep convolutional neural network (DCNN)--based face mask monitoring system that is better than manual surveillance, particularly in high-density settings. This study offers three methods: First, the pre-trained algorithms model, which included seven different algorithms and was optimized with hyperparameters to find optimal settings; the Darknet-53 algorithm performed the best among them, achieving an accuracy of 97.5%. The second was a customized DCNN model that achieved 96.4% accuracy in binary mask detection. The last suggested system is a hybrid model that improves the accuracy and stability of the model by using pre-trained algorithms as classifiers and a DCNN as a feature extractor. AlexNet and Darknet-53 were tested as classifiers in our study; Darknet-53's accuracy was 98%.

Keywords Airborne Diseases, Deep Convolution Neural Network, Pre-trained Algorithms.

1. Introduction

Recently, the spread of airborne diseases and epidemics has increased and caused many deaths all over the world. The airborne disease is caused by the spreading of viruses, bacteria, or fungi by a microorganism through the air. These can be transferred from one to another by sneezing, coughing, talking, spraying liquid, etc., and the particles that spread from the infected source can be suspended in the air for a while. As time went on, novel diseases arose that had not before occurred, such as COVID-19 in late 2019, which, according to WHO estimates, killed over 7 million people worldwide [1]. Before COVID-19 appeared, there were around a billion cases of seasonal influenza annually, causing 290,000 to 650,000 respiratory deaths annually [2]. Besides air pollution, which leads to many diseases and deaths, wearing a face mask is not welfare anymore; it comes from health care precautions, especially in places that are the source of epidemics, like hospitals, in addition to crowded areas like airports, schools, etc. and some governments make wearing a face mask in some places obligatory to reduce the risk of catching the virus [3]. We use a deep convolutional neural network to replace labor-intensive, scalable manual surveillance methods, particularly in high-density environments, to improve the Face Mask Smart Monitoring System. We use the most powerful tool for visual tasks like face mask detection and recognition, which is the Deep Convolutional Neural Network (DCNN) algorithm, which has a variety of algorithms and architectures used in this task. So, we try three different methods to get the most accurate and effective technique and algorithm. On a huge dataset to make it simulate the real world with different poses, contrast, and lightness, as well as different shapes of face masks, as there are a lot of face mask types, and the most popular are N95s, cloth face masks, surgical masks, and face shields. So, the publicly available dataset has effective rules to make smart system models more realistic. This paper presents an extensive analysis of this research issue and is divided into several sections. The paper's second section discusses related works that were considered with the same problem, as well as the approaches they used. The third section explains the materials and preprocessing steps applied and used in this work, along with the methodology implemented, such as the architecture and learning algorithm and the effective hyperparameters. Also, the system the process was tried on, the evaluation metrics, and the results alongside the model summary are all explained in the fourth section. The fifth section concludes and summarizes the work, and the sixth section talks about our future work, our passionate steps, and the goals set for the upcoming advancements. Finally, the list of references and sources we use in the citation.

2. Related Work

Face masks were one of the safety precautions that reduced the airborne diseases prevalence. So, wearing a face mask is recommended by the World Health Organization to control the infection rate and prevent rapid spread in the absence of effective antivirals and limited medical resources and in the presence of new airborne diseases that cause many deaths every few years, as well as the common influenza diseases do. In addition to the development of artificial intelligence and its role in the image processing field and classification and prediction tasks, it has become a notable topic since the 1990s [4]. Traditional machine learning (ML) approaches are used by many facial and object identification systems to improve network training and produce higher performance than earlier models. Face masks have become widely used, making it difficult for traditional facial recognition surveillance systems that rely on full-face visibility to reliably identify people. IoT devices, such as security cameras, can be used in conjunction with specific algorithms and technologies to enable masked-face recognition, which allows people to be identified and verified even when they are wearing masks [5]. This section summarizes the numerous studies that have been published on the topic of face mask detection. It focuses on recent studies that have employed DL for similar purposes. In AI-Based Monitoring of Different Risk Levels in the COVID-19 Context, Melo et al. [6] developed a CNN model to monitor COVID-19 risk levels by detecting the presence of face masks and taking body temperature. The model was pre-processed using synthetic data generation techniques and a large dataset to ensure that it had images in every possible circumstance. It was trained using ResNet-50-based key point detector and YOLOv5 for object detection. The system detects masks, spectacles, and caruncles with precisions of 96.65% and 78.7%, respectively, using thermal imaging. Its average accuracy at identifying masks in RGB images is 82.4%.

Two distinct datasets—AIZOO and Mola RGB CovSurv—are used for training and testing the model in the paper Face mask identification based on algorithm YOLOv5s [7]. The authors propose a smart method to detect face masks using YOLO v5, and the accuracy they achieved was 80.5%.

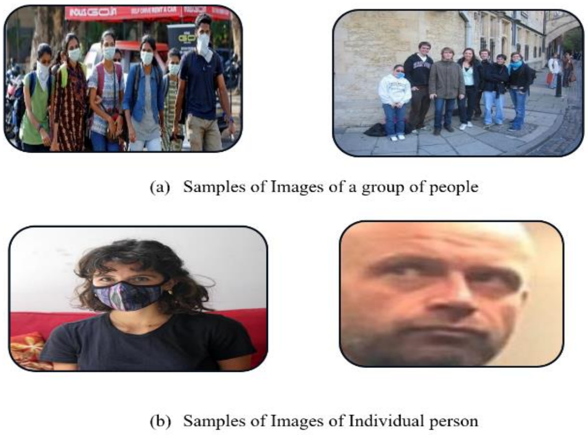
This paper, "Face Mask Detection using Deep Convolutional Neural Network and Multi-stage Image Processing [8], presents a comprehensive face mask detection system leveraging deep learning and image processing techniques. The proposed system uses a custom-designed CNN model along with a four-stage image preprocessing pipeline to enhance detection accuracy, and they achieved 97.25% accuracy.

Loey et al. [9] utilized YOLOv2 in conjunction with ResNet-50 to identify medical face masks; they divided the dataset into 90% for validation and 10% for testing, combining the Medical Masks Dataset (MMD) and Face Mask Dataset (FMD) for feature extraction and detection, respectively. The Adam optimizer was used to maximize the model's performance, and the result was an average precision (AP) of 81% for the detection of medical face masks. The study presents the RRFMDS (Rapid Real-Time Face Mask Detection System), an automated solution designed to monitor face mask compliance in real time using video feeds from CCTV cameras. Utilizing a Single-Shot MultiBox Detector (SSD) for 17 face detections and a fine-tuned MobileNetV2 model for mask classification, the system effectively identifies faces with or without a mask. Trained on a custom dataset of 14,535 images [10], the accuracy achieved is 97%.

The study introduces the RILFD (Real Image-based Labelled Face Mask Dataset) [11], which includes real images annotated with labels "with mask" and "without mask." The face mask detection system, ResNet Hybrid-Dilation-Convolution Face-Mask-Detector (RHF), is presented in this research, which makes use of hybrid dilation convolutional networks. To train and assess the model, the authors created the Light Masked Face Dataset (LMFD) and the Masked Face Dataset (MFD). By improving the convolutional kernel's perception, the hybrid dilation convolution network resolves problems with image discontinuity. The RHF model achieves better identification results with a mean Average Precision (mAP) of 93.45% while requiring a much shorter amount of training time than ResNet50. This study emphasizes the system's effectiveness in identifying masks in a variety of scenarios, highlighting its practicality.

3. Materials and Methods

In this study, we utilized the dataset provided by Mendeley Dataset [12]. This dataset is generated by Melo, César, et al. They generated a dataset for robust and dependable models to be generated, the amount of data utilized is another important consideration. It became necessary to create a technology that could create synthetic images. This tool was created to allow a large range of masks to be applied to publicly available datasets to have a variety of mask, The samples were taken from existing datasets such as Celeba [13], Coco [14], Helen [15], IMM [16], Wider [17], and Group Images [18], increasing sample diversity and enhancing training quality. Valuable due to its extensive range and diversity. It includes a large collection of images with various poses, image qualities, different contrast, lightness, etc., which are essential for robust and comprehensive analysis. The dataset encompasses both group images and images of people, providing a versatile array of data for training and testing machine learning models. Besides, the images are captured in different settings and conditions, improving the dataset's capability for use in real-world circumstances. Moreover, the dataset is freely accessible. By leveraging this dataset, we were able to achieve more accurate and generalized results in my project. It consists of 37469 images of people with masks and 19957 images of people without masks, so the total of the images was 57426 images after filtering them. Figure 1. displays samples of the dataset, showing a sample of images of a group of people, and a sample of images containing only individual people.



(a) Samples of Images of a group of people

(b) Samples of Images of Individual person

Figure 1. Samples from Mendeley Dataset (a) samples of images of a group of people (b) samples of images of a person [12].

3.1 Preprocessing

In this paper, first, we separate the dataset into images with groups of people and images with only one person in them. The basis for many face-related technologies, including face recognition and verification, is face detection. The Viola-Jones method, which was first presented in 2001, is used in this procedure to recognize faces in pictures. Grayscale photos are used by this technique, which is well known for its efficiency in real-time face detection. This algorithm uses Haar-like features that are used in a cascade of classifiers to improve detection speed, and it performs well in real-time face-detection images or videos [19]. Then it works with great efficiency and extracts faces from images each face alone and saves it to detect whether the face is with or without a mask. Then because images have varying dimensions, they must be resized to a certain dimension. Every image was scaled to 500×500 pixels. Third, Figure 2. indicates that images were cropped to an appropriate dimension.

Figure 2. Images after crop and resized

All prior preprocessing procedures are shown in Figure 3. In practice, the segmentation stage had an unfavourable effect on classification accuracy and system performance, therefore we skipped it to achieve better results.

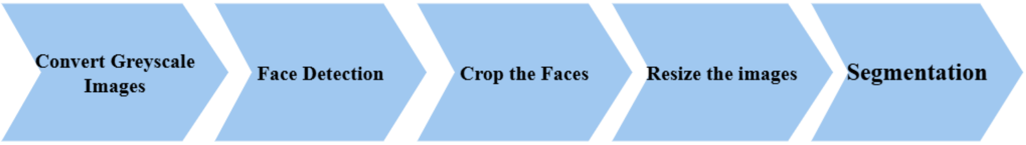


Figure 3. Preprocessing Steps of the Dataset Images

3.2 Model Architecture

The goal is to determine which of the three suggested model systems is the most accurate and effective. Therefore, we implemented three suggested systems: the first used a customized CNN architecture, the second used seven pre-trained algorithms, and the third is a hybrid model system that uses a customized CNN architecture to extract features and then pre-trained algorithms as classifiers that learn from the network and extracted features. And Figure 4. shows the general architecture of the suggested face mask detection system. The system is intended to assess and contrast three distinct model architectures: a hybrid model that combines the two methods, a customized Convolutional Neural Network (CNN), and several pre-trained algorithms.

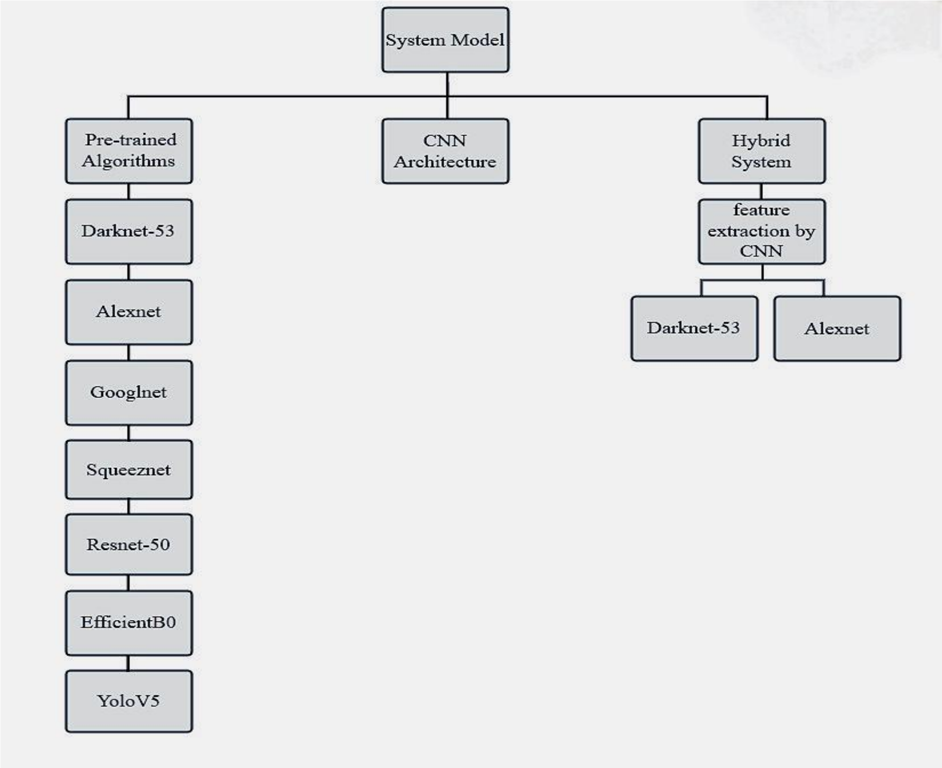


Figure 4. System Model

3.2.1 First proposed model system

Our first proposed model is a customized deep convolution neural network system model, CNN is an effective method for acquiring and analyzing the data by combining basic characteristics like edges and curves to create complex features map, Convolutional, nonlinear pooling, and fully connected layers are some of CNN's hidden layers [20]. First, we divided the dataset into 80% for training and 20% for testing the model. Images are resized to 224x224 pixels for compatibility with the input layer, we use five convolutional layers. To detect and analyze different degrees of features, from edges to extremely complex patterns, it begins with an image input layer and proceeds through five convolutional layers with increasing filter sizes (32, 64, 128, 256, 512, and 1024). Each convolutional layer is paired with batch normalization and ReLU activation which plays a crucial role in introducing non-linearity into the network It also helps in modulating the data passing through the network in a controlled manner. Besides helping in controlling gradients avoid issues like vanishing or exploding gradients. Each convolutional block is followed by a max-pooling layer, which lowers computational overhead and spatial dimensions, after that, there is a fully linked layer with 1024 units and dropout regularization to avoid overfitting. A SoftMax layer for classification comes after the final fully connected layer, which has two units for the two classes ('with mask 'and 'without mask '). Stochastic gradient descent with momentum (SGDM) is used in training, with a maximum of 20 epochs, a mini-batch size of 128, and a lower learning rate to improve convergence. The validation set is used to assess the network's performance after it has been trained on the augmented training set. Accuracy and sensitivity (true positive rate) are displayed to gauge the model's efficacy in picture classification tasks.

3.2.2 Second proposed model system

Convolutional Neural Networks (CNNs) have evolved significantly in pre-trained Algorithms inception, leading to various architectures tailored for different tasks and efficiency levels. So, we tried seven pre-trained algorithms (Alexnet, Squeeznet, YOLOV5, Googlnet, Resnet 50, Darknet 53, and EfficientB0) to get a more accurate model between them. These models were tested and worked as feature extractors and classifiers. We first upload all the datasets we have prepared before by separating them into two classes mask and unmask and converting all images to a jpeg extension. Then the dataset was split into 80% for training the model and 20% for testing it. Then ran the network, the network's convolution layers began to extract features and gather the information that the final learnable and classification layers used to classify. Then the function systematically identifies the learnable and classification layers that need to be replaced. Preserving the fundamental stability of the network. The first layers' learning rates are set to zero. This speeds up the process and eliminates the need for significant retraining by preserving the important properties that were learned during the pre-trained network's first training. In addition to using data augmentation to improve the network's capacity to generalize to fresh, untested data, avoid overfitting, and raise the model's overall performance. Also, we set hyperparameters as mini-batches setting to 64, epochs 5 and the initial learning rate to 1e-6, This ensures that the network is trained efficiently and effectively. Finally, the model classifiers classify the test data compute the confusion matrix and calculate the evaluation metrics.

3.2.3 Third proposed model system

After testing the customized Deep convulsion neural network model and pre-trained algorithms, we developed a hybrid system consisting of two sections which is known as Transfer Learning with Feature Extraction. In this approach, a custom CNN is trained to extract meaningful features from the data, and these features are then fed into a pre-trained model or classifier for the final classification task. So, the first part is trained images by CNN architecture and then extracting features from the trained network and training a classifier on these extracted features for classification. We use this model to get benefit from both CNN and pretrained algorithms and make the most of them. Algorithms to get the highest accuracy, more stability and less loss. To determine the best performance, we examined them before by setting the most effective hyperparameters and then we used the most effective algorithms as a classifier.

First, we employ CNN architecture consisting of an input layer specifying the size 227 x 227 pixels with 3 channels RGB so first, we resized images to fit the size and converted greyscale images to RGB and split them 80% for training and 20% for testing. We tried using the augmentation technique, but we got lower accuracy and a slower process, so we have dispensed. We use three convolution layers to extract the feature map, and we use the RELU function for the convolution layer and one for the fully connected layer. In pooling we preferred using max pooling with 2 strides and pool size 2 x 2. and we used two fully connected layers, one with 256 output neurons and the other with several neurons equal to the number of classes in the training dataset. SoftMax is applied and the final classification layer and we adjust the hyperparameter max epochs 7, And the initial learning rate ant 0.01, then extract test and train labels and the trained network and save them.

Second, we trained the classifier using pre-trained algorithms on the pre-extracted features from a pre-trained network and evaluated its performance on a test set. This strategic approach bypasses the computationally intensive process of retraining a CNN from scratch, instead capitalizing on feature representations already learned from a vast dataset. We apply different classifiers: Alexnet and Darknet, and the architecture (\`layers\`) is configured to accept these pre-extracted features for those different classifiers, adapting input dimensions accordingly to seamlessly integrate with the transferred knowledge. Training is streamlined using \`train Network\`, where the emphasis lies on fine-tuning the classifier to learn discriminative patterns specific to the target dataset, ensuring effective generalization and performance. Setting the perfect parameters that fit each classifier and extracting a confusion matrix to compute metrics such as accuracy, sensitivity (Recall), and specificity, provide us with comprehensive insight into the classifier's efficacy and make it possible to compare between classifiers' accuracy and losses too.

We proposed this method not only to accelerate model development but also to get higher accuracy and stability and enhance its adaptability across diverse domains and datasets, where precise and reliable classification is pivotal for decision-making and automation tasks.

4. Results

4.1 System implementation

The frameworks that have been implemented have been tested and reviewed on the following software and hardware configurations:

- Processor Intel(R) Core (TM) i7-6820HQ CPU @ 2.70GHz, 2701 MHz, 4 Core(s), 8 Logical Processor(s).
- Operating system: Windows 10 Pro.
- Installed RAM: 16.0 GB (15.9 GB usable).
- System type: 64-bit operating system, x64-based processor.
- Compiler: MATLAB R2020b.

4.2 Evaluation metrics

The research employs various metrics to evaluate the performance of a classification model on automatic image classification using machine learning. The traditional accuracy metric may not be sufficient if the distribution of class labels is imbalanced. Precision-recall metrics are especially useful when dealing with highly imbalanced classes as it can simply guessing the majority class for all but a few instances, and performing poorly on the minority class, in imbalanced datasets (when one class has a much higher number than the other). So thus metrics are typically recommended, where precision measures the ability of the model to identify relevant data points while recall measures the ability of the model to find all relevant cases. Furthermore, the confusion matrix is a suitable method for summarizing the performance of a classification algorithm in the context of imbalanced classes [21][22].

The subsequent equations are commonly employed metrics in the field of machine learning. In these metrics:

- True Positive (TP): an outcome where the model correctly predicts positive values.
- False Positive (FP): is an outcome where the model incorrectly predicts the positive values.
- False Negative (FN): is an outcome where the model incorrectly predicts the negative values.
- True Negative (TN): is an outcome where the model correctly predicts the negative values.

Accuracy: this metric measures how the classifier predicts correctly all the classes to the total number of instances.

$$\boldsymbol{accuracy} = \frac{\boldsymbol{TP} + \boldsymbol{TN}}{\boldsymbol{TP} + \boldsymbol{FP} + \boldsymbol{TN} + \boldsymbol{FN}} \qquad (\mathbf{1})$$

Recall (sensitivity): it is also called true positive Rate It calculates the percentage of real positives in a classification problem that are genuine positives. This measure shows how consistently the model labels positive units in the dataset.

$$recall = \frac{TP}{TP + FN} \qquad (2)$$

Precision: determines the percentage of true positives in a classification issue relative to all positive predictions. This score shows how reliable the model is at classifying a person as positive.

$$precision = \frac{TP}{TP + FP} \qquad (3)$$

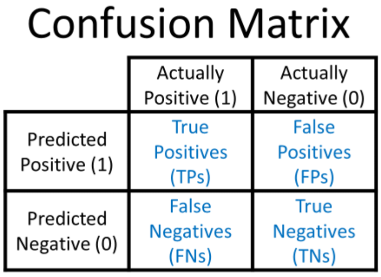
Specificity (or true negative rate, TNR): This metric measures the ratio of correctly predicted negative cases to all actual negative cases.

$$specificity = \frac{TN}{TN + FP} \qquad (4)$$

F1-Score combines the precision and recall scores of a model. F1-score is especially useful when dealing with imbalanced datasets where one class has significantly more samples than the other.

$$F1 - Score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \qquad (5)$$

Confusion Matrix: The most comprehensive performance metric for a classification model is the confusion matrix, represented in Figure 5. which provides an in-depth analysis of the model's behavior. In the case of a binary classifier, the confusion matrix displays a matrix that helps to evaluate the model's overall performance. The rows of the matrix correspond to the model's predictions, whereas the columns correspond to the true labels of the data samples. By analyzing the confusion matrix, we can gain insight into the model's strengths and weaknesses, which can guide us in refining further training to improve the model [23].



Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

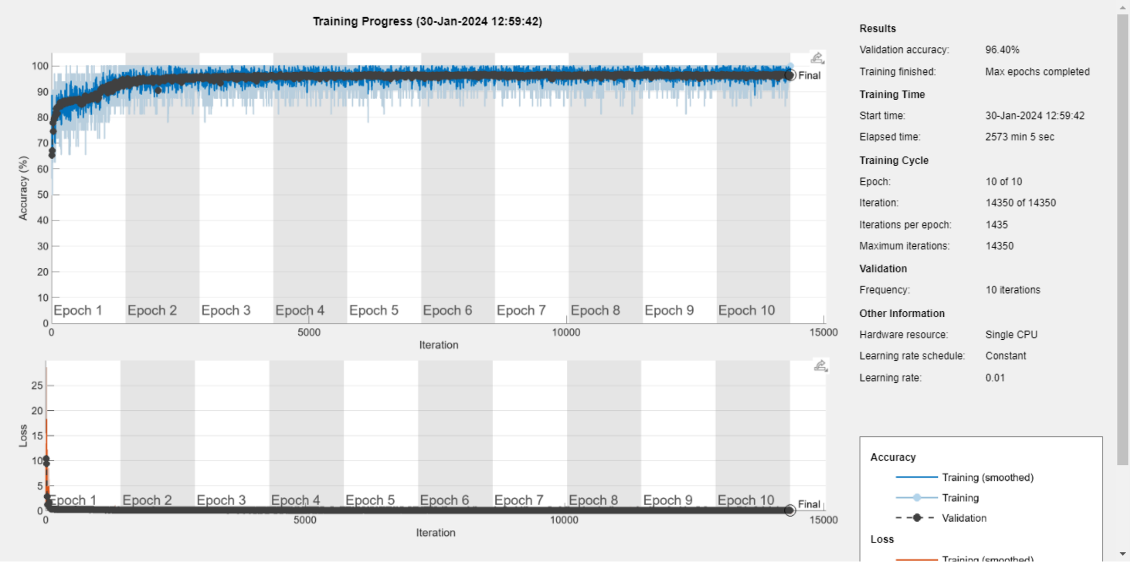
Figure 5. Confusion Matrix [23]

4.3 Training and Results

In this section, we show and discuss the findings and parameters we utilized in the three proposed methods and techniques. In the first proposed technique, we utilized a CNN model and in the second proposed system, we tried seven pre-trained algorithms including Alexnet, Resnet-50, Draknet-53, YOLOV5, Squeeznet, Googlnet and Efficientb0.and the last proposed system is we deployed an in third system that In is s hybrid In system combining CNN for feature extraction and a classifier for final classification. The hybrid system utilized the advantages of both CNNs and pre-trained algorithms to achieve higher accuracy, stability, and lower loss. After extracting features and labels from the trained network, we trained classifiers (AlexNet, and DarkNet) on these features, evaluating their performance on a test set. The hybrid system demonstrated that using a CNN for feature extraction followed by a classifier for final classification can effectively leverage pre-trained algorithms, resulting in higher accuracy and stability. The most effective parameters were determined through extensive experimentation, and the model's performance was assessed using confusion matrices to compute metrics such as accuracy, sensitivity (recall), and specificity. Overall, this approach accelerates model development, enhances accuracy, and improves adaptability across diverse domains and datasets, making it suitable for tasks requiring precise and reliable classification. The methodologies proposed in this study can significantly contribute to practical applications in face mask detection and other image classification tasks, offering valuable insights and potential for future research.

4.3.1 The First Proposed System

We proposed a CNN model to classify the images into two classes, first, we uploaded the dataset and split it into 80% for training which is equal to 45940, and 20% for testing which is equal to 11485, then the images resized to 224 x 224 pixels with 3 RGB Channels, the model system composed of Six convolutional layers with filters of sizes 32, 64, 128, 256, 512, and 1024, each with a 3x3 kernel size and 'same' padding and each convolution layer followed by batch normalization layer and ReLU activation function, max pooling layer and A fully connected layer with 1024 neurons followed by a ReLU activation and a dropout layer with a 0.6 dropout rate to prevent overfitting. Then the output layer was a fully connected layer with 2 neurons, followed by a SoftMax layer to convert the outputs to probabilities. The network was trained using Stochastic Gradient Descent with Momentum (SGDM), and Figure 6. presents the accuracy and loss curves that observed during the training process of the proposed CNN model. The graph shows a stable improvement in accuracy over successive epochs, proving that the model's has ability to learn and generalize effectively. Simultaneously, the loss function shows a continuous decline, indicating that the network is successfully minimizing classification errors. The convergence of the training and validation curves shows that the model does not suffer from overfitting and is well-optimized for real-world deployment. The CNN model achieved an accuracy of 96.4%, with a sensitivity of 92.68%, proving its effectiveness in distinguishing between masked and unmasked individuals.



Results
Validation accuracy: 96.40%
Training finished: Max epochs completed
Training Time
Start time: 30-Jan-2024 12:59:42
Elapsed time: 2573 min 5 sec
Training Cycle
Epoch: 10 of 10
Iteration: 14350 of 14350
Iterations per epoch: 1435
Maximum iterations: 14350
Validation
Frequency: 10 iterations
Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.01

Figure 6. Accuracy and Loss During the training process by the proposed CNN

4.3.2 The second Proposed System

Pre-trained models that were examined are Alexnet, Squeeznet, Resnet-50, Efficientb0, YOLOV5, Googlenet and Darknet-53. These models were tested and worked as feature extractors and classifiers. We first uploaded all the datasets we have prepared before by separating them into two classes mask and unmask and converting all images to jpeg extension. Then the dataset was split into 80% for training the model and 20% for testing it. Then ran the network, the network's convolution layers began to extract features and gather the information that the final learnable and classification layers used to classify. Then the function systematically identifies the learnable and classification layers that need to be replaced. This ensures that the network's final layers are appropriate for the new classification task, maintaining the integrity of the network's structure. We set the learning rates of the initial layers to zero. This preserves the valuable features learned during the original training of the pre-trained network, reducing the need for extensive retraining and speeding up the process. in addition to applying data augmentation to enhance the network's ability to generalize to new, unseen data and prevent overfitting and improve overall the performance of the model. Also, we set hyperparameters as mini-batches setting to 64, epochs 5, and initial learning rate to 1e-6, This ensures that the network is trained efficiently and effectively. Finally, the model classifiers classify the test data compute the confusion matrix and calculate the evaluation metrics. Figure 7. presents a comparative analysis of the accuracy of the pre-trained algorithms used in this study. Among the seven tested models, Darknet-53 achieved the highest accuracy of 97.51% and stability as shown in figure 8. While, Other models, such as ResNet-50, AlexNet, and GoogleNet, also performed well, with accuracy values exceeding 96%, but slightly below Darknet-53. On the contrary, YOLOv5 and EfficientNet-B0 get the lowest accuracies at 69.4% and 54.1%, respectively, indicating their limitations in this specific classification task. This comparison highlights Darknet-53 as the most effective pre-trained model, making it a suitable model system for integration into the hybrid system to enhance classification accuracy and overall model stability. and in Table 1 the comparison between all pre-trained algorithms we apply with their accuracies, precisions, and f1-score.

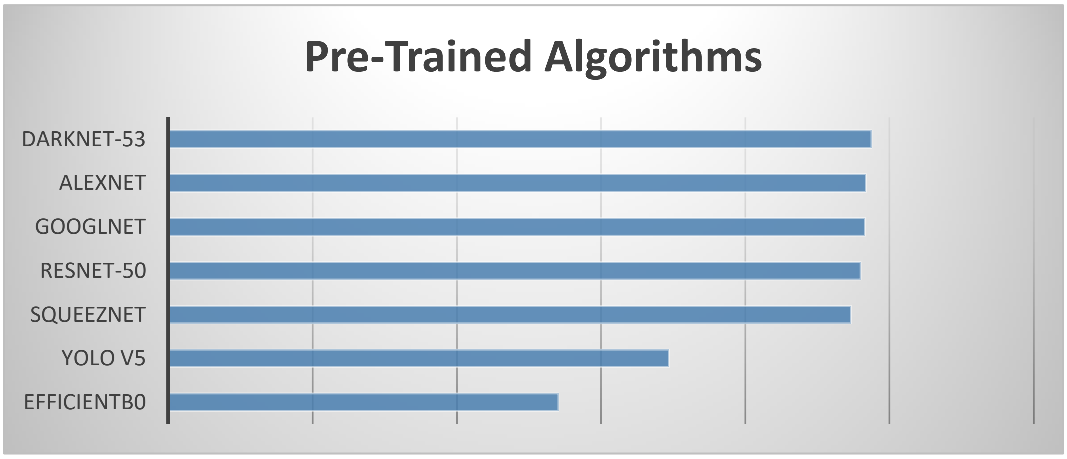
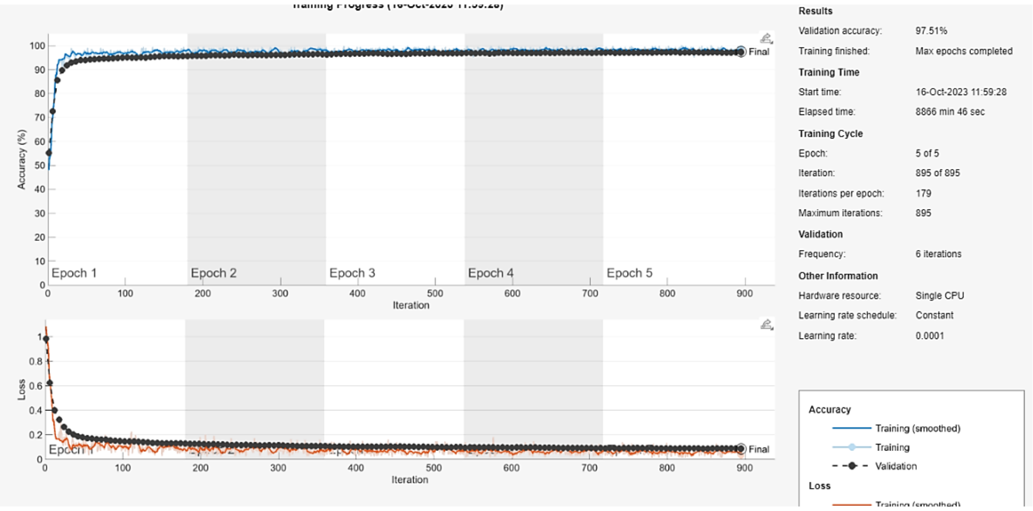


Figure 7. Accuracy comparison chart among pre-trained Algorithms.

Table 1. The performance of the pre-trained models and the best values are shown in bold.

Algorithm	Accuracy	Precision	F1-Score	Sensitivity	Specificity
Darknet-53	**97.51%**	97.95%	**96.35%**	**94.81%**	98.94%
Alexnet	96.76%	97.3%	95.2%	93.23%	98.64%

Googlenet	96.64%	97.09%	95.06%	93.11%	98.51%
Resnet50	96.02%	**98%**	94%	90.37%	**99.025%**
Squeeznet	94.68%	93.49%	92.24%	91.03%	96.62%
YoloV5	69.4%	83.7%	74.3%	67%	74.1%
Efficient-B0	54.096%	36.41%	45%	42.99%	60%



Results

Validation accuracy: 97.51%

Training finished: Max epochs completed

Training Time

Start time: 16-Oct-2023 11:59:28

Elapsed time: 8866 min 46 sec

Training Cycle

Epoch: 5 of 5

Iteration: 895 of 895

Iterations per epoch: 179

Maximum iterations: 895

Validation

Frequency: 6 iterations

Other Information

Hardware resource: Single CPU

Learning rate schedule: Constant

Learning rate: 0.0001

Figure 8. Accuracy and Loss During the training process by the Darknet algorithm

4.3.3 The Third Proposed System

After the two proposed systems that we have discussed, we proposed a new hybrid system to take the benefits from both Models, so we use CNN as a feature extractor along with the pre-trained algorithms as a classifier to increase the accuracy and stability of the model system. Som the CNN model which we use as a feature extractor build of convolutional layer with a 3x3 filter and 256 filters, ReLU activation function also SoftMax at the output layer, 2 layers of max pooling in addition to one fully connected layer, Adaptive Moment Estimation (adam) is better as optimizer as it adjusts the learning rates for each parameter dynamically, which helps in converging faster and avoiding issues like vanishing or exploding gradients. We try augmentation but it gives us lower accuracy, so we ignore using this technique as the dataset is enough. After training the model then extracts test and train labels and the trained network and uses them to be the input for the pre-trained algorithm as Alexnet or Darknet.

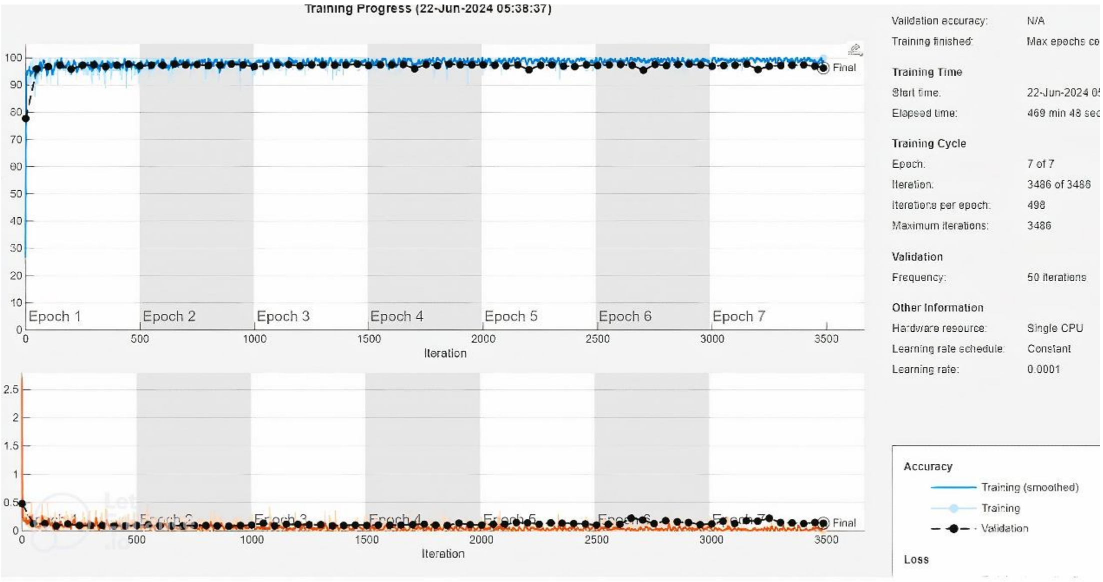
4.3.3.1 CNN-Alexnet

We load the trained network and the extracted features and labels as input to the alexnet model. We retain all layers except the last three layers which are responsible for image classification and add a new fully connected layer equal to the number of the classes in the trained network and a new softmax and classification output layer. Then the Alexnet network uses the trained network and labels with specified layers and options for optimal performance and a more stable model.

The accuracy we get from CNN-Alexnet is 97.74%.

4.3.3.2 CNN-darknet 53

We load the trained network and the extracted features and labels as input to the darknet model. We retain all layers except the last three layers and darknet 53 (YOLOV3) has a complex structure with residual connections, we need to ensure that the final layers are correctly replaced and connected. Which are responsible for image classification and add a new fully connected layer equal to the number of the classes in the trained network and new SoftMax and classification output layer and choose an appropriate layer for feature extraction Then train the darknet-53 network using the trained network and labels with specified layers and options for optimal performance and more stable model. And Figure 9 illustrates the CNN-Darknet53 hybrid model's system accuracy and stability throughout training by displaying the accuracy and loss curves. To improve stability and accuracy, this model combines a CNN as a feature extractor with Darknet-53 as the last classifier, as we mentioned before. and the curve indicates that there is a steady increase in accuracy over epochs, indicating that the model is effectively learning features from the dataset. and the loss curve shows a consistent decline, confirming that classification errors are being minimized throughout the training process. The CNN-Darknet53 model achieved the highest accuracy of 98.20%. This improvement highlights the advantage of combining CNN feature extraction with a powerful classifier like Darknet-53, ensuring a more robust and efficient face mask detection system and the confusion matrix for this method is shown in Figure 10 providing a detailed evaluation of its classification performance. The matrix visually represents the number of correctly classified masked and unmasked images, as well as misclassifications. The high values in the diagonal entries indicate a strong ability to accurately differentiate between the two categories.



Validation accuracy: N/A
Training finished: Max epochs co

Training Time
Start time: 22-Jun-2024 0
Elapsed time: 469 min 48 sec

Training Cycle
Epoch: 7 of 7
Iteration: 3486 of 3486
Iterations per epoch: 498
Maximum iterations: 3486

Validation
Frequency: 50 iterations

Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.0001

Figure 9. Accuracy and loss during the training process by CNN-Darknet

7380	114
80	3911

Figure 10.Confusion matrix for CNN-Darknet

Table 2. The performance of the hybrid models and the best values are shown in bold and underlined.

System model	Accuracy	Precision	Sensitivity	Specificity	F1-score
CNN-Alexnet	97.74%	98.34%	99.18%	96.89%	98.167%
CNN-Darknet53	98.00%	98.42%	98.33%	97.07%	98.38%

From the above table 2. The proposed system models which extract features by a built-up system using CNN and then apply Alexnet and Darknet -53 are more accurate and stable.

5. Conclusions

In this study, we must improve the facemask detection system's efficacy because it is crucial in the age of airborne illness transmission and the rise in disease-related fatalities. We investigated the efficiency of facemask detection using three suggested techniques to facilitate surveillance, particularly in congested areas like hospitals. AlexNet, DarkNet-53, ResNet-50, GoogleNet, SqueezeNet, YOLOV5, and EfficientNet-B0 were the seven pre-trained algorithms used in the first technique. DarkNet-53 performed the best, with an accuracy of 97.51%. The second technique used CNNs to customize a system model, which has a high sensitivity and 96.4% accuracy. By integrating a CNN for feature extraction with a pre-trained algorithm as the classifier for final predictions, we created a third hybrid system model. The advantages of CNNs and pre-trained algorithms are used in this hybrid technique to improve accuracy and stability and reduce loss. We use classifiers like AlexNet or DarkNet after removing features and labels from the trained network. The outcomes showed that using a CNN for feature extraction and then a pre-trained classifier substantially boosts performance, attaining the highest accuracy and stability. Overall, this method is appropriate for tasks requiring accurate and trustworthy classification since it speeds up model construction, improves accuracy, and increases adaptability across various domains and datasets. The approaches put forward in this paper have the potential to greatly advance real-world applications in picture classification tasks, such as face mask recognition, while also providing insightful information and avenues for further investigation.

References

[1] World Health Organization. (2024, March). COVID-19 deaths | WHO COVID-19 dashboard. Retrieved from Datadot website: https://data.who.int/dashboards/covid19/deaths

[2]World Health Organization. (2023, October 3). Influenza (Seasonal). Retrieved from Who.int website: https://www.who.int/news-room/fact-sheets/detail/influenza-(seasonal)

[3] The government makes wearing face masks mandatory. (2021, May 19). The British Medical Association Is the Trade Union and Professional Body for Doctors in the UK. https://www.bma.org.uk/news-and-opinion/government-makes-wearing-face-masks-mandatory.

[4] Zou, X. (2019). A Review of Object Detection Techniques. 2019 International Conference on Smart Grid and Electrical Automation (ICSGEA). https://doi.org/10.1109/icsgea.2019.00065

[5] De Fazio, R., Giannoccaro, N. I., Carrasco, M., Velazquez, R., & Visconti, P. (2021). Wearable devices and IoT applications for symptom detection, infection tracking, and diffusion containment of the COVID-19 pandemic: a survey. Frontiers of Information Technology & Electronic Engineering/Frontiers of Information Technology & Electronic Engineering, 22(11), 1413–1442. https://doi.org/10.1631/fitee.2100085

[6] Melo, C., Dixe, S., Fonseca, J. C., Moreira, A. H. J., & Borges, J. (2021b). AI-Based Monitoring of Different Risk Levels in COVID-19 Context. Sensors, 22(1), 298. https://doi.org/10.3390/s22010298

[7] Al-Tamimi, M. S. H., & Mohammed Ali, F. A. (2023). Face mask detection based on algorithm YOLOv5s. International Journal of Nonlinear Analysis and Applications, 14(1), 679–697. https://doi.org/10.22075/ijnaa.2022.28178.3824

[8] Umer, M., Sadiq, S., Alhebshi, R. M., Alsubai, S., Hejaili, A. A., Eshmawi, A. A., Nappi, M., & Ashraf, I. (2023). Face mask detection using deep convolutional neural 65 network and multi-stage image processing. Image and Vision Computing, 133, 104657. https://doi.org/10.1016/j.imavis.2023.104657

[9] Loey, M., Manogaran, G., Taha, M. H. N., & Khalifa, N. E. M. (2021b). Fighting against COVID-19: A novel deep learning model based on YOLO-v2 with ResNet-50 63 for medical face mask detection. Sustainable Cities and Society, 65, 102600. https://doi.org/10.1016/j.scs.2020.102600

[10] Sheikh, B. U. H., & Zafar, A. (2023). RRFMDS: Rapid Real-Time Face Mask Detection System for Effective COVID-19 Monitoring. SN Computer Science/SN Computer Science, 4(3). https://doi.org/10.1007/s42979-023-01738-9

[11] Wang, S., Wang, X., & Guo, X. (2023). Advanced Face Mask Detection Model Using Hybrid Dilation Convolution Based Method. Journal of Software Engineering and Applications, 16(01), 1–19. https://doi.org/10.4236/jsea.2023.161001

[12] Melo, C., Dixe, S., Fonseca, J. C., Moreira, A., & Borges, J. (2021b). MoLa RGB CovSurv. Mendeley Data. https://doi.org/10.17632/vzf939jbxy.1

[13] Liu, Z., Luo, P., Wang, X., & Tang, X. (2014). Deep Learning Face Attributes in the Wild. arXiv (Cornell University). https://doi.org/10.48550/arxiv.1411.7766

[14] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common Objects in Context. In Lecture notes in computer science (pp. 740–755). https://doi.org/10.1007/978-3-319-10602-1_48

[15] Le, V., Brandt, J., Lin, Z., Bourdev, L., & Huang, T. S. (2012b). Interactive Facial Feature Localization. In Lecture notes in computer science (pp. 679–692). https://doi.org/10.1007/978-3-642-33712-3_49

[16] Nordstrøm, M.M., Larsen, M., Sierakowski, J., & Stegmann, M.B. (2004). The IMM Face Database: An Annotated Dataset of 240 Face Images. Available online: http://www.imm.dtu.dk/~aam/aamexplorer/ (accessed on 15 June 2021).

[17] Yang, S., Luo, P., Loy, C. C., & Tang, X. (2015b). WIDER FACE: A Face Detection Benchmark. arXiv (Cornell University). https://doi.org/10.48550/arxiv.1511.06523

[18] Understanding images of groups of people. (2009, June 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/5206828

[19] Liu, Z., Luo, P., Wang, X., & Tang, X. (2014). Deep Learning Face Attributes in the Wild. arXiv (Cornell University). https://doi.org/10.48550/arxiv.1411.7766

[20] Harley, A. W. (2015). An Interactive Node-Link Visualization of Convolutional Neural Networks. In Lecture notes in computer science (pp. 867–877). https://doi.org/10.1007/978-3-319-27857-5_77

[21] Journal of Machine Learning Technologies. (n.d.). Bio info Publications, 2229-3981(6). https://doi.org/10.9735/2229-3981

[22] Ahmadzadeh, R., & Angryk, R. A. (2022). On the sensitivity of performance metrics to class imbalance in binary classification. arXiv preprint arXiv:2206.09981. https://arxiv.org/pdf/2206.09981

[23] Draelos, V. a. P. B. R., MD PhD. (2019, May 18). Measuring Performance: the Confusion Matrix. Glass Box. https://glassboxmedicine.com/2019/02/17/measuring-performance-the-confusion-matrix/