# Cascade Detection Technique for Face Mask Recognition Based on YOLOv9 and CNN

**Amal Sufiuh Ajrash[1,*], Wildan Jameel Hadi[1], Ammar Hussein Jassim[1]**
[1]The College of Science for Women, University of Baghdad, Iraq
Emails: amalsa_comp@csw.uobaghdad.edu.iq; wildanjh_comp@csw.uobaghdad.edu.iq; ammarhj_comp@csw.uobaghdad.edu.iq

**Abstract**

Deep learning showed promise in many real-world applications. Recognition and item identification are the most common. This publication tries to design and describe a system that can classify people from images based on whether they are correctly wearing masks. The proposed system is two-part. The first part is designed for facial detection using the YOLOv9 (You Only Look Once version 9) compact deep learning model, which uses the mean intersection method over union to determine an optimal number of anchoring boxes and the Adam optimizer to improve facial detection efficacy. The second component is a convolutional neural network for face feature extraction. These faces are classified as a mask, without_mask, and incorrect_mask. These two components are integrated into the proposed system for facemask recognition. Empirical evaluations were conducted on the two self-collected datasets to train and evaluate the proposed system's performance. The observed precision value of this system was 94.6% in the last epoch; the recall value is 87.1%, and the mean average precision results are 92.74% as a face detector. The classifier model train accuracy is 98.35%, and validation accuracy is 98.8%. Finally, the comparative results indicated that the proposed framework was an effective model for face detection, attaining a higher mean average precision value and outperforming other networks assessed on the designated dataset for this task. The suggested network effectively detects and classifies several faces in photos, including small faces in congested places.

**Keywords:** Deep Convolutional Neural Network (DCNN); Face Mask Detection; Face Mask Recognition; Machine Learning (ML); YOLOv9

## 1. Introduction

The best way to prevent the spread of viral infections is to wear masks. In recent years, Various Artificial intelligence technologies have been utilized in multiple ways to support efforts in preventing viruses, and computer vision techniques have become a hot topic in preventing viruses from spreading. They are a rapidly growing technology set to revolutionize healthcare. It leverages powerful artificial intelligence algorithms with optical sensors and cameras [1]. In addition, significant advancements have been made with the expansion of technology and the fast evolution of computers in building applications to recognize wearing masks. Therefore, in the last few years, governmental and private organizations have adopted several safety standards to restrict the spread of infections, such as the COVID-19 epidemic. One is the requirement that facemasks be worn in public places.

Currently, With the growing success and continuous development of deep learning, deep convolution neural networks (DCNNs) have sparked significant attention in various uses, including machine vision [2] uses several scenarios, such as object detection, semantic segmentation, object recognition, facial recognition [3], motion detection, image classification, Stock Market Prediction, and vehicle detection [4] and medical imaging. In many of these applications, DNNs can now exceed human-level accuracy. Their goal is to extract the desired visual characteristics from images that are created at random. Therefore, various ways for recognizing facial masks using deep learning algorithms have been developed to increase human protection. These algorithms utilize a deep



DOI: https://doi.org/10.54216/JCIM.170109
Received: February 28, 2025 Revised: May 30, 2025 Accepted: July 20, 2025

learning framework to identify people wearing masks or not in public places, for example, the RPN (region proposal network) [5], and the FR-CNN (faster region-based convolutional neural networks) methods. In contrast, these approaches have poor detection speeds, especially if deployed on lightweight computing devices.

Deep learning-based object identification algorithms can primarily be classified into two distinct types with respect to their classification. The first such method is a two-stage detection technique that depends upon candidate boxes represented by SPP net, R-CNN, Fast-RCNN, Faster R-CNN [6], Mask R-CNN, and FPN. Based on feature extraction, this algorithm first generates many candidate regions by independent network branches and then classifies and regresses them. The other is a one-stage detection technique based on a regression analysis that SSD, DSSD, RetinaNet, EfficientDet, and YOLO [7,8]. In addition to creating candidate frames, these algorithms are responsible for carrying out operations like classification and regression by utilising forward prediction via the network. Compared to the two-stage detection approach, these methods are substantially faster.

The YOLO (You Only Look Once) algorithm [9] is an innovative deep-learning technique that enhances and speeds up traditional approaches. YOLO has been proven ten times more effective than fast techniques like Faster R-CNN.

The innovation comes in the fusion of YOLOv9 with a CNN to perform both face detection as well as identifying whether or not there is a mask present on the detected faces, enabling better detection while being computationally more efficient than prior methods. Emphasize that the adoption of PGI in the YOLOv9c model is an important progress, which guarantees retention of critical context information and improves the model's effectiveness, making it a better solution over previous generations of YOLO. Additionally, highlight that this research offers a combined solution using two modern techniques that have not been widely explored within the real-time context of facemask detection, in particular, environments with large crowds. Thus, this work introduces a new cascade detection approach for facemask detection that establishes an improved state-of-the-art (SOTA) by introducing the following:

1. **YOLOv9c Face Detection with Programmable Gradient Information (PGI):** YOLOv9c is employed as a face detector, utilizing the Programmable Gradient Information (PGI) to preserve essential information during the gradient flow, which is a key innovation that enhances the model's efficiency and accuracy. This method significantly improves face detection, particularly in crowded areas or when identifying small faces. It raises mAP by 5.1% compared to YOLOv8, showing a clear improvement in detection abilities.
2. **CNN for Mask Classification:** The CNN classifier is designed to categorize faces into three distinct mask-wearing conditions: with mask, without mask, and incorrect mask. Using YOLOv9c to find faces and CNN to classify masks makes an end-to-end system that can classify things with high accuracy, with a training accuracy of 98.36% and a validation accuracy of 98.8%. This integration has been tested with real-world datasets and gives a huge performance boost over older models.
3. **Use of Custom Datasets:** Instead of relying solely on pre-existing datasets, the study utilizes two custom datasets for training and validation, ensuring the system is specifically tailored for the mask-wearing classification problem. This enhances the model's capacity to manage the complexities and variances of the real world.
4. **System Efficiency and Scalability:** The proposed system's optimally designed architecture that combines GPU optimization and the parallel process of utilizing multiple CPU cores demonstrates considerable improvement in training time and performance. This is especially important in typical real-world applications in places like schools, shopping centers, or public spaces where analysis is performed in real time.
5. **Comparison to Existing Models:** The findings show that YOLOv9c surpasses comparable models, such as YOLOv5 and YOLOv7, in its precision of face detection, (92.74% mAP) as well as its adaptability on challenging and diverse datasets. The CNN used for mask classification, likewise, provides improved precision and recall than previously researched and evaluated systems demonstrating good capabilities for a range of facemask detection applications.

Overall, the method represents major advances by integrating face detection and mask classification in a single, efficient framework (see Figure 1), ensuring greater accuracy at low computational cost, which is paramount for real-world use. This research advances the field of mask recognition, paving the way for further future work or enhancement based on newer technology using YOLOv10 or YOLOv12.



DOI: https://doi.org/10.54216/JCIM.170109
Received: February 28, 2025 Revised: May 30, 2025 Accepted: July 20, 2025

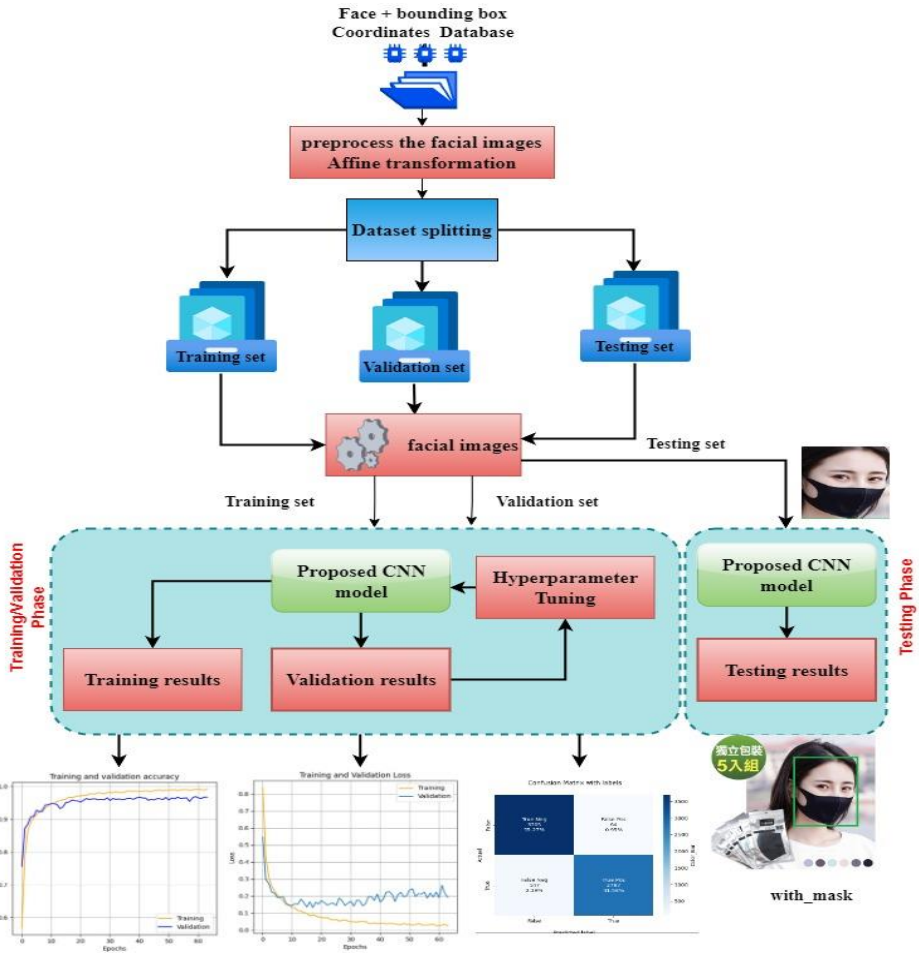



**Figure 1.** Block Diagram of Proposed Mask Detection and Classification Model

## 2. Related Work

Many studies give high attention to the people who wear facemasks and try to determine their identity. The following will display FR studies that different researchers have suggested.

In [10], a mask identification method using deep learning and the Convolutional Block Attention Module (CBAM) is presented. Derives representative characteristics from incoming photos using supervised neural networks to enhance recognition accuracy with constrained computational resources. The YOLOv7 network model was selected, and CBAM was included in its architecture compared to the original iteration of YOLOv7. In this paper, the dataset considered is the Face Mask Detection (FMD) dataset. The suggested model enhances the mean Average Precision (mAP) by 95.4% in the face mask detection process.

In [11], suggested to use YOLOv8n-SLIM-DYHEAD, an enhanced version of the YOLOv8 method, for face mask detection. Can make it more applicable to scenes where people are wearing masks by the mosaic data augmentation method enhances the model's general applicability for real-world detection algorithms. To simplify the model without sacrificing accuracy, the Slim-neck structure is employed on the neck net to merge features of varying sizes that have been extracted using the leading net. To mark to the feature variation in the detecting layer brought about by variances in target size and form position, DyHead is employed. The modified YOLOv8n-SLIMDYHEAD method achieved mAP @0.5 of 95.7% as well as mAP @0.5:0.95 of 65.0%, according to the experiments.

In [12] proposes a new deep-learning detecting model that can effectively locate a medical mask-wearing face in an image. YOLOv8 precisely determines facemasks in this investigation. For this work, they integrated the Face Mask Dataset (FMD) and Medical Mask Dataset (MMD) into one dataset. YOLOv8s, YOLOv8n, and YOLOv8m are compared. The training result for FMD and MMD using YOLOv8n achieved a mAP@.5 of 61.9%, YOLOv8s attained 70.4%, and YOLOv8m reached 78.4% over 100 epochs, surpassing previous models applied to FMD and MMD in terms of mAP. The proposed YOLOv8m model enhanced detection accuracy for FMD and MMD, achieving a 99.1% mAP, surpassing previous models employed for these tasks.



DOI: https://doi.org/10.54216/JCIM.170109
Received: February 28, 2025 Revised: May 30, 2025 Accepted: July 20, 2025

The authors in [13] suggested a model utilizes the Viola-Jones algorithm for FR and the YOLOv5 technique for mask identification and classification. The suggested work demonstrates a mask detection success rate of 92.8% upon testing. The dataset used for training was taken from the Kaggle website, where the data was divided into 1167 for training and 233 for testing data. The model was trained for several iterations, where the model's accuracy reached 88.27% and validation precision was 88.17% in iteration 30.

For facial feature extraction, a modified Res2-Net structure called Im-Res2Net-101 was employed in [14]. The authors propose a facemask detection system called FMD-Yolo. Low-level details and high-level semantic information were combined using an aggregation network called En-PAN. Experiments showed that FMD-Yolo achieved average precisions of 92.4% and 88.4% for two separate datasets at the IoU level of 0.5.

In [15] authors investigate the performance of the YOLOv8 object detection algorithm to characterize its work depending on people who wear the mask into three classes. The FMD dataset was used to train the proposed system. They used two models, the YOLOv8 and YOLOv5. YOLOv8 uses transfer learning methods on FMD dataset to classify face mask-wearing conditions without error. The YOLOv5 model training on the same dataset for comparative analysis. They achieve mAP (0.5) 78% for 150 epochs and 96% for 200 epochs. For yolov5 mAP (0.5), 79% for 150 epochs and 86% for 200 epochs.

An improved YOLO-v4 model for facemask recognition and standard wear detection was suggested in [16]. In order to make the YOLO-v4 algorithm work better, researchers enhanced CSPDarkNet53, optimized PANet's structure, decreased complexity and redundancy in the flexible image scaling technique and created a collection of images for facial mask recognition based on how people often wear masks. The experimental findings demonstrate that the suggested model of face mask identification achieves an impressive mAP of 98.3% and an excellent frame accuracy of 54.57 FPS when compared to other well-known algorithms for facemask recognition, like SSD, and YOLO-v4.

YOLOv4-Tiny, YOLOv5l, YOLOv5m, YOLOv5s, and YOLOv5x were utilized in [17] to test the used dataset. The images in the dataset depict individuals in crowded settings, both wearing and not wearing medical masks. Compared to YOLOv5s, which achieved a maximum acceleration of 2.1 ms and a maximum mAP of 45.3%, YOLOv4-Tiny attained a mAP of 55.5% with a processing speed of 4.0 milliseconds. It has been observed that YOLOv5x not only achieves the lowest speed of 6.1 ms but also the lowest mAP of 43.7%.

## 3. YOLOv9 Face Detection

YOLOv9 is one of the most popular models in the area of object detection in computer vision. This deep neural network predicts the bounding boxes and classification probabilities of objects in its input image or video frame, enabling the detection of objects in a timely and accurate manner. One of the main problems in deep learning is an "Information Bottleneck." When data are moving through a deep layer of neural networks, critical information could be lost, leading to incorrect prediction results. YOLOv9 is designed to explicitly deal with this problem by introducing an innovative concept called Programmable Gradient Information (PGI) that functions as a smart bypass lane in a tunnel. Essentially, this process ensures the preservation of necessary information for the accurate detection of objects within the network. This is accomplished by way of a reversible auxiliary branch in addition to the main processing path. This reversible branch helps the model "remember" and access pivotal information that would be lost [their main path], which will deliver valid gradients. These gradients optimize the learning process for modelling overall, which positively affects the overall detection performance. PGI comprises three main components: the inference-oriented primary branch structure, the auxiliary reversible branch delivering reliable gradients to the primary branch for back propagation, and an auxiliary signal at multiple levels, to control the primary functionalities. Central to YOLOv9 is a state-of-the-art architectural innovation known as GELAN, which provides several important advantages to the model:

- **Superior Parameter Utilization**: GELAN allows YOLOv9 to maximize its parameters, resulting in a leaner and more efficient model. This translates to faster processing and lower computational demands.
- **Computational Efficiency**: GELAN's design was created to effectively process information, enabling YOLOv9 to attain remarkable outcomes without sacrificing speed.
- **Flexibility**: GELAN's flexible architecture supports the convenient incorporation of different computational blocks. Because of this, YOLOv9 can be used in several applications without losing performance, adapting to different contexts as required.

Essentially, GELAN enhances the robustness and efficiency of YOLOv9, making it an effective tool trans versing through various challenging scenarios. PGI secures data when gradient updates occur, while GELAN enhances lightweight models through gradient path routing. YOLOv9 incorporates PGI and utilizes the adaptive architecture of GELAN to not only improve learning, but also protect sensitive data throughout the detection process. This leads to exceptional accuracy as well as effectiveness.



DOI: https://doi.org/10.54216/JCIM.170109
Received: February 28, 2025 Revised: May 30, 2025 Accepted: July 20, 2025

## 4. Datasets Characteristics

There are many medical facemask datasets; this research used two of them. The first one is the MMD [18], which consists of 682 images with over 3k medical masked faces wearing masks. samples of images in MMD are shown in Figure 2 (a). The second is the FMD [19]. It consists of 853 images. Some samples of the FMD are shown in Figure 2(b). The proposed system combines MMD and FMD leads to get 1415 images by removing bad quality and redundant images.

(a) MMD

(b) FMD

**Figure 2.** Samples of datasets

### 4.1 Image Pre-processing: Normalization and Scaling

Pre-processing is a crucial step in preparing the Medical Mask Dataset for model training. Two common techniques for image pre-processing are normalization and scaling:

**Normalization:** This refers to the act of changing the values of the pixels in the images to a specific range, typically between 0 and 1. This makes the model learn better through a lessening of the impact of the different values of the pixel intensity. Normalizing is done by dividing the pixel value by 255 (in the case of the pixel value being between 0-255). Normalizing takes away the uniqueness of normalized images so the model can understand the entire input effectively.

**Scaling:** Scaling changes the size of an image to a fixed value, which is helpful when working with deep learning models. Scaling ensures that the images are of the same shape and ready to be passed to the model as input. Typically scaling involves resizing the images to a predefined height and width, such as 224x224 pixels. Scaling images ensures efficiency of computation in a model and consistency in feature extraction.

By normalizing and scaling the images, the model can learn the relevant patterns more effectively, resulting in improved performance and accuracy in classifying faces based on mask-wearing status.

## 5. Integration of YOLOv9 and CNN for Face Mask Recognition

The suggested system for recognizing facemasks is a hybrid system that combines the YOLOv9 face detection algorithm with a CNN designed for mask classification. The integrated system has two primary stages: face detection and mask classification. We describe the data flow and fusion between the two systems to enhance seamless functionality.

### 5.1 Face Detection using YOLOv9

The first step of the system detects faces in the input image using the YOLOv9 model. YOLOv9 is a fast one-stage detector that provides bounding boxes with respective confidence scores for the detected faces. YOLOv9 was selected for its real-time object detection at very high accuracy. YOLOv9 uses a PGI method, which retains critical information at certain points throughout the network, so important data for face detection does not get lost. The detection consists of the following sub-steps:



DOI: https://doi.org/10.54216/JCIM.170109
Received: February 28, 2025 Revised: May 30, 2025 Accepted: July 20, 2025

- **Input Image**: The image is passed through the YOLOv9 model, where it is processed to detect faces.
- **Bounding Box Prediction**: The model generates bounding boxes that enclose the faces, each accompanied by a confidence score indicating the likelihood that a face has been correctly detected.
- **Feature Extraction**: The features from the detected faces are extracted, including positional information (x1, y1, width, height) and the confidence score, which are stored for use in the next stage.

### 5.2 Face Mask Classification using CNN:

Once faces are detected and their bounding boxes are identified, the cropped face regions (regions of interest or RoIs) are passed to the CNN classifier for mask classification. The CNN is designed to classify each detected face into one of three categories: "with mask," "without mask," and "incorrect mask." The CNN performs the following steps:

- **Input Preprocessing**: The detected face images (cropped from the original image) are pre-processed by resizing and normalizing the image data. The preprocessing ensures that the images are in a suitable format for the CNN input.
- **Feature Extraction and Classification**: The CNN consists of several convolutional layers designed to extract hierarchical features from the input images. These features are then passed through fully connected layers to classify the images into the three mask-wearing categories. The CNN architecture uses ReLU activations for hidden layers and a softmax function at the output layer to determine the final class of each face.
- **Output**: The CNN outputs the class label (mask, no mask, or incorrect mask) for each detected face.

### 5.3 Data Flow and Fusion Mechanism

The integration of YOLOv9 and CNN follows a sequential data flow:

- First, YOLOv9 processes the input image for face detection, resulting in bounding boxes for each detected face.
- The detected faces, cropped from the image according to the bounding boxes, are then passed as input to the CNN for classification.
- The CNN classifies each face into one of the three categories, and the results are mapped back to the original image with corresponding bounding boxes and classification labels.

This fusion of the two models allows for a comprehensive system that detects faces and classifies mask usage with high accuracy. YOLOv9 handles the real-time detection of faces, while the CNN ensures precise classification of mask usage based on the extracted facial features.

## 6. The Proposed Detector Model

The suggested mask recognition methodology uses deep neural networks, which are more efficient. It uses Yolov9 for face detection and CNN for classification purposes. Generally, just chain the models together and pipe the detector's outputs as inputs into the classifier model. However, deep neural network training is very costly because it requires a lot of computation and time. For this reason, the suggested model uses Num workers (processes on CPU cores) and GPU capabilities for data loading. The proposed model used two workers to prepare images and train the detection system and classifier. It divides the pre-processing activity into parts that may be completed concurrently and batches all of them for parallel processing (Threading Technique). Due to the independence of input images, each thread can process a portion of the input photos, allowing for parallelization of processes over several CPU cores. This can assist in leveraging the parallelism provided by today's CPUs and enhancing the pre-processing pipeline.

This system consists of two parts: The initial one is based on the YOLOv9 algorithm to detect faces in images. The second one receives the detected faces as an input image to detect and specify if the faces wear masks or not. It does that by using a CNN, which serves as the primary component of our classifier, where it’s responsible for extracting features from facial photos and transforming them into a feature map for detection and classification. The affine transformation is employed to identify facial characteristics due to potential variations in face dimensions and positioning inside the clipped region of interest. Figure 3 illustrates the main components of the proposed system, which shows the face detection and mask recognition process in the following subsections, and displays a description of the proposed architecture.



DOI: https://doi.org/10.54216/JCIM.170109
Received: February 28, 2025 Revised: May 30, 2025 Accepted: July 20, 2025

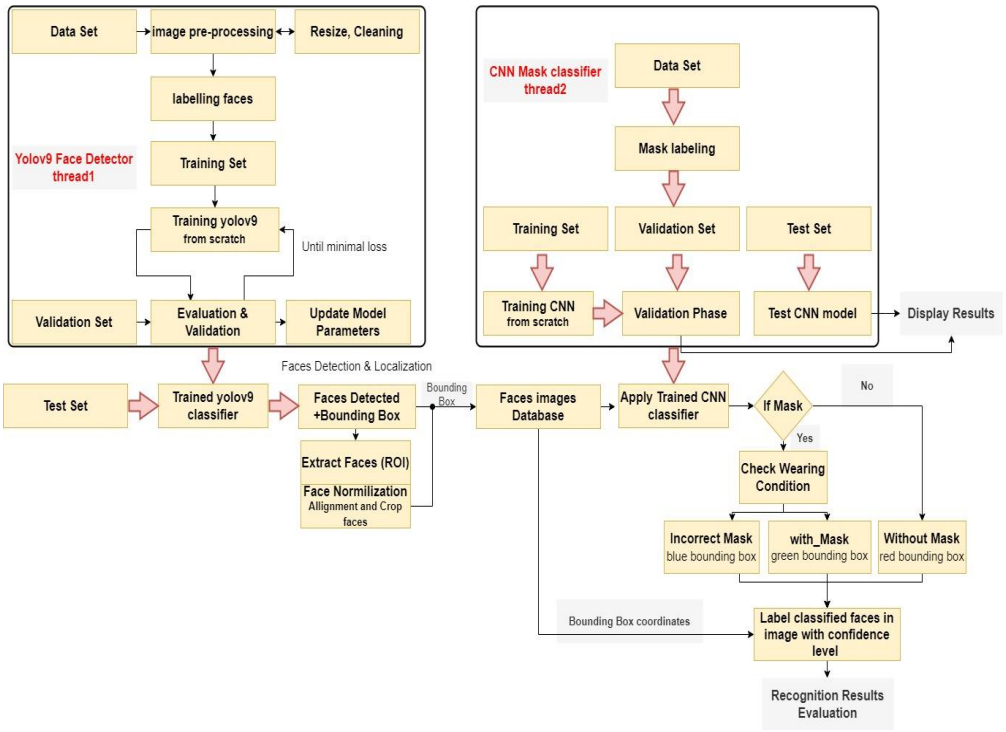



**Figure 3.** Proposed Methodology

### 6.1 YOLOv9 Classifier

In the proposed system, the Yolov9 face detector gets a good result as shown in the following steps:

**Step 1:** Initialize the yolov9 detector.

**Step 2:** Load the input image from the dataset.

**Step 3:** Image pre-processing by normalization and scaling.

**Step 4:** Detect the faces in the image using the yolov9 model.

**Step 5:** Check if the face was detected:

- Drawing a bounding box around the face. (x1, y1, width, height)
- Continue.

**Step 6:** Save the txt file containing the image name with a bounding box coordinate (x1, y1, width, height).

**Step 7:** Load the image and extract the faces from a given image.

The proposed system trains the YOLOv9 algorithm by using the MMD and FMD datasets. Then the algorithm will use this information to identify an input image and produce an output. We will use the GELAN-C architecture for 100 epochs with 16-batch sizes to train our model on the dataset. GELAN-C can be quickly trained. The inference times of GELAN-C are also rapid.

### 6.2 CNN Classifier

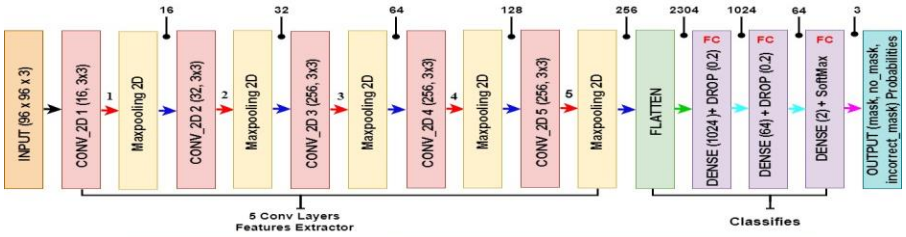
Face mask classification is the final and most crucial step in a face mask recognition system. In this step, available faces must be found, cropped, and aligned before feeding images to the proposed network. Then, the system uses the extracted facial information to classify the face and determine whether the individual is wearing a mask.

First, use a CNN configured with starting hyperparameters and input face photos processed via a series of steps. Figure 3 shows the exact architecture of the suggested networks. The faces in the CNN output are categorized into three classes: with masks, without masks, and incorrect masks. The suggested classifier model contains five convolutional layers with two fully connected layers and one dense softmax classifier as the final layer. The 3x3 kernel sizes used in each convolutional layer were positioned at the top. The system has 16, 32, 64, and 128 convolutional layers and 256 (3x3) kernel filters. As it passes through the convolutional layers, the image is transformed from a RGB image with three depth levels to (96x96x3). Each convolution layer's output is then routed to the max pooling layer, which uses a (2x2) window to accomplish pooling with a stride of 2.



DOI: https://doi.org/10.54216/JCIM.170109
Received: February 28, 2025 Revised: May 30, 2025 Accepted: July 20, 2025

The final stages of the convolutional layers consist of fully connected hidden layers with 2304 units, followed by fully connected dense layers with 1024 units. The network ends with a dense softmax result layer composed of 2 units. To reduce the issue of overfitting, a dropout rate of 0.2 is employed to deactivate layers randomly. Rectified Linear Unit (ReLU) is the activation function used by all-hidden layers and employs the soft-max function as the decision-maker in the final layer. The optimizer utilized is the Adam optimizer, compiled with loss as binary cross-entropy. Figure 4 below depicts the model after it has been updated. In the previous stage, the model received a input of 96,96,3, and at the end of the process, it classifies the faces present in the image.




| Symbol | Description | Parameters |
|---|---|---|
| 1 | Conv2D + ReLu | Kernel =16,(3x3), padding=same, strides=2/ 96*96*16 |
| 2 | Conv2D + ReLu | Kernel =32,(3x3), padding=same, strides=2/ 48*48*32 |
| 3 | Conv2D + ReLu | Kernel =64,(3x3), padding=same, strides=2/ 48*48*64 |
| 4 | Conv2D + ReLu | Kernel =128,(3x3), padding=same, strides=2/ 24*24*128 |
| 5 | Conv2D + ReLu | Kernel =256,(3x3), padding=same, strides=2/ 6*6*256 |
| | MaxPooling | Kernel =16,(3x3), padding=same, strides=2 |
| | FC layer | Flate(2304) |
| | FC layer | Dense1(1024),Dense2(64), Dropout(0.2) |
| | FC layer | Dense3(2), activation=softmax |

**Figure 4.** The proposed CNN Layers structure with Parameters

### 6.3 Hyperparameter Optimization

To achieve the ideal prediction outcomes from our model, we should aim to identify the optimal hyperparameter value set. An iterative procedure assesses the proposed model's performance on a validation dataset to prevent overfitting and choose the optimal hyperparameters based on the validation outcomes score. To optimize hyperparameters systematically, the suggested model uses Grid Search CV (automated Python modules) in the Scikit API package. The epochs =32, batch size=32, learning rate=1e-3, and dropout=0.2. The model in Figure 5 shows that hyperparameters will yield the best results and maximize their validation accuracy.

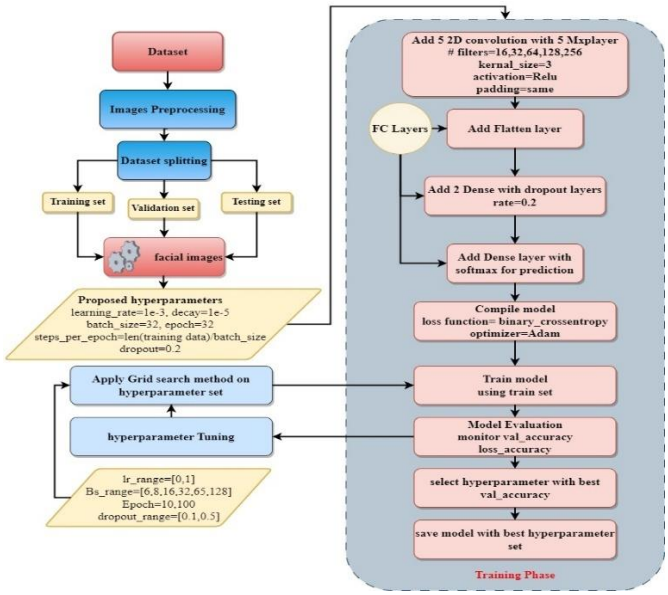



**Figure 5.** The proposed CNN (Training Phase) with Hyperparameter Settings

## 6. Experiment Result Analysis

Two phases comprise this work are Yolov9 face detection and CNN mask recognition. In the face detection phase, we will manually label the dataset using the tool LabelImg. After that, we separated the data into three sets: training, validation, and testing. Then, create two threads:



DOI: https://doi.org/10.54216/JCIM.170109
Received: February 28, 2025 Revised: May 30, 2025 Accepted: July 20, 2025

- One for training yolov9c to get a person's facial features (face detector).
- Then, at the same time, use another thread to train the proposed CNN using a facial dataset that is divided into three folders according to three classes (mask, without mask, incorrect mask).

After that, we used the test set to implement the trained detector for the facial detection stage. The detection findings are presented in the form of a bounding box (From the sides, it ends at the ends of the eyebrows, and from the bottom, it ends at the end of the chin) for the face area and saved into a YAML file. Next, the proposed model extracts the bounding faces to be processed in the next stage and specifies whether these faces were masked or not. The next step is to apply the top-performing model to a dataset that has never been seen before. Combining face detection and image classification models can be a powerful approach to enhance the capabilities of our face mask classifier. Here is a concise step to the proposed implementation:

1. Object Detection: Use the yolov9c object detection model to detect faces in input images and obtain bounding boxes with confidence scores for each detected face.
2. Extract Detected Regions: Crop and extract the detected face regions from the original image based on the bounding boxes for each detected face. This will give a set of cropped images, each containing one face. These cropped images will be the input for the facemask classification model. This will provide the class labels for each face.
3. Image Classification: Apply the mask classification model on each cropped image to determine the class of the detected face.
4. Combine Results: Overlay the classification results onto the original image at the location of each bounding box. Bounding boxes are drawn, and the original image is annotated with text labels corresponding to the classification results.

This approach should give a combined output with bounding boxes and classification labels, as illustrated in Figs. 6 and 7, which displays the results of our algorithm in three distinct scenarios involving the use of masks. When faces are detected, we use boundary boxes that include classification results.

(a) (b) (c)

**Figure 6.** The result of the proposed system (a) input images, (b) bounding box from yolov9c, (c) labelled images from CNN

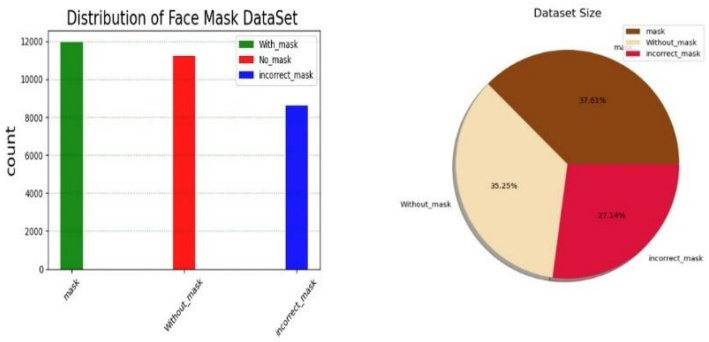



**Figure 7.** Distribution result of proposed CNN



DOI: https://doi.org/10.54216/JCIM.170109
Received: February 28, 2025 Revised: May 30, 2025 Accepted: July 20, 2025

During training, random initialization is used for all convolution layer parameters. Tables 1 and 2 show the hyperparameter settings of YOLOv9 and the proposed CNN, respectively.

**Table 1:** Hyperparameter settings of YOLOv9

| hyperparameter | value |
|---|---|
| epochs | 32 |
| optimizer | ADAM |
| worker | 8 |
| label smoothing | 0.0 |
| local rank | -1 |
| close mosaic | 15 |
| device | GPU |
| initial learning rate | 1e-5 |
| final One Cycle LR learning rate | 0.01 |
| learning rate decay | linear |
| momentum | 0.9 |
| warm-up epochs | 3.0 |
| weight decay | 0.0005 |
| warm-up bias learning rate | 0.1 |
| warmup initial momentum | 0.8 |
| box loss gain | 0.5 |
| class loss gain | 0.5 |
| DFL loss gain | 1.0 |
| object loss gain | 1.0 |
| object BCE Loss positive weight | 1.0 |
| anchor-multiple threshold | 4.0 |
| IoU training threshold | 0.2 |

**Table 2:** CNN Model Parameter

| hyperparameter | value |
|---|---|
| epochs | 35 |
| Batch size | 32 |
| optimizer | ADAM |
| initial learning rate | 1e-3 |
| decay | 1e-5 |
| Step_per_Epoch | Len (training data)/batch size |
| dropout | 0.2 |
| Kernal size | 3 |
| strides | 0.2 |



DOI: https://doi.org/10.54216/JCIM.170109
Received: February 28, 2025 Revised: May 30, 2025 Accepted: July 20, 2025

| | |
|---|---|
| Activation | RELU/SOFTMAX |
| padding | same |
| Loss function | Binary_crossentropy |
| Total params | 2,918,723 |
| Trainable params | 2,918,723 |
| Non-trainable params | 0 |

The confusion matrix in Figure 8 shows a more comprehensive understanding of the Yolov9 algorithm's behaviour at the last epoch. We note a considerable number of blue boxes in the Face class (TP 1550, FP 357, and FN 0). This means we have many positives from the validation stage. We can see how many class predictions were accurate along the major diagonal. It extends from the upper left corner to the lower right corner. The off diagonal, meaning the cases in which the model incorrectly predicted the positive class (false positive), will show the mistakes made. This results in zero instances of false negatives and true negatives, leading to empty cells in the matrix. It is a regular occurrence, especially with imbalanced datasets. The precision value at the last epoch is 94.6%. The recall value is 87.1%. mAP results are 92.74% and mAP50-95 67.42%. These performance results are summarized in Table 3, while the precision, recall, and mAP values for each epoch are illustrated in Figure 9. Furthermore, as indicated in Table 4, the suggestion YOLO v9c attained a 5.1% superior mAP value relative to YOLO v8. It outperformed other evaluated systems in the utilized proposed collecting data for facial identification. Thus, the designed system is suitable for detecting most faces in images, even if the faces are small.

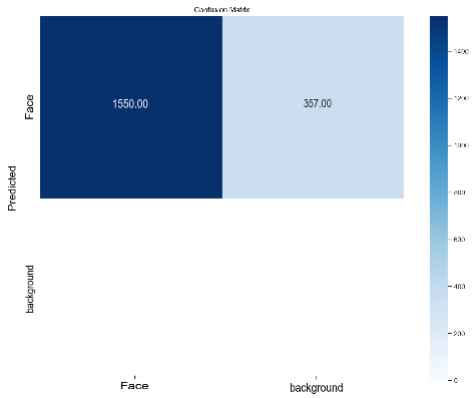



**Figure 8.** Confusion Matrix of face detection model

**Table 3:** Yolov9 Model Performance

| P | R | mAP50 | mAP50-95 | train/box_loss | train/cls_loss | train/dfl_loss |
|---|---|---|---|---|---|---|
| 0.94626 | 0.87051 | 0.92744 | 0.67422 | 1.1649 | 0.58596 | 1.38304 |

**Table 4:** Networks Performance Comparisons

| Network | mAP50 | mAP50-95 |
|---|---|---|
| Yolov5 | 0.7320 | 0.5581 |
| Yolov7 | 0.8330 | 0.5905 |
| Yolov8 | 0.8763 | 0.6447 |
| Yolov9c | 0.9274 | 0.6742 |



DOI: https://doi.org/10.54216/JCIM.170109
Received: February 28, 2025 Revised: May 30, 2025 Accepted: July 20, 2025

Figure 9 also displays the training and verification loss curves for the Bounding box regression, Confidence, and Classification loss functions. The last variation in the curves of all loss functions is small, indicating that the network stability is good. Figure 9 shows the performance results of mAP0.5:0.9, mAP0.5, precision, Recall, and loss in graphical form. YOLOv9 model graph performance increases until the last epoch. The graphs in Figure 9 show that the training procedure does not overfit or underfit, which shows that the Yolov9 model learning trained smoothly. The precision-recall curve is generated from the validation set after training is completed.

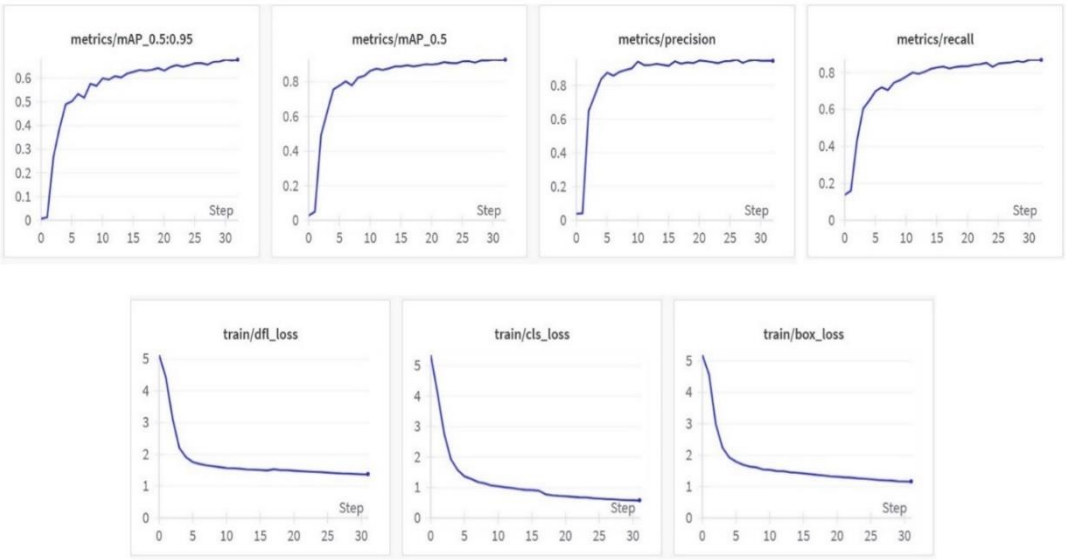



**Figure 9.** Yolov9 Model Performance Results

The model achieves an Average Precision (AP) of 92.74% among all classes over a Intersection over Union (IoU) criterion of 0.5, as shown via mAP@50. In this case, mAP50-95 takes the mean of the AP using a 0.05 step size and IoU levels between 0.5 and 0.95. The calculation used can be reduced as follows in Equation 1:

$$mAP = 1/n * \Sigma (AP @ IoU[i]) \quad (1)$$

In this context, n represents the total count of IoU thresholds, with IoU[i] values spanning from 0.5 to 0.95.

Figure 10 summarizes the classification performance. Values that lie off the diagonal denote misclassifications, while values that lie on the diagonal represent accurate predictions. Overall, the model is failing, as indicated by the confusion matrix, to detect 2.5% incorrect masks, 1% without masks, and 2% with masks (a few cases that fall into the opposite class). These classes have a few false positives since the detector is failing to detect true positive cases. For With_mask, 98% of samples are predicted correctly, whereas 2% (56 images) are confused with no_mask and an incorrect mask. For no_mask, 99% of samples are predicted correctly, whereas 1% (24 images) are confused with with_mask and an incorrect mask. For incorrect_mask, 97.5% of samples are predicted correctly, whereas 2.5% (50 images) are confused with with_mask and without_mask.

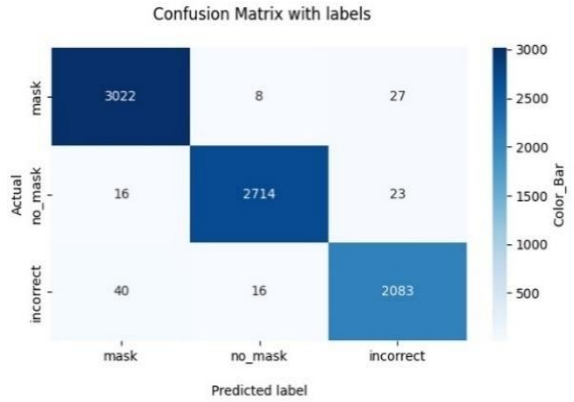



**Figure 10.** Classifier's Confusion Matrix



DOI: https://doi.org/10.54216/JCIM.170109
Received: February 28, 2025 Revised: May 30, 2025 Accepted: July 20, 2025

With the use of 35 epochs, the suggested classifier network was trained. As shown in Table 5, the training accuracy is equal to 98.35%, with a training loss of 3.05%. Meanwhile, the validation accuracy is recorded at 98.8%, with a validation loss of 2.34%, indicating that 98% of the true positive predictions were correct. There is no correlation between the acquired accuracy and any specific class. In spite of this, the developed classifier gets better at predicting actual value, which is remarkable considering the dataset's complexity and size, as well as the great architecture of the developed model. The low mean squared error of 0.042 and performance across all classes further support this claim.

**Table 5:** Performance Evaluation for Each Epoch

| # | loss | accuracy | val_loss | val_accuracy |
|---|---|---|---|---|
| 0 | 0.242837 | 0.849796 | 0.106598 | 0.956703 |
| 1 | 0.117394 | 0.943663 | 0.070937 | 0.963307 |
| 2 | 0.096080 | 0.952903 | 0.056349 | 0.969667 |
| 3 | 0.080211 | 0.961079 | 0.064839 | 0.972603 |
| . . . . . . | . . . . . . | . . . . . . | . . . . . . | . . . . . . |
| 30 | 0.033995 | 0.982993 | 0.022135 | 0.987769 |
| 31 | 0.031622 | 0.983483 | 0.015399 | 0.990704 |
| 32 | 0.029641 | 0.985773 | 0.035866 | 0.983855 |
| 33 | 0.034485 | 0.982175 | 0.033908 | 0.985568 |
| 34 | 0.030557 | 0.983565 | 0.023449 | 0.988014 |

It is crucial to examine the effectiveness of the suggested classification model and generalizability to new data after we have constructed and trained it. As can be seen in Table 6, the classification report is a widely utilized evaluation instrument for this specific purpose. The classification report is advantageous for managing imbalanced datasets or when the costs of misclassification vary among different classes. Table 6's classification report provides a more in-depth look at how well the model performed in each class. Precision, recall, f1-score, and support are some of the class-specific metrics included, along with general averages.

**Table 6:** Classification Report of the Proposed Model

| Classes | precision | recall | f1-score | support |
|---|---|---|---|---|
| Mask | 0.9818 | 0.9886 | 0.9852 | 3057 |
| Without_mask | 0.9912 | 0.9858 | 0.9885 | 2753 |
| Incorrect | 0.9766 | 0.9738 | 0.9752 | 2139 |
| accuracy | | | 0.9836 | 7949 |
| macro avg | 0.9832 | 0.9827 | 0.9830 | 7949 |
| weighted avg | 0.9837 | 0.9836 | 0.9836 | 7949 |

The precision for the "Mask" class is 0.9818, which means that "Mask" was predicted to be correct for 98% of the masked images (positive predictions with fewer false positive errors). This highlights the model's capacity to stop false positives.



DOI: https://doi.org/10.54216/JCIM.170109
Received: February 28, 2025 Revised: May 30, 2025 Accepted: July 20, 2025

In contrast, Recall represents the number of accurately recognized positive instances. Where it highlights the model's capacity that help in detect positive cases and reduce false negatives.

F1-score that allows for a balance between recall and precision. This number is very near to 1, which indicates that the model is entirely accurate in determining whether or not faces will be wearing masks. It is often used when there is an uneven class distribution.

Support describes the number of instances for every dataset class. It indicates the dataset's class distribution and helps identify whether the model's performance is constant across all of the distinct classes.

In this context, accuracy is a measure of performance over all classes and not relative to any one class in particular; it represents the overall accuracy of the model. It is 0.983565 for training and 0.983614 for testing. Also, a macro-average means that instead of determining overall accuracy, it takes a mean of each class's accuracy, which could be more instructive when dealing with classes that are not evenly distributed. In addition, the smooth training history plot confirms that there was no overfitting throughout training, as shown in Figure 11.

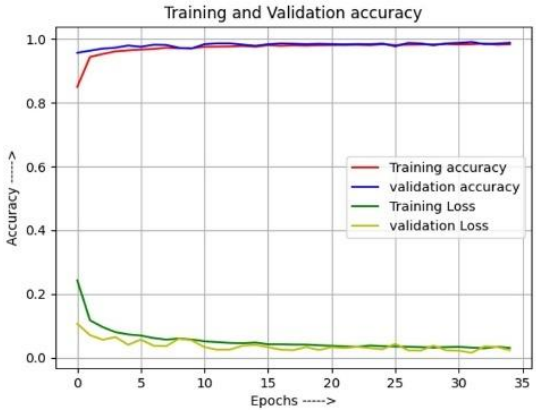



**Figure 11.** Training and Validation Curve of Proposed Classifier

The proposed algorithms need to verify effectiveness and validity to demonstrate their detection and classification performance. Comparative analyses were conducted based on accuracy and mAP by the researchers who had worked on the same data set, as shown in Table 7.

**Table 7:** Comparing the Suggested System's Accuracy and mAP to Other Algorithms

| Reference No. | Model | Description | Task | Result | Dataset used |
|---|---|---|---|---|---|
| [12] | YOLOv8n | Utilizes YOLOv8 and Test YOLOv8n, YOLOv8s, and YOLOv8m technique | Detection | mAP 61.9% | MMD and FMD |
| | YOLOv8s | | Detection | mAP 70.4% | |
| | YOLOv8m | | Detection | mAP 78.4 % Acc 99.1 % | |
| [20] | YOLOv5 | YOLOv5 detected two classes and was compared to other SOTA techniques. | Classification | Acc 92.0% | Dataset of 9000 images. The source was private. |
| [21] | YOLOv4 | The backbone CSPDarknet53 and neck PANet path-aggregation network make up YOLO v4. Three-layered neural networks form the head. | Classification | 99.5% | For training, 7320 images were utilized, whereas 2139 images were utilized for testing the model. |
| [22] | YOLOv5 | Jetson Nano TX1-implemented YOLO v5 model for facial mask identification and counting | Detection | mAP 70% | The Kaggle facemask detection dataset included 848 images |



DOI: https://doi.org/10.54216/JCIM.170109
Received: February 28, 2025 Revised: May 30, 2025 Accepted: July 20, 2025

| | | | | | |
|---|---|---|---|---|---|
| [23] | YOLOv7-CPCSDSA | This model, based on YOLOv7, incorporates Faster Net's partial convolution (P_Conv) and substitutes some of the original model's convolutions to create a Cat P_Conv (CPC) structure, SD module and SA mechanism. | Detection | mAP 88.4% | Datasets VOC_MASK and some user-generated mask images totaling 7,972 were found on the Baidu AI Studio website. |
| [24] | YOLOv4-SPP | Adding the spatial pyramid pooling (SPP) feature to tiny YOLO v4 | Detection | AP 86.31% mAP 64.31% | self-generated dataset for the detection of facial masks, which includes over 50,000 images |
| [25] | YOLO v5 | Optimizing the use of yolov5s variants with combination settings on image size values between 416 and 640 | Detection | mAP 86% | Kaggle facemask detection dataset |
| [26] | Improved YOLOv5 | Yolov5 + Coordinate Attention mechanism mask-wearing detection algorithm (YOLOv5-CBD) | Detection | mAP 96,7% | 3 classes, 3010 images, 2410 (training), and 600 (validation) |
| [27] | SMD-YOLO | a new variant of YOLOv4-tiny for small or medium-sized face mask detection | Detection | mAP 67,01% | public 3 classes dataset |
| [28] | YOLOv1 | upgraded tiny YOLOv4's backbone design with a modified-dense SPP network | Detection | mAP 52.40% | face mask detection dataset (FMD) |
| | YOLOv2 | | | mAP 55.34% | |
| | YOLOv3 | | | mAP 65.84% | |
| | EfficientNet-YOLOv4 | | | mAP 40.04% | |
| | tiny YOLOv4 | | | mAP 57.71% | |
| | ETL-YOLOv4 | | | mAP 67.64% | |
| [29] | YOLOv3 SPP | People were grouped using the YOLOv5m based on whether they wore masks or not. | Detection | mAP 65.2 % | FMD and MMD |
| | YOLOv3 | | | mAP 65.1% | |
| | YOLOv5s | | | mAP 66.2% | |
| | YOLOv5m | | | mAP 67.1% | |
| [30] | YOLOv4 | three attention-based modules—CA-Net, SE-Net, and convolutional block attention module—are introduced to various layers of the YOLOv4 model. | Detection | mAP 88.90% | 3,895 photos from the WIDER Face dataset plus 4,064 photos of masked faces from the MAFA dataset make up a total of 7959 photos. |
| | YOLOv4-CBAM-A | | | mAP 93.00% | |
| | YOLOv4-SENET-A | | | mAP 92.89% | |
| | YOLOv4-CANET-A | | | mAP 93.10% | |
| Proposed model | YOLOv9c | YOLOv9 for face detector | Detection | mAP 92.74% | Custom dataset |
| | CNN | CNN for face mask classifier | Classification | Acc 98.36% | |



DOI: https://doi.org/10.54216/JCIM.170109
Received: February 28, 2025 Revised: May 30, 2025 Accepted: July 20, 2025

## 7. Conclusion and Future Works

Within the scope of this study, we have presented and put into practice the proposed model for face mask status identification, which can determine whether a mask is worn or not, and whether it is worn correctly. The YOLOv9c (compact) has been utilized to achieve high-performance results for face detection. The YOLOv9c model enhances detection performance by incorporating mean IoU to determine the optimal anchor boxes and integrating Programmable Gradient Information (PGI) and reversible functions to ensure essential data retention, enhancing the efficiency and accuracy of the model. Comparative outcome showed the suggested framework for YOLOv9c was an effective model for detecting faces, achieving a higher mAP value and superior performance compared to other networks evaluated on the specified dataset used for detecting faces. Thus, the suggested network is suitable for most faces in images to detect them, even if the faces are small and in crowded scenes.

The proposed CNN classifies face area into three classes: mask, without mask, and incorrect mask correctly and with much higher accuracy, indicating that the combination of YOLOv9c with CNN meets the objectives for overall performance. The proposed methodology worked substantially well on the used dataset for three-class classification, which uses a few input dimensions, compared with studies in the literature that used diverse datasets, and it is suitable for practical settings in the world, like educational institutions, shopping malls, and military applications.

In the future, we plan to incorporate image preprocessing into the process, create a unique architecture to test on other datasets, and extend it to more object detection tasks. In addition, in future work, we intend to recognize masked faces in images and videos using YOLOv10 deep learning models, reduce the data imbalance using a large dataset, and add slightly distorted photos. We will also explore the YOLOv10 pose estimation. Furthermore, we intend to establish a lightweight network with improved efficiency by replacing Yolov9's Backbone network with MobileNetV3, conforming to the lightweight specifications for identifying targets in embedded or mobile equipment.

## References

[1] B. Rezazadeh, P. Asghari, and A. M. Rahmani, "Computer-aided methods for combating COVID-19 in prevention, detection, and service provision approaches," *Neural Comput. & Applic.*, vol. 35, pp. 14739–14778, 2023, doi: https://doi.org/10.1007/s00521-023-08612-y.

[2] C. Bisogni, A. Castiglione, S. Hossain, F. Narducci, and S. Umer, "Impact of Deep Learning Approaches on Facial Expression Recognition in Healthcare Industries," *IEEE Trans. Ind. Informatics*, vol. 18, pp. 5619–5627, 2022, doi: https://doi.org/10.1109/TII.2022.3141400.

[3] N. Ullah, A. Javed, M. Ali Ghazanfar, A. Alsufyani, and S. Bourouis, "A novel Deep Mask Net model for face mask detection and masked facial recognition," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 10, pp. 9905–9914, 2022, doi: https://doi.org/10.1016/j.jksuci.2021.12.017.

[4] J. Li, Z. Xu, L. Fu, X. Zhou, and H. Yu, "Domain adaptation from daytime to nighttime: A situation-sensitive vehicle detection and traffic flow parameter estimation framework," *Transp. Res. Part C Emerg. Technol.*, vol. 124, p. 102946, 2021, doi: https://doi.org/10.1016/j.trc.2020.102946.

[5] J. Zhu, G. Zhang, S. Zhou, and K. Li, "Relation-aware Siamese region proposal network for visual object tracking," *Multimed. Tools Appl.*, vol. 80, no. 10, pp. 15469-15485, 2021, doi: https://doi.org/10.1007/s11042-021-10574-z.

[6] M. Y. Muhamad, A. Rozniza, and S. H. Muhammad, "Comparison of Faster R-CNN and YOLOv5 for Overlapping Objects Recognition," *Baghdad Sci. J.*, vol. 20, no. 3, p. 0893, 2023, doi: https://doi.org/10.21123/bsj.2022.7243.

[7] L. Wang, Y. Shoulin, H. Alyami, A. A. Laghari, M. Rashid, J. Almotiri, H. J. Alyamani, and F. Alturise, "A novel deep learning-based single shot multibox detector model for object detection in optical remote sensing images," *Geosci. Data J.*, vol. 11, no. 3, pp. 237-251, 2022, doi: https://doi.org/10.1002/gdj3.162.

[8] H.S. Amit, M.H. Siti Z., S. Hussein, and K. Nurulaqilla, "YOLO: A Competitive Analysis of Modern Object Detection Algorithms for Road Defects Detection Using Drone Images," *Baghdad Sci. J.*, vol. 21, no. 6, 2024, doi: https://doi.org/10.21123/bsj.2023.9027.

[9] N. Rachburee and W. Punlumjeak, "An assistive model of obstacle detection based on deep learning: YOLOv3 for visually impaired people," *Int. J. Electr. Comput. Eng.*, vol. 11, no. 4, p. 3434, 2021, doi: https://doi.org/10.11591/ijece.v11i4.pp3434-3442.

[10] X. Gao, M. Nguyen, and W. Q. Yan, "Human Face Mask Detection Using YOLOv7+CBAM in Deep Learning," in *Handbook of Research on AI and ML for Intelligent Machines and Systems*, IGI Global, 2023, pp. 94-106, doi: https://doi.org/10.4018/978-1-6684-9999-3.ch005.



DOI: https://doi.org/10.54216/JCIM.170109
Received: February 28, 2025 Revised: May 30, 2025 Accepted: July 20, 2025

[11] B. Lin and M. Hou, "Face Mask Detection Based on Improved YOLOv8," *J. Electr. Syst.*, vol. 20, no. 3, pp. 365-375, 2024, doi: https://doi.org/10.52783/jes.2859.

[12] C. Dewi, D. Manongga, Hendry, E. Mailoa, and K.D. Hartomo, "Deep Learning and YOLOv8 Utilized in an Accurate Face Mask Detection System," *Big Data Cogn. Comput.*, vol. 8, no. 1, p. 9, 2024, doi: https://doi.org/10.3390/bdcc8010009.

[13] M. Banerjee, R. Goyal, P. Gupta, and A. Tripathi, "Real-Time Face Recognition System with Enhanced Security Features Using Deep Learning," *Int. J. Exp. Res. Rev.*, vol. 32, pp. 131-144, 2023, doi: https://doi.org/10.52756/ijerr.2023.v32.011.

[14] P. Wu, H. Li, N. Zeng, and F. Li, "FMD-Yolo: An efficient face mask detection method for COVID-19 prevention and control in public," *Image Vis. Comput.*, vol. 117, p. 104341, 2022, doi: https://doi.org/10.1016/j.imavis.2021.104341.

[15] S. Tamang, B. Sen, A. Pradhan, K. Sharma, and V. K. Singh, "Enhancing COVID-19 Safety: Exploring YOLOv8 Object Detection for Accurate Face Mask Classification," *Int. J. Intell. Syst. Appl. Eng.*, vol. 11, no. 2, pp. 892–897, 2023, [Online]. Available: https://ijisae.org/index.php/IJISAE/article/view/2966.

[16] J. Yu and W. Zhang, "Face Mask Wearing Detection Algorithm Based on Improved YOLO-v4," *Sensors*, vol. 21, no. 9, p. 3263, 2021, doi: https://doi.org/10.3390/s21093263.

[17] S. Hraybi and M. Rizk, "Examining YOLO for real-time face-mask detection," in *4th Smart Cities Symposium (SCS 2021)*, Online Conference, Bahrain, 2021, pp. 571-575, doi: https://doi.org/10.1049/icp.2022.0402 .

[18] "Medical Mask Dataset (MMD)," Kaggle Inc., [Online]. Available: https://www.kaggle.com/vtech6/me medical-masks-dataset. [Accessed Jan. 11, 2024].

[19] "Face Mask Dataset (FMD)," Kaggle Inc., [Online]. Available: https://www.kaggle.com/andrewmvd/face-mask-detection. [Accessed Jan. 11, 2024].

[20] L. Guo, Q. Wang, W. Xue, and J. Guo, "Detection of Mask Wearing in Dim Light Based on Attention Mechanism," *J. Univ. Electron. Sci. Technol. China*, vol. 51, no. 1, pp. 123-129, 2022, doi: https://doi.org/10.12178/1001-0548.2021222.

[21] S. Abbasi, H. Abdi, and A. Ahmadi, "A Face-Mask Detection Approach based on YOLO Applied for a New Collected Dataset," in *26th International Computer Conference, Computer Society of Iran (CSICC)*, Tehran, Iran, 2021, pp. 1-6, doi: https://doi.org/10.1109/CSICC52343.2021.9420599.

[22] H. F. Al-Selwi, H. Nawaid, G. Hadhrami, A. Nur, and A. Azlan, "Face mask detection and counting using You Only Look Once algorithm with Jetson Nano and NVDIA giga texel shader extreme," *IAES Int. J. Artif. Intell. (IJ-AI)*, vol. 12, no. 3, pp. 1169-1177, 2023, doi: http://doi.org/10.11591/ijai.v12.i3.pp1169-1177.

[23] J. Wang, J. Wang, X. Zhang, and N. Yang, "A Mask-Wearing Detection Model in Complex Scenarios Based on YOLOv7-CPCSDSA," *Electronics*, vol. 12, no. 14, p. 3128, 2023, doi: https://doi.org/10.3390/electronics12143128.

[24] Kumar, A. Kalia, A. Sharma, and M. Kaushal, "A hybrid tiny YOLO v4-SPP module-based improved face mask detection vision system," *J. Ambient Intell. Humaniz. Comput.*, vol. 14, pp. 6783–6796, 2023, doi: https://doi.org/10.1007/s12652-021-03541-x.

[25] L. Suroiyah, Y. Rahmawati, and R. Dijaya, "Facemask Detection Using Yolo V5," *J. Tek. Inform. (JUTIF)*, vol. 4, no. 6, pp. 1277-1286, 2023, doi: https://doi.org/10.52436/1.jutif.2023.4.6.1043.

[26] S. Guo, L. Li, T. Guo, Y. Cao, and Y. Li, "Research on Mask-Wearing Detection Algorithm Based on Improved YOLOv5," *Sensors*, vol. 22, no. 13, p. 4933, 2022, doi: https://doi.org/10.3390/s22134933.

[27] Z. Han, H. Huang, Q. Fan, L. Yiting, L. Yuqin, and X. Chen, "SMD-YOLO: An efficient and lightweight detection method for mask wearing status during the COVID-19 pandemic," *Comput. Methods Programs Biomed.*, vol. 221, p. 106888, 2022, doi: https://doi.org/10.1016/j.cmpb.2022.106888.

[28] Kumar, A. Kalia, and A. Kalia, "ETL-YOLO v4: A face mask detection algorithm in era of COVID-19 pandemic," *Optik*, vol. 259, p. 169051, 2022, doi: https://doi.org/10.1016/j.ijleo.2022.169051.

[29] Dewi and H.J. Christanto, "Automatic Medical Face Mask Recognition for COVID-19 Mitigation: Utilizing YOLO V5 Object Detection," *Revue d'Intelligence Artificielle*, vol. 37, no. 3, pp. 627-638, 2023, doi: https://doi.org/10.18280/ria.370312.

[30] G. Zhao, S. Zou, and H. Wu, "Improved Algorithm for Face Mask Detection Based on YOLO-v4," *Int. J. Comput. Intell. Syst.*, vol. 16, p. 104, 2023, doi: https://doi.org/10.1007/s44196-023-00286-7.



DOI: https://doi.org/10.54216/JCIM.170109
Received: February 28, 2025 Revised: May 30, 2025 Accepted: July 20, 2025