YOLOv8 Assisted Face Mask Detection

NAGA SARANYA HARINI K

Team Members

SHAHITHYA N, RAM KUMAR P, PRADEEP P

Student, Department of Computer Science with Artificial Intelligence and Data Science,KPR College of Arts, Science and Research, Coimbatore, India.

E-mail: nagasaranyaharini6@gmail.com

Abstract

The rapid spread of infectious diseases has emphasized the importance of wearing face masks to reduce the transmission of viruses and protect public health. However, manually monitoring whether people wear face masks in crowded places in difficult, time –consuming and prone to human error. To address this problem, this project proposes an automated Face Mask Detection System using the YOLOv8 (You Only Look Once Version 8) deep learning algorithm. The system is trained on a Kaggle face mask detection dataset to accurately identify and classify people as “Mask” or “No Mask” in real time. This project is designed for use in hospitals, educational institutions, airports, railway stations, shopping malls, offices, industries and other public places where continuous monitoring of mask compliance is required. It assists security personnel, healthcare workers and authorities in ensuring that safety protocols are followed efficiently without the need for constant manual supervision.The trained model achieved a Precision of 87.4%, Recall of 82.9% and mAP@50 of 86.7%, demonstrating reliable performance in detecting face masks. The proposed system offers several benefits, including improved public safety, reduced human effort, faster and more accurate monitoring, cost-effective surveillance and the ability to integrate with CCTV cameras for real-time detection. Furthermore, it can be expanded to detect other personal protective equipment such as helmets and safety vests, making it useful in healthcare, industrial and public safety applications. Overall, this project demonstrates how artificial intelligence, computer vision and deep learning can provide an efficient, accurate and scalable solution for automated face mask detection, contributing to safer environments and supporting the development of smart surveillance systems.

Keywords

Face Mask Recognition, Real-Time Detection, Convolutional Neural Networks (CNN),Image Processing, Public Health Monitoring, Surveillance System.

1. Introduction

Artificial Intelligence (AI) has revolutionized numerous fields by enabling computers to perform tasks that traditionally required human intelligence. One of the most significant applications of AI is computer vision, which allows machines to analyze, interpret and understand visual information from images and videos. Computer vision has become an essential technology in various sectors including healthcare, security, transportation, retail and industrial automation. Among its many applications, face mask detection has gained considerable importance in recent years due to the global spread of infectious diseases such as COVID-19. Wearing a face mask is one of the most effective preventive measures to reduce the transmission of airborne diseases. However, monitoring mask compliance manually in crowded public places is challenging, time consuming and error-prone task. To overcome these limitations, automated face mask detection systems have been developed using deep learning and object detection techniques. These systems can accurately detect whether a person is wearing a face mask or not by analyzing real-time images or video streams. The integration of artificial intelligence with computer vision has significantly improved the efficiency, accuracy and speed of surveillance systems, making them suitable for use in hospitals, educational institutions, airports, railway stations, shopping malls, offices and other public safety and health regulations must be maintained.

The Face Mask Detection using YOLOv8 project focuses on developing an intelligent and automated system capable of detecting face masks in real time using the latest YOLOv8 object detection algorithm. YOLOv8 is a state-of-the-art deep learning model known for its high detaction accuracy, fast processing speed and ability to identify multiple objects within a objects within a single image. In this project, a publicly available face mask detection dataset obtained from Kaggle was used to train the model using Google with GPU support. The model was trained for 120 epochs, enabling it to learn the distinguishing features of masked and unmasked faces effectively. After training, the model demonstrated strong performance in accurately classifying individuals as “Mask” or “No Mask” under different conditions. The proposed system can assist security personnel, healthcare professionals, educational institutions, government authorities and private organizations in enforcing safety measures efficiently without continuous manual supervision. Furthermore, the system offers benefits such as reduced monitoring time, improved public safety, cost-effective surveillance and easy integration with CCTV cameras for real-time monitoring. This project demonstrates the practical application of artificial intelligence, deep learning and computer vision in solving real-world problems and contributes to the development of smart surveillance systems for safer and healthier public environments.

2. Related Works

Table 1. Comparative summary of existing approaches for Face Mask detection

Author	Method	Accuracy	Limitations
Jiang et al. (2021)	Improved YOLOv3-Slim	92.50%	Performance can be affected by small face regions, lighting variations, and complex backgrounds.
Ding, Li & Yastremsky (2021)	YOLO-based object detection	89%	Requires a larger bounding-box-labelled dataset for improved generalization.
Kumar et al. (2021)	Improved Tiny YOLOv4-SPP	64.31%	Detection of small mask regions remains challenging.
ETL-YOLOv4 study (2022)	Improved ETL-YOLOv4	67.64%	Performance can vary with different datasets and real-world conditions.
YOLOv5 study (2024)	YOLOv5	88.90%	Performance can be affected by different environments and image conditions.

Previous studies show that deep learning techniques such as CNN, YOLOv3, YOLOv4, YOLOv5, and YOLOv8 are effective for face mask detection. Researchers have improved detection accuracy and speed using lightweight networks, attention mechanisms, and feature-enhancement methods. However, existing systems face limitations such as poor lighting, face occlusion, crowded environments, small faces, improper mask wearing, and dataset dependency. Recent YOLOv8-based approaches provide better real-time detection performance and accuracy. These studies motivated the development of the proposed YOLOv8 Face Mask Detection System, which focuses on accurately identifying mask and no-mask faces while reducing manual monitoring and supporting efficient real-time surveillance.

3. Proposed Work – Methodology

The proposed methodology of the face mask detection system is illustrated in Figure 1. It begins with the collection of face images, followed by data pre-processing and annotation. The annotated dataset is then used to train the YOLOv8 model. The trained model is used for real-time detection and classification of faces into Mask and No Mask classes. Finally, the performance of the model is evaluated using standard evaluation metrics.

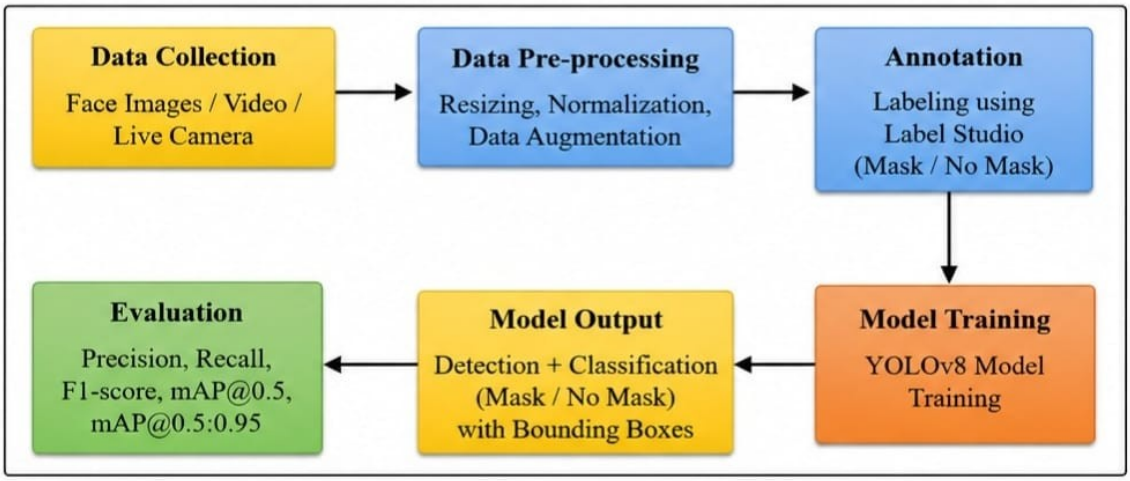

Figure 1. Workflow of the proposed system

Images are annotated using Label Studio by labeling the face region and class categories (Mask / No Mask). This annotated dataset is then used to train the YOLOv8 model to learn the features of both classes. The trained model is used to detect faces in new images or video frames and classify them into Mask or No Mask. The performance of the model is evaluated using Precision, Recall, F1-score, and mean Average Precision (mAP) metrics.

3.1 Data Collection

For this study, a face mask detection dataset was obtained from the Kaggle platform. The dataset contains images of people with different face-mask conditions and is organized into two classes: **Mask** and **No Mask**. The images represent different face orientations, sizes, backgrounds, and lighting conditions, providing varied samples for training the object detection model. The dataset was organized into training and validation directories along with the corresponding YOLO-format annotation files. In the present implementation, **1,161 images were used for training and 290 images were used for validation**. The two classes enable the YOLOv8 model to learn

the visual characteristics of masked and unmasked faces and distinguish between them during detection.

Table 2. Total number of images per class

Class	Total Images
Mask	230
No mask	116

3.2 Data Pre-processing

Prior to model training, the collected face images were pre-processed and organized to make them suitable for the YOLOv8 object detection algorithm. The dataset was divided into training and validation sets, and each image was associated with its corresponding YOLO-format annotation file. The annotation files contain the class identifier and the normalized bounding-box coordinates of the detected face regions. During training, the images were processed with an input size of **640 × 640 pixels**. Image resizing, pixel scaling, bounding-box normalization, and data augmentation were applied to improve the consistency and generalization capability of the model. The preprocessing operations help the YOLOv8 model learn the visual characteristics of masked and unmasked faces under different conditions.

3.2.1 Image Resizing

The original images may have different dimensions. To provide a consistent input size to the YOLOv8 model, the images were resized to 640 × 640 pixels during training. The resizing operation can be represented as:

$$I_{resized}=Resize(I,Wt,Ht)$$

where I represents the original image, $I_{resized}$ represents the resized image, and Wt and Ht represent the target width and height. In this work:

$$W_t=640 \qquad H_t=640$$

Therefore,

$$I_{resized}=Resize(I,640,640)$$

This ensures that the input images have a consistent spatial resolution during model training.

3.2.2 Pixel Value Scaling

Digital images generally contain pixel intensity values in the range of 0 to 255. For numerical processing, pixel values can be represented on a normalized scale between 0 and 1. The normalization operation is expressed as:

$$I_{norm}(x,y)=I_{resized}(x,y)/255$$

where $I_{resized}(x,y)$ represents the pixel intensity at the position (x,y), and $I_{norm}(x,y)$ represents the corresponding normalized value.

Thus,

$$0 \leq I_{norm}(x,y) \leq 1$$

This scaling provides a consistent numerical range for image processing and neural-network computation.

3.2.3 Bounding Box Normalization

The YOLO annotation format represents each detected object using five values:

$$(Class,xc,yc,w,h)$$

where Class represents the class identifier, xc and yc represent the center coordinates of the bounding box, and w and h represent its width and height.

The pixel coordinates are normalized according to the image dimensions. The normalized center coordinates are calculated as:

$$x_c=xpixel/W \qquad y_c=ypixel/H$$

The normalized bounding-box dimensions are calculated as:

$$w=wpixel/W \qquad h=hpixel/H$$

where W represents the image width and H represents the image height.

For an image of size 640 × 640 pixels:

W=640 H=640

Therefore,

x_c=xpixel/640 w=wpixel/640

y_c=ypixel/640 h=hpixel/640

This normalization makes the bounding-box representation independent of the original image dimensions.

3.2.4 Bounding Box Representation

If the bounding box is represented by its left, top, right, and bottom coordinates (xmin,ymin ,xmax,ymax), the center and dimensions can be calculated as:

x_c=xmin+xmax/2

y_c=ymin+ymax/2

$w=x_{max}-x_{min}$

$h=y_{max}-y_{min}$

The resulting values are then normalized using the image width and height:

x_c'=xc/W y_c'=yc/H

w′=w/W h′=h/H

The final YOLO annotation can therefore be represented as:

(Class,x_c',y_c',w′,h′)

3.2.5 Data Augmentation

To improve the ability of the model to generalize to unseen images, YOLOv8 applies data augmentation during training. In the implemented training configuration, augmentation included **mosaic augmentation, horizontal flipping, scaling, translation, HSV-based colour augmentation, and RandAugment**. These transformations generate variations of the training images while preserving the important characteristics of the face-mask classes.

Horizontal Flipping

For an image with normalized horizontal coordinate x, horizontal flipping can be represented as:

$$x' = 1 - x$$

The vertical coordinate remains unchanged:

$$y' = y$$

This allows the model to learn from faces appearing in different orientations.

Scaling

Scaling changes the size of the image or detected object by a scale factor s:

$$x' = sx \; y' = sy \; w' = sw \; h' = sh$$

Scaling helps the model detect faces appearing at different sizes and distances.

Translation

Image translation shifts an object horizontally and vertically:

$$x' = x + t_x$$

$$y' = y + t_y$$

where t_x and t_y represent horizontal and vertical translation values.

Translation helps the model learn to detect faces at different positions within an image.

HSV Colour Augmentation

Colour variations can be introduced by modifying the **Hue (H), Saturation (S), and Value (V)** components of an image. In general, the transformed image can be represented as:

$$I'_{HSV} = \text{Augment}(I_{HSV}, \Delta H, \Delta S, \Delta V)$$

where ΔH, ΔS, and ΔV represent changes applied to the hue, saturation, and brightness components.

This helps the model handle differences in illumination and colour conditions.

3.2.6 Mosaic Augmentation

Mosaic augmentation combines multiple training images into a single composite image. Conceptually, the resulting image can be represented as:

$$I_{mosaic}=Combine(I_1,I_2,I_3,I_4)$$

where I_1,I_2,I_3,I_4 represent four different training images. The corresponding object annotations are also transformed according to their new positions in the combined image.

Mosaic augmentation exposes the model to multiple objects, scales, and backgrounds within a single training sample, improving its ability to detect faces in different environments.

3.2.7 Class Encoding

The dataset contains two detection classes:

$$C=\{0,1\}$$

where:

$$0=Mask$$

$$1=No\ Mask$$

The class identifier is stored as the first value in each YOLO annotation.

3.2.8 Final Pre-processed Dataset

After preprocessing and augmentation, the images and corresponding annotations were supplied to the YOLOv8 training pipeline. The overall preprocessing sequence can be summarized as:

Original Image → Resize → Pixel Scaling → Bounding Box Normalization → Data Augmentation → YOLOv8 Training

The pre-processing stage therefore ensures that the input data are consistently represented and that the model receives sufficient variation to learn the distinguishing features between **Mask** and **No Mask** classes.

3.4 Model Selection and Training

The YOLOv8 model architecture is selected for the development of the proposed face mask detection system. YOLOv8 is a state-of-the-art object detection model that provides a good balance between detection accuracy and computational efficiency. The YOLOv8 architecture consists of three major components: the **Backbone, Neck, and Detection Head**. The Backbone is responsible for extracting important visual features from the input face images through a series of convolutional operations and feature extraction layers. These features include facial regions and patterns that help distinguish between people wearing masks and those without masks.

The extracted feature maps are then passed to the **Neck**, which performs multi-scale feature fusion to improve the detection of faces at different sizes and positions. This is particularly useful for detecting multiple faces in a single image. The Detection Head processes the fused features and generates bounding boxes, class predictions, and confidence scores. In the proposed system, the model is trained to identify two classes: **Mask** and **No Mask**. The model is trained using the prepared and annotated dataset in Google Colab with GPU acceleration. During training, the model learns the visual characteristics of both classes by minimizing the localization, classification, and distribution focal losses. After training, the best-performing weights are saved and used for validation and prediction on unseen images. The performance of the trained YOLOv8 model is evaluated using standard object detection metrics such as **Precision, Recall, mAP@0.5, and mAP@0.5:0.95**.

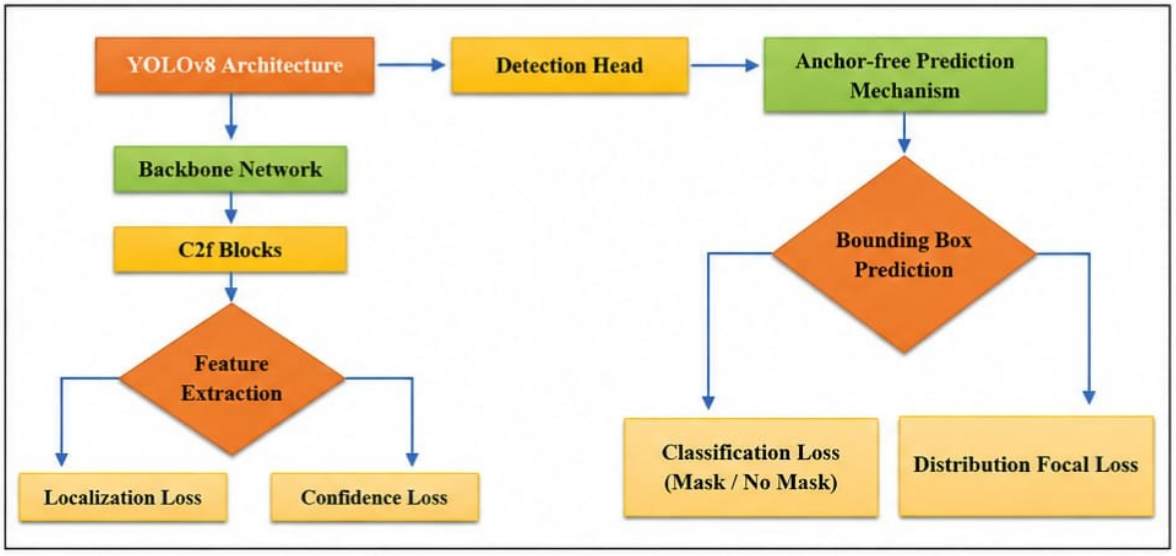

Figure 2. YOLOv8 architecture for face mask detection and classification

3.5 Model Evaluation

After training, the YOLOv8 model is evaluated using the validation dataset to measure its ability to correctly detect and classify faces into **Mask** and **No Mask** categories. The evaluation is performed using standard object-detection metrics such as **Intersection over Union (IoU), Precision, Recall, F1-score, mAP@0.5, and mAP@0.5:0.95**. These metrics help determine how accurately the model identifies the location and class of each face.

3.5.1 Intersection over Union (IoU)

IoU measures the overlap between the **predicted bounding box** and the **ground-truth bounding box**.

IoU=Area of Intersection/ Area of Union

A higher IoU indicates that the predicted bounding box is closer to the actual object location.

3.5.2 Precision

Precision measures how many of the faces predicted by the model are actually correct.

Precision=TP/TP+FP

Where:

- TP = True Positives
- FP = False Positives

Higher precision means the model produces fewer incorrect detections.

3.5.3 Recall

Recall measures how many of the actual faces in the dataset are successfully detected by the model.

Recall=TP/TP+FN

Where:

- TP = True Positives
- FN = False Negatives

Higher recall indicates that the model is able to detect more of the actual faces.

3.5.4 F1-Score

F1-score provides a balance between Precision and Recall.

F1=2xPrecision×Recall/Precision+Recal

A higher F1-score indicates a better balance between correctly identifying faces and avoiding incorrect detections.

3.5.5 Mean Average Precision at IoU 0.5 (mAP@0.5)

mAP@0.5 measures the average detection performance across the classes when a predicted bounding box is considered a correct detection at an IoU threshold of **0.5**.

mAP@0.5=N1i=1∑NAPi

Where:

- AP_i = Average Precision of class *i*
- N = Number of classes

For this project, the two classes are **Mask** and **No Mask**.

4. RESULTS AND DISCUSSION

4.1 System Configuration

The proposed face mask detection system was developed using Google Colab as the development environment with Python 3.12.13 and the Ultralytics YOLOv8 framework (version 8.4.114). PyTorch 2.11.0 was used as the deep learning library, with CUDA 12.8 enabled for GPU acceleration. The model was trained and evaluated using an NVIDIA Tesla T4 GPU with 14,913 MiB of GPU memory. The YOLOv8 model used in the project contains 3,006,038 parameters and requires approximately 8.1 GFLOPs for computation. The dataset consists of two classes, namely Mask and No Mask, and the validation set contains 290 images with 1,079 annotated instances. Google Drive was used for storing the dataset, trained model weights, and prediction results, while Google Colab provided the computational environment for model training, validation, and testing.

4.2 Performance Analysis

The performance of the proposed YOLOv8-based face mask detection system was evaluated using **Precision, Recall, mAP@0.5, and mAP@0.5:0.95**. The validation dataset contained **290 images and 1,079 annotated instances**. The model achieved an overall Precision of **87.4%**, Recall of **82.9%**, mAP@0.5 of **86.7%**, and mAP@0.5:0.95 of **50.8%**. The class-wise results show that the model performed better for the **Mask** class than the **No Mask** class.

Table 3. Class-wise performance metrics

Class	Precision	Recall	mAP@0.5	mAP@0.5:0.95
Mask	**91.8**	**89.9**	**93.7**	**60.1**
No Mask	**82.9**	**75.9**	**79.7**	**41.5**
Overall	**87.4**	**82.9**	**86.7**	**50.8**

The Mask class achieved the highest performance, with 91.8% Precision, 89.9% Recall, and 93.7% mAP@0.5, indicating that the model was highly effective in identifying people wearing masks. The No Mask class obtained 82.9% Precision, 75.9% Recall, and 79.7% mAP@0.5, showing comparatively lower detection performance. The difference may be related to variations in facial appearance, pose, background, and the distribution of samples between the two classes. The overall mAP@0.5 of 86.7% demonstrates good detection performance, while the 50.8% mAP@0.5:0.95 reflects the stricter requirement for accurate bounding-box localization across multiple IoU thresholds. Overall, the results indicate that the proposed YOLOv8 model can effectively detect and classify face-mask usage in the given validation dataset.

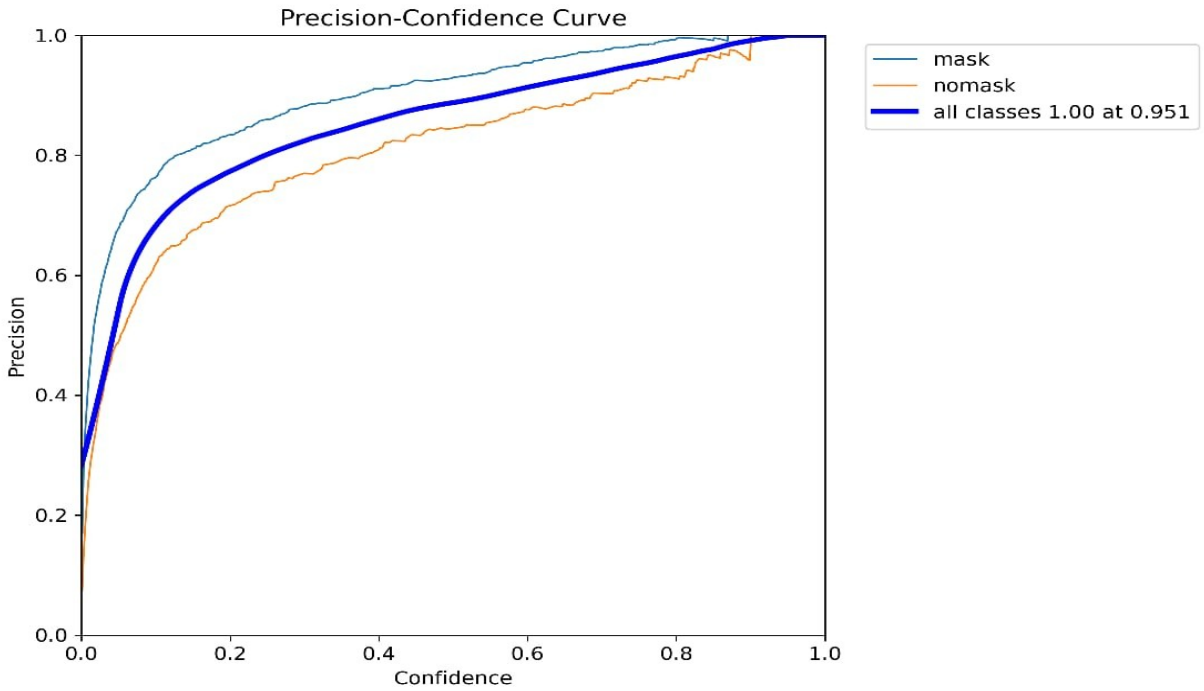

Figure 3. Precision-confidence curve

The Precision–Confidence Curve shows how the precision of the YOLOv8 face mask detection model changes as the confidence threshold is increased. The X-axis represents the confidence level, ranging from 0 to 1, while the Y-axis represents precision. The graph contains separate curves for the Mask, No Mask, and all classes. As the confidence threshold increases, the precision of the model generally improves because the model keeps only predictions in which it is more confident. The Mask class shows stronger performance than the No Mask class across most confidence levels, indicating that the model is more reliable when identifying faces wearing masks. The overall curve for all classes reaches a precision of approximately 1.00 at a confidence level of 0.951, showing that highly confident predictions are very accurate. At lower confidence levels, precision is comparatively lower because more uncertain detections are included. Therefore, the graph demonstrates that selecting an appropriate confidence threshold can improve the reliability of the face mask detection system.

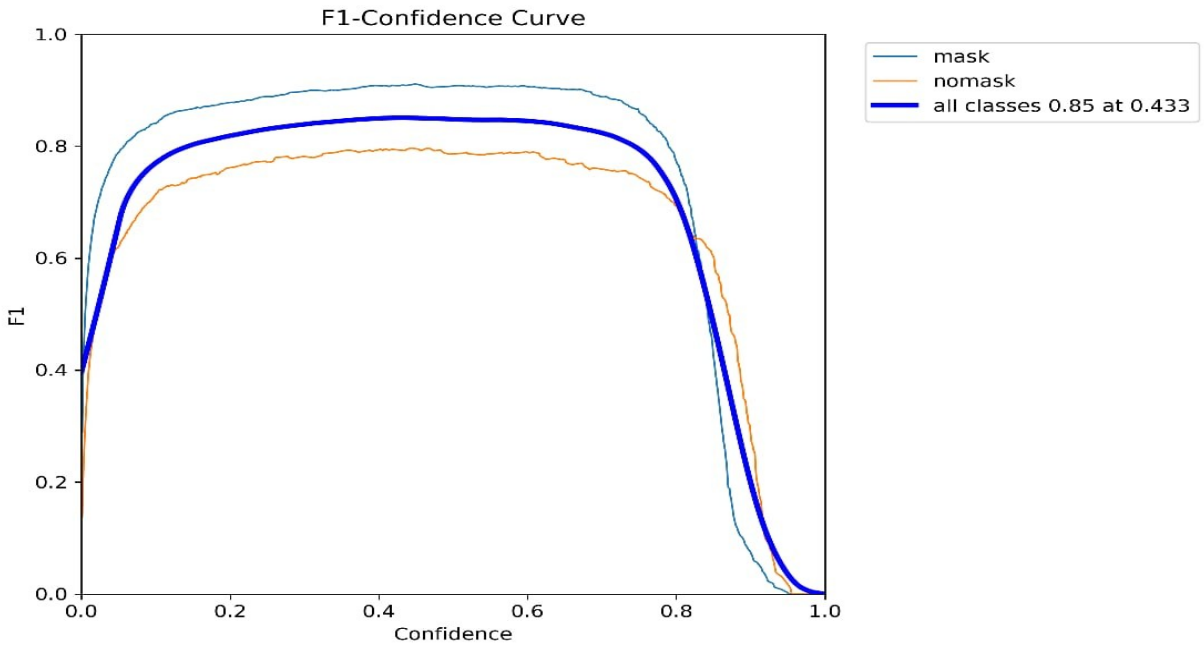

Figure 4. F1-confidence curve

The F1–Confidence Curve shows the relationship between the confidence threshold of the YOLOv8 model and its F1-score. The X-axis represents the confidence level, while the Y-axis represents the F1-score. The F1-score combines Precision and Recall and provides a balanced measure of the model's detection performance.

The Mask class achieves a higher F1-score than the No Mask class across most confidence levels, indicating better overall detection performance for the Mask category. The No Mask class reaches an F1-score of approximately 0.80 around the middle confidence range. The thick blue curve represents the performance of all classes, with the highest overall F1-score of approximately 0.85 at a confidence threshold of 0.433. This indicates that a confidence threshold of about 0.433 provides the best balance between precision and recall for the overall model. At very high confidence levels, particularly above approximately 0.8, the F1-score decreases sharply because the model becomes too selective and rejects more valid detections. Therefore, the graph indicates that a moderate confidence threshold provides the most balanced performance for the proposed face mask detection system.

The normalized confusion matrix **shows how accurately the YOLOv8 face mask detection model classifies the three categories:** mask, nomask, and background**. The** columns represent the true classes**, while the** rows represent the predicted classes**. Since the matrix is normalized, the values are expressed as proportions ranging from 0 to 1, where values closer to 1 indicate better classification performance.**

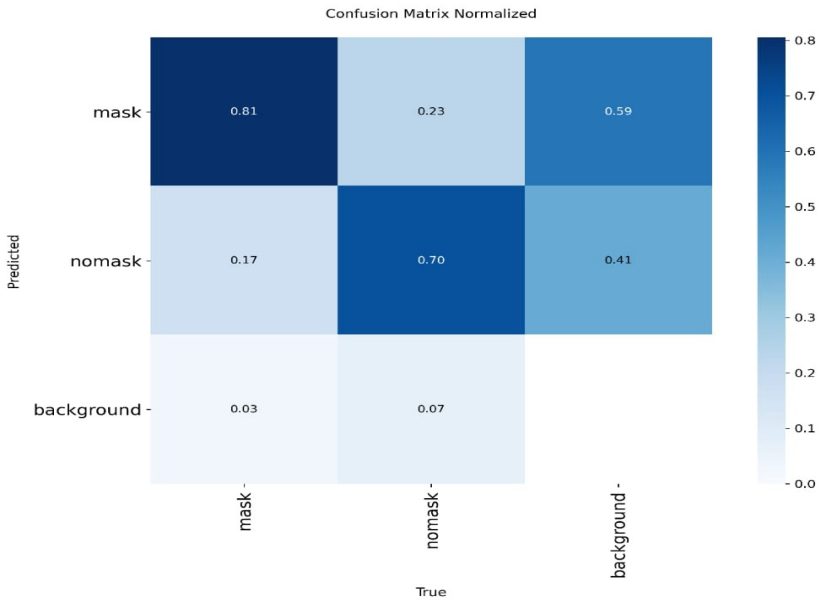

Figure 5. Normalized confusion matrix

For the **mask** class, **81%** of the actual mask samples are correctly predicted as mask, while **17%** are incorrectly predicted as nomask and approximately **3%** are classified as background. For the **nomask** class, **70%** of the actual nomask samples are correctly identified as nomask, whereas **23%** are incorrectly classified as mask and **7%** are classified as background. For the **background** class, approximately **59%** of the background samples are predicted as mask and **41%** are predicted as nomask, with almost no samples correctly identified as background. Overall, the matrix indicates that the model performs better in distinguishing **mask** and **nomask** faces than in identifying background regions. The diagonal values, particularly **0.81 for mask and 0.70 for nomask**, show the model's correct classification capability, while the off-diagonal values indicate the areas where misclassification occurs. This analysis helps identify the classes that require further improvement through additional training data, better annotations, or suitable data augmentation techniques.

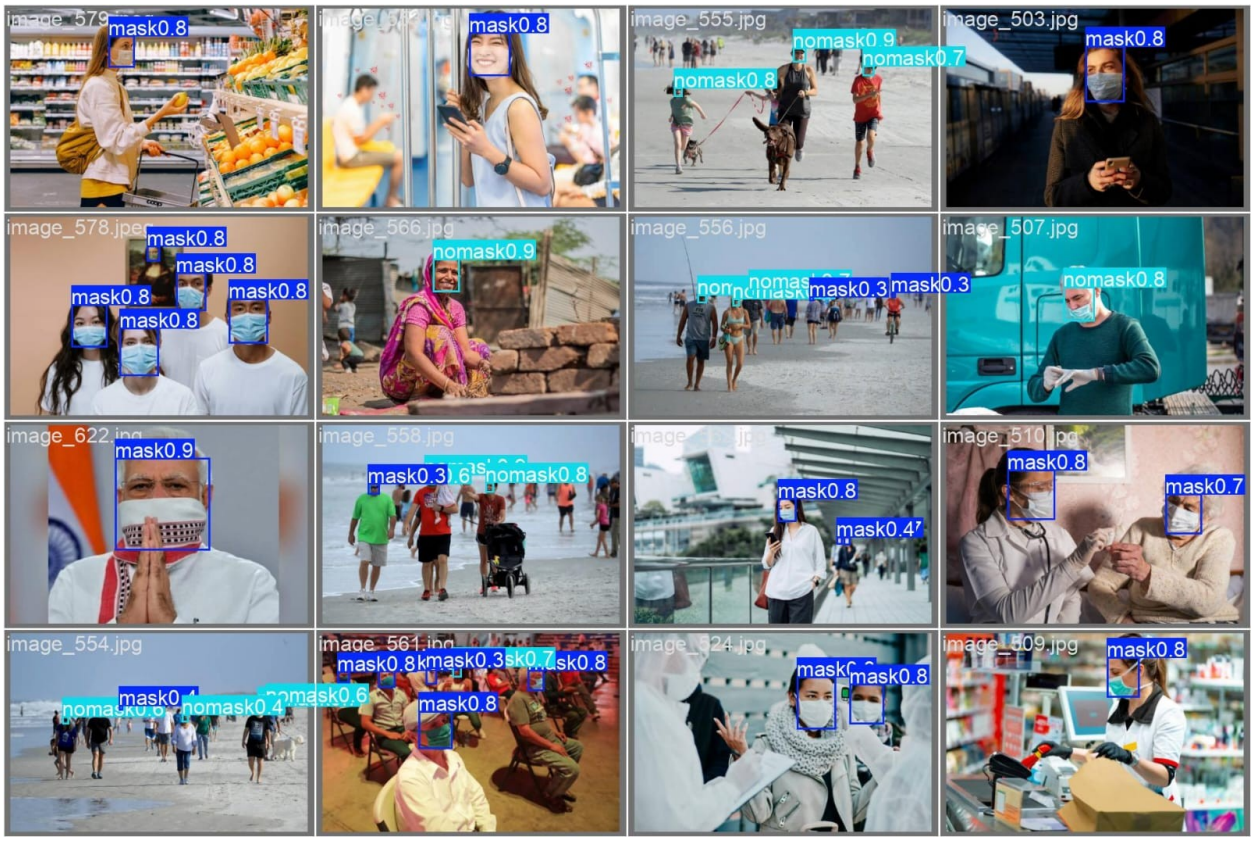

Figure 6. Sample prediction images

The following Figure 6 presents examples of predictions generated by the trained YOLOv8 model, where bounding boxes have been used to indicate the detected regions of the faces, along with their respective class label predictions. The YOLOv8 model is able to localize the face regions under different orientations, backgrounds, and image conditions, thereby indicating the effectiveness of the model in terms of robust face detection. The localization is then accompanied by **Mask** or **No Mask** class labels and their corresponding confidence scores, which demonstrates the ability of the proposed model to detect and classify faces accurately. The predictions shown in the figure include images containing single as well as multiple individuals, demonstrating the capability of the model to perform face mask detection in different real-world scenarios.

Conclusion

The proposed face mask detection system successfully demonstrates the application of the YOLOv8 deep learning model for automatic detection and classification of faces into **Mask** and **No Mask** categories. The system follows a systematic workflow consisting of data collection, data preprocessing, annotation, model training, prediction, and performance evaluation. The YOLOv8 model was trained using an annotated face mask dataset and was able to detect multiple faces in different environments, backgrounds, orientations, and image conditions. The prediction results show that the model can localize faces using bounding boxes and assign the appropriate Mask or No Mask label along with a confidence score. The performance of the trained model was evaluated using important metrics such as **Precision, Recall, F1-score, mAP@0.5, and mAP@0.5:0.95**, providing a comprehensive assessment of detection and classification performance. The Precision-Confidence and F1-Confidence curves indicate that the model achieves good prediction performance at suitable confidence thresholds. The normalized confusion matrix further demonstrates the model's ability to distinguish between the Mask and No Mask classes, although some misclassifications occur under challenging conditions. The prediction examples also show that YOLOv8 can identify multiple faces within a single image and provide confidence-based classifications. Overall, the proposed system provides a fast and automated approach for face mask detection and can be useful in environments such as educational institutions, hospitals, workplaces, public transportation areas, and other crowded locations where monitoring mask usage may be required. The system reduces the need for continuous manual monitoring and provides consistent detection results. With a larger and more diverse dataset, improved annotation quality, and further model optimization, the system can potentially achieve better accuracy and robustness in real-world applications. Thus, the study demonstrates that YOLOv8 is a suitable deep learning approach for developing an efficient and real-time face mask detection system.

References

1. Jiang, X., Gao, T., Zhu, Z., & Zhao, Y. (2021). *Real-Time Face Mask Detection Method Based on YOLOv3.* Electronics, 10(7), 837. The study proposed a YOLOv3-based real-time face-mask detection method for detecting mask-wearing conditions.

2. Liu, R., & Ren, Z. (2021). *Application of YOLO on Mask Detection Task.* The authors investigated YOLO for real-time face-mask detection and highlighted its advantage in processing speed compared with computationally heavier approaches.

3. Ding, Y., Li, Z., & Yastremsky, D. (2021). *Real-time Face Mask Detection in Video Data.* The proposed object-detection approach achieved 89% mAP@0.5 on real-world images while maintaining 52 FPS on video data.

4. Sapakova, S., & Yilibule, Y. (2022). *Deep Learning-Based Face Mask Detection Using YOLOv5 Model.* The study developed a YOLOv5-based system for detecting people without masks in images and live-camera environments.

5. Xu, S., Guo, Z., Liu, Y., Fan, J., & Liu, X. (2022). *An Improved Lightweight YOLOv5 Model Based on Attention Mechanism for Face Mask Detection.* The improved model achieved 95.2% mAP, while also improving inference speed compared with the original YOLOv5.

6. Guo, S., Li, L., Guo, T., Cao, Y., & Li, Y. (2022). *Research on Mask-Wearing Detection Algorithm Based on Improved YOLOv5.* The researchers proposed YOLOv5-CBD to improve mask-wearing detection under conditions such as occlusion, crowded scenes, and small targets.

7. Khoramdel, J., Hatami, S., & Sadedel, M. (2023). *Wearing Face Mask Detection Using Deep Learning Through COVID-19 Pandemic.* The researchers compared SSD, YOLOv4-tiny, and YOLOv4-tiny-3l. The best model achieved 85.31% mAP and 50.66 FPS.

8. Abunemreh, M. M., & Ibrahim, A. A. (2026). *Real-Time Face Mask Detection Using YOLOv8n with Performance Optimization.* The work specifically investigates a YOLOv8n-based binary face-mask detection system, making it particularly relevant to your YOLOv8 project.

9. Mostafa, S. A., Ravi, S., Zebari, D. A., Zebari, N. A., Mohammed, M. A., & Nedoma, J. (2024). *A YOLO-Based Deep Learning Model for Real-Time Face Mask Detection via Drone Surveillance in Public Spaces.* The proposed C-Mask system detected three categories—mask, incorrectly worn mask, and no mask—and reported 92.20% overall accuracy, 92.04% precision, 90.83% recall, and 89.95% F1-score.

10. Face-mask detection using deep learning and object-detection approaches. Recent research has continued to investigate YOLO-based approaches because they can simultaneously localize faces and classify mask-wearing status, making them suitable for real-time applications. Your proposed YOLOv8 system follows this direction while