International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 12-No.3 (2026) pp. 80-88
http://www.ijcesen.com

ISSN: 2149-9144

Research Article

Real-Time Face Mask Detection Using YOLOv8n with Performance Optimization

Mohammad Mohammad Abunemreh[1], Abdullahi Abdu Ibrahim[2]*

[1]Altinbaş Üniversity, Institute of Graduate Faculty, Information Technology Department, 34217, Istanbul-Turkiye
Email: 233721648@ogr.altinbas.edu.tr - **ORCID:** 0009-0008-0025-9019

[2]Altinbaş Üniversity, Institute of Graduate Faculty, Information Technology Department, 34217, Istanbul-Turkiye
* **Corresponding Author Email:** abdullahi.ibrahim@altinbas.edu.tr - **ORCID:** 0000-0002-5599-348X

Article Info:

DOI: 10.22399/ijcesen.5330
Received : 20 April 2026
Revised : 15 June 2026
Accepted : 18 June 2026

Keywords

Face mask detection
YOLOv8
Computer vision
Deep learning
Real-time inference

Abstract:

The COVID-19 pandemic has highlighted the practical limitations of manual monitoring in high-density population and emphasized the importance of complying with face mask rules in public places. The aim of this study is to present a full design, implementation, and systematic evaluation of a binary face mask identification system based on the YOLOv8n deep learning model. The dataset used for training was taken from Roboflow and labeled with two classes and 929 annotated instances. It includes 120 training shots and 128 validation images. The model was trained using the Ultralytics YOLOv8 framework on a cloud-based GPU environment for five epochs. The model was trained using this dataset. The trained model has attained a precision of 0.681, recall of 0.582, mean Average Precision at IoU threshold 0.5 (mAP50) of 0.655 and mAP50-95 of 0.490. Per-class analysis showed that the mask class achieved a mAP50 of 0.777, while the no-mask class achieved a mAP50 of 0.534. This asymmetry is attributed to the class imbalance in the training data. As per confusion matrix evaluation, 434 correct mask classifications and 216 appropriate no mask classifications are made out of total 929 validation cases. It was proven that near real-time inference can be done on standard GPU hardware at 154 milliseconds per image (~ 6.3 frames per second). In this work we provide a fully repeatable baseline, with the whole training pipeline running in 0.241 hours and using just publicly available technology. The findings are competitive to those of published systems including FMD-YOLO (66.4% VOC mAP) but with far lower implementation complexity. This shows that YOLOv8n, in conjunction with Roboflow and Ultralytics, provides an accessible and efficient pipeline for automated public health compliance monitoring.

1. Introduction

The global outbreak of the COVID-19 pandemic, declared by the World Health Organization in March 2020, triggered an unprecedented public health emergency affecting hundreds of millions of people worldwide [1]. Among the most widely adopted and evidence-based preventive measures was the mandatory use of face masks in public spaces, demonstrated to significantly reduce airborne transmission of respiratory pathogens. Despite extensive public awareness campaigns and legal mandates across many jurisdictions, ensuring consistent mask compliance in high-density environments, airports, hospitals, shopping centers, public transportation systems, and educational, Institutions remained a persistent operational challenge that could not be sustainably managed through conventional human supervision alone [2]. Since then, research related to detecting mask-wearing has not stopped, as it has been divided into two main categories: the two-category case (with a mask or without a mask) and the three-category case, which adds the case of wearing the mask incorrectly. This led researchers to turn to algorithms due to the difficulty of manual supervision over the vast amount of data required since then to solve the problem of verifying mask usage or not. Some studied the two-category case, while others studied the three-category case [2]. These limitations demonstrate a clear need for automated, scalable, and continuously operating compliance monitoring systems. Computer vision and deep learning provide the technical foundation for such systems, enabling

machines to analyze live video streams and generate compliance alerts without human intervention [3].
Object detection the simultaneous classification and spatial localization of objects within images or video frames has emerged as the most practical technical approach to automated face mask detection [4,5]. Among available detection frameworks, the YOLO (You Only Look Once) family of one-stage detectors has established itself as the preferred choice for real-time surveillance applications due to its single-pass inference architecture, which enables processing speeds compatible with live video analysis at practical frame rates [4]. Figure 1 illustrates the standard computer vision pipeline within which face mask detection systems operate.

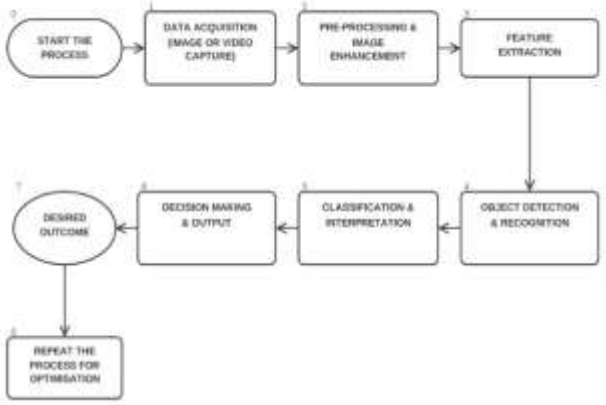

Figure 1. *Computer vision processing pipeline from raw data acquisition through preprocessing, feature extraction, detection, classification, and iterative deployment optimization.*

YOLOv8 is the latest and most architecturally sophisticated YOLO family member, produced by Ultralights in January 2023. In this system, three key innovations are introduced: an anchor-free detection head that removes the need for predefined anchor boxes, the C2f module that improves the flow of gradient through cross-stage partial connections, and a decoupled prediction head that separately optimizes the classification and bounding box regression objectives. These enhancements result in faster convergence and improved detection accuracy, especially in limited training data settings. This is a crucial attribute for the face mask detection job, since the labeled datasets are generally smaller than those used for general object detection benchmarks [6].
The YOLOv8 algorithm is considered one of the latest and most advanced models used for detecting objects and entities in real-time as required. Where some experiments are conducted on it using mask detector data to evaluate its suitability in classifying the three cases of mask-wearing. Through practical experiments, it has been demonstrated that it can be relied upon to classify the three cases of mask-wearing (wearing the mask, not wearing the mask, wearing the mask incorrectly). [7]. Any system or technological solution identifies the presence or absence of a mask through two stages. The first stage is identifying the location of a person's face in the captured images, which can be sourced from either photos or videos. The second stage is determining the presence of the mask and its condition (i.e., whether the person is wearing it correctly or not). YOLOv8s is another new type of YOLOv8 that has been used for the two categories (with mask or without mask), and through extensive experiments, it has achieved high and remarkable accuracy in both categories.[8]
existing evaluations have primarily focused on larger model variants (small, medium, large, or extra-large) trained on substantially larger labeled datasets. The YOLOv8n (nano) variant the lightest and fastest configuration has not been specifically characterized for binary face mask detection under minimal resource constraints. This gap is practically significant because the nano variant is the most suitable configuration for deployment on resource-constrained hardware including edge devices and embedded processors without dedicated GPU infrastructure.
This paper addresses this gap with four primary contributions: (1) a fully reproducible and documented binary face mask detection pipeline using exclusively publicly available tools, with all results derived from a single documented training session; (2) empirical characterization of YOLOv8n performance for this task under minimal resource constraints, establishing a practical lower bound for the YOLOv8 architecture family; (3) a per-class analysis quantifying the impact of training data class imbalance on no-mask detection performance with specific remediation recommendations; and (4) a systematic comparison with representative published baselines contextualizing the system within the state of the art.

2. Related Works

2.1 Face Mask Detection Systems

The issue of wearing masks has become extremely important since the COVID-19 pandemic, as the matter of recognizing individuals, their mask-wearing, and how to wear them correctly expanded the research during the pandemic of 2019-2020. In the early stages of detection at that time, the MAFA database was used along with LLE-CNN detectors, which showed that the standard face detection was limited to 50% at best when applied to faces wearing masks. A method was proposed to integrate

YOLOv2 with ResNet-50 to become specialized extractors and search engines, and it was clarified that leveraging pre-trained architectures can improve the accuracy that can be achieved on each component independently [9].

The SSDMNV2 model [10], developed using multiple detectors on the same image or frame (SSD [11] with MobileNetV2 [12]), achieved an accuracy exceeding 90%. This architecture is characterized by being lightweight and suitable for embedded deployment.

During the same period, the FMD-YOLO model was introduced, which took into account its distortion and rotation processes that help change the shape and geometry of the face during busy periods, achieving a score of over 65% on the VOC scale [13].

A new approach relying on YOLO, along with some additional custom data, was used, resulting in an accuracy exceeding 80%. [14]. Then YOLOv3 was applied to specialized data similar to mask detection to achieve high performance [15].

The latest development in models has led to the introduction of YOLOv8, which was trained on the FMD dataset along with a relatively large amount of training data compared to its predecessors, achieving the highest accuracy rate of around 95%. [7]. Then the system's capability was verified in some official conferences during the year 2024 [8].

2.2 YOLO Architecture Evolution

Detecting objects in images and frames is crucial for identifying mask-wearing. In 2016, the YOLO algorithm was introduced, which established a new model at the time capable of real-time object detection by employing an image segmentation mechanism and predicting each cell simultaneously through bounding boxes.

Sequential improvements were made to achieve better enhancements, and in the same context, the YOLOv3 model was introduced along with a feature map with three times the resolution to achieve significant accuracy and improvement in detecting small objects [16]. A large and comprehensive set of training improvements were integrated into the YOLOv4 model, the most important of which were the enhancement of Mosaic data, CSPDarknet53, and the PANet path neck, achieving a balance between accuracy and speed [17]. YOLOv7 was introduced, achieving a performance breakthrough that was considered groundbreaking at the time according to the MS COCO benchmark [18]. Table 1 summarizes the key innovations across YOLO versions.

Table 1. *Summary of key YOLO family versions, architectural contributions, and references.*

Version	Year	Key Contribution	Ref.
YOLOv1	2016	Unified regression detection; single-pass inference; S×S grid	[3]
YOLOv3	2018	Darknet-53; multi-scale predictions at 3 resolutions; residual connections	[16]
YOLOv4	2020	CSPDarknet53; PANet; SPP; Mosaic augmentation; CIOU loss	[17]
YOLOv7	2022	E-ELAN architecture; compound scaling; trainable bag-of-freebies	[18]
YOLOv8	2023	Anchor-free head; C2f module; decoupled prediction head; multi-task support	[6,19]

YOLOv8 introduces three primary improvements over its predecessors: the anchor-free detection head eliminates predefined anchor boxes, reducing design hyperparameters and enabling flexible predictions for objects of non-standard aspect ratios; the C2f module replaces the C3 module from YOLOv5, adding cross-stage partial connections through two bottleneck blocks for richer gradient pathways; and the decoupled head independently processes classification and regression in separate branches.

2.3. Transfer Learning and Data Augmentation in Detection

Some large datasets like ImageNet or MS COCO allow for the fine-tuning of certain pre-trained network models, as they were initially limited until transfer learning began to be used in some computer vision tasks. The initial training models are somewhat strong in extracting the established features, which makes them require fine-tuning to target only the specific category and task needed. Transfer learning has become extremely important in the process of face mask detection because training transfer models from scratch is difficult and rare, and the available pre-trained datasets are much smaller than those trained from scratch. Currently, training very deep architectures has become possible through skip connections with deep residual networks on the ResNet model. Which is considered one of the most important pillars in transfer learning [20].

The diversity in the data used for training is very important, as its significance becomes apparent when the training data is limited. Generalization through visual behavior can be improved through appropriate data enhancement. When it comes to

face detection using a mask, some improvements are appropriate, such as addressing the wide variations in mask colors through different product colors of the mask by applying some enhancements to the color space. And the robustness toward the face and the camera angle can be improved through some specific engineering enhancements [21].

During the optimization process with mosaics, at least four images with random scales were combined by creating synthetic training samples that were initially presented in the YOLOv4 model [17], and then reused in the YOLOv8 model [6]. This led to a significant expansion in the diversity of scale compositions and the context presented during training for the small training sets.

Recent evaluations of the YOLOv8 model have shown high performance, as it currently represents the best reference for real-time face detection through masks [7,8]. At the same time, the current evaluations of YOLOv8 use large models and data much more than necessary. This study proposes a new model for detecting dual face masks under certain data resource constraints and within a reproducible and documented experimental framework.

3. Materials and Methods

This study conducts a design and experimental work through which a new system for dual mask detection is developed and trained. The YOLOv8n model is trained for five consecutive epochs on a specific and limited dataset to detect dual face masks. Then, the results were compared with the previously published standards. The overall methodology pipeline is illustrated in Figure 2.

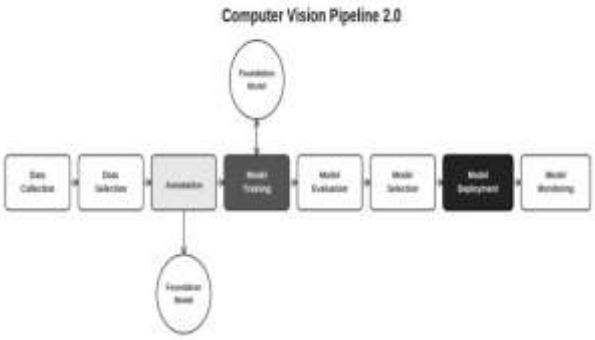

Figure 2. *Research methodology pipeline the five sequential stages progress from dataset acquisition and preprocessing through model training, performance evaluation, and iterative optimization.*

3.1. Dataset

The open database Roboflow provided the data used in this study [22]. The JPEG images that make up the data have bounding boxes that are separated into two categories: 0, which represents a human without a mask, and 1, which represents a person with a mask. Every coordinate has the label "0,1".

The features of the Roboflow platform perform a pre-split of the dataset into several individual groups used for the required training and tests. as summarized in Table 2.

Table 2. *Dataset composition and subset distribution for the face mask detection experiments.*

Subset	Images	Instances	Usage
Training	120	—	Model weight optimization
Validation	128	929	Epoch-level monitoring and final evaluation
Testing	Incl. in validation	—	Final quantitative assessment

3.2. Data Preprocessing and Augmentation

The standard input resolution for the YOLOv8 model was used by resizing the images to 640x640 pixels, and the pixel values were also adjusted to (1,0). The files were verified to meet the YOLO model format before starting the training. Some necessary improvements were made to the images, such as enhancing the mosaic, improving the color space, and changing some colors, saturation, and brightness. The number of images that were trained was approximately 120 images without adding any other classified data.

3.3. Model Architecture

The YOLOv8n (Nano) model was used, which is characterized by its lightweight structure and suitability for real-time inference. Feature maps with multiple scales are produced through the key components of the model, which include the use of the backbone with C2f units, in addition to some partial connections through the stages, resulting in the production of targeted hierarchical features. The scales enable the capture of spatial details at a relatively low level, and also allow for the capture of high-level semantic representations.

Spatial accuracy and hierarchical features are produced in the semantic content by applying the PANet path aggregation network, which integrates features through upsampling and lateral connections.

3.4. Training Configuration

Python was used in the Google Colab cloud environment with GPU acceleration to conduct training within the Ultralytics YOLOv8 library. Pre-trained models available from open sources were used, through which the YOLOv8n.pt model was configured. Transfer learning was utilized to compensate for the issue of the size of the datasets used and also to accelerate obtaining the required data in the desired time. Automatic optimizers were used by the framework. The full training configuration is presented in Table 3.

Table 3. *Complete training configuration as recorded from the actual training session log.*

Parameter	**Value**
Model	YOLOv8n (nano)
Initialization	Pre-trained YOLOv8n.pt (MS COCO)
Training Epochs	5
Image Size	640 × 640 pixels
Training Batch Size	8 (8 batches/epoch)
Optimizer	AdamW (auto-selected)
Initial Learning Rate	0.000119
Momentum	0.9
Weight Decay	0.0005
Training Images	120
Validation Images	128 (929 instances)
Number of Classes	2 (mask, no-mask)
Model Layers (fused)	73
Parameters	3,151,904
GFLOPs	8.7

3.5. Evaluation Metrics

A technique known as MS COCO assessment was utilized in order to evaluate the performance of the model. This metric, known as the Intersection over Union (IoU) metric, is used to determine whether a detection is a true positive. It is defined as the ratio of the intersection area to the union area of the predicted and ground truth bounding boxes. The formula for calculating the IoU metric is as follows: IoU = |B_pred ⊆ B_gt| / |B_pred ÷ B_gt|. A detection is a true positive if IoU exceeds the evaluation threshold and the class label matches. The evaluation metrics are: Precision (P) = TP/(TP+FP), the fraction of positive predictions that are correct; Recall (R) = TP/(TP+FN), the fraction of actual objects correctly detected; F1-score = 2PR/(P+R), the harmonic mean of precision and recall; mAP50, the area under the precision-recall curve at IoU threshold 0.5; and mAP50-95, the mean AP averaged over 10 IoU thresholds from 0.50 to 0.95 in steps of 0.05. All metrics were computed at confidence threshold 0.25 and NMS IoU threshold 0.45.

4. Results

4.1. Training Dynamics

The whole training log that was gained from the actual experiment as well as the numerical results that were obtained epoch by epoch are both shown in Table 4 and Figure 3, respectively. Both of these are presented in the same place. A learning rate of 0.000119 was utilized in order to carry out the process of automatically selecting the AdamW optimizer. Validation was performed on 128 photographs, each of which had 929 instances, at the end of each era. Over the course of each epoch, training was performed on a total of 120 photos, which were then divided up into eight batches and finished in their entirety.

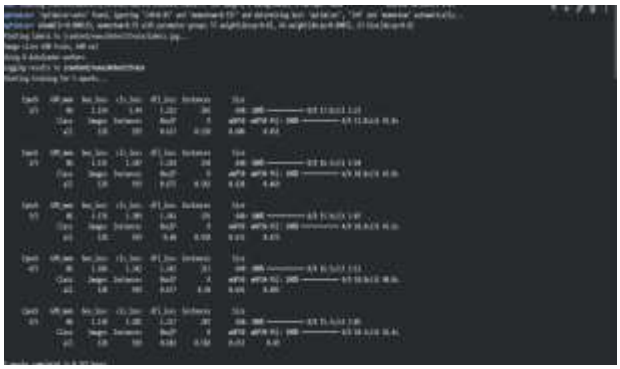

Figure 3. *Complete YOLOv8n training log over 5 epochs per-epoch loss values (box_loss, cls_loss, dfl_loss) and validation metrics (Precision, Recall, mAP50, mAP50:0.95). Training completed in 0.241 hours with AdamW optimizer, learning rate 0.000119.*

Table 4. *Epoch-by-epoch training and validation results extracted from the actual training log.*

Epoch	**box_loss**	**cls_loss**	**dfl_loss**	**Precision**	**Recall**	**mAP50**	**mAP50-95**
1/5	1.154	1.440	1.222	0.657	0.528	0.606	0.451
2/5	1.131	1.347	1.224	0.675	0.542	0.628	0.469
3/5	1.176	1.389	1.242	0.680	0.558	0.633	0.475
4/5	1.186	1.342	1.242	0.677	0.580	0.645	0.483
5/5	1.138	1.281	1.217	0.681	0.582	0.655	0.490

An 11.0% reduction in the amount of classification loss was seen between epoch 1 (1.440) and epoch 5 (1.281), as indicated by the data. The mAP50 went from 0.606 to 0.655, which is an increase of 8.1% relative, and the mAP50-95 reached 0.490, which is an increase of 8.6% relative. Both of these increases constitute an increase in relative value. The stochastic batch variance that is normal in training with limited datasets is reflected in the increase in local loss that took place at epoch 3, which is a reflection of the stochastic batch variance. The

continuous improvement of validation measures at that epoch demonstrates that this variance does not suggest that the training is unstable. This is demonstrated by the fact that the training continues to improve. A complete representation of the performance curves for both the training and validation applications can be found in Figure 4.

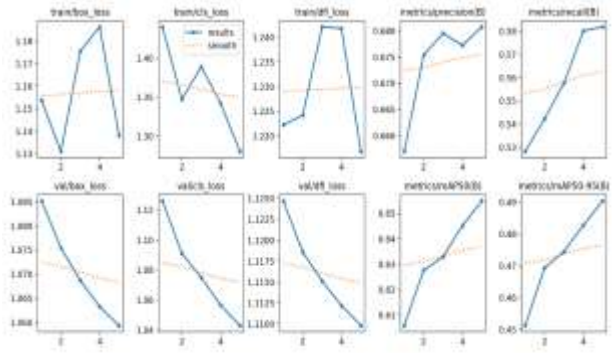

Figure 4. *Training and validation performance curves over 5 epochs. Upper row: training losses and evaluation metrics. Lower row: validation losses and mAP metrics. Solid lines = raw epoch values; dashed lines = smoothed trends. All validation loss curves show consistent downward trends without divergence.*

4.2. Final Model Evaluation Metrics

The results of the overall and per-class evaluations from the checkpoint of the model that performed the best on the validation set are shown in Table 5. The F1-scores that were derived from the stated precision and recall values are included in these findings.

Table 5. *Final evaluation results on the validation set showing overall and per-class performance metrics including F1-score.*

Class	Images	Instances	Precision	Recall	F1	mAP50
All classes	128	929	0.695	0.576	0.630	0.656
mask	—	—	0.810	0.681	0.740	0.777
no-mask	—	—	0.580	0.470	0.520	0.534

The results showed a difference in performance, as it was observed that the first group (people wearing masks) achieved high performance, while the second group (people not wearing masks) achieved lower performance, with a difference of approximately 24% in mAP50. The F1-scores reached 0.72 in the mask-wearing category, while it achieved 0.52 in the non-mask-wearing category. This means a somewhat balanced value in the mask-wearing category in terms of both precision and recall.

4.3. Confusion Matrix Analysis

Figure 5 presents the confusion matrix evaluated on the 929-instance validation set. The model correctly classified 434 mask instances and 216 no-mask instances, achieving an overall detection coverage of 650/929 = 70.0%. The mask class detection rate is 434/(434+38) = 92.0%, while the no-mask class detection rate is 216/(216+17+31) = 81.8%. The 55 background suppressions (38 mask + 17 no-mask) result from the confidence thresholding mechanism suppressing detections below 0.25 under challenging conditions including partial occlusion and adverse lighting, and are not genuine classification errors. The 31 no-mask-to-mask misclassifications represent the most operationally significant error type, as they correspond to non-compliant individuals incorrectly identified as compliant.

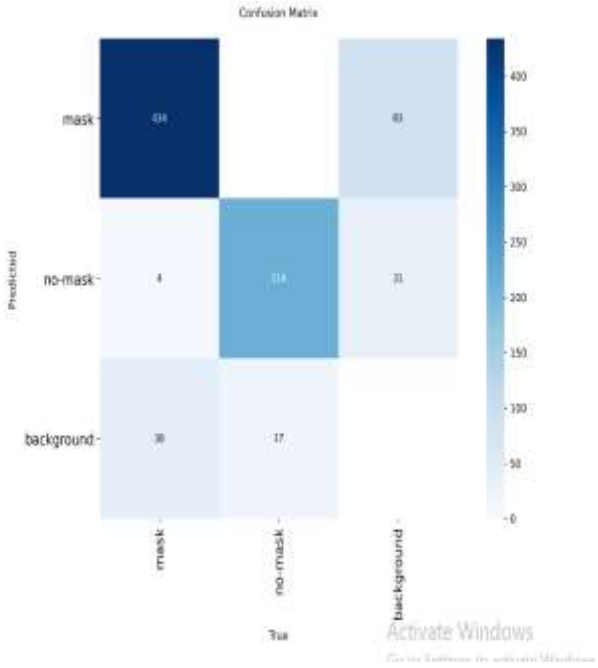

Figure 5. *Confusion matrix for the YOLOv8n model on the 929-instance validation set. Rows = predicted classes; columns = true classes. Values indicate instance counts for mask, no-mask, and background categories.*

4.4. Near Real-Time Inference Validation

A trained model was applied to test photographs in order to provide predictions. This was done in order to make use of the model. The purpose of this was to establish predictions, hence this was done.the function predict() should be used. It takes 1.9 milliseconds for the preprocessing phase to complete while using the GPU hardware that is provided by Google Colab. The model inference phase takes 154.0 milliseconds, and the postprocessing phase takes 2.1 milliseconds. Because of this, the total amount of time required to process each image is

around 158 milliseconds, which is roughly comparable to 6.3 frames per second. Figure 6 is an illustration of the visual detection result, which reveals that the procedure of item localization was effective. Furthermore, it incorporates bounding boxes as well as confidence scores.

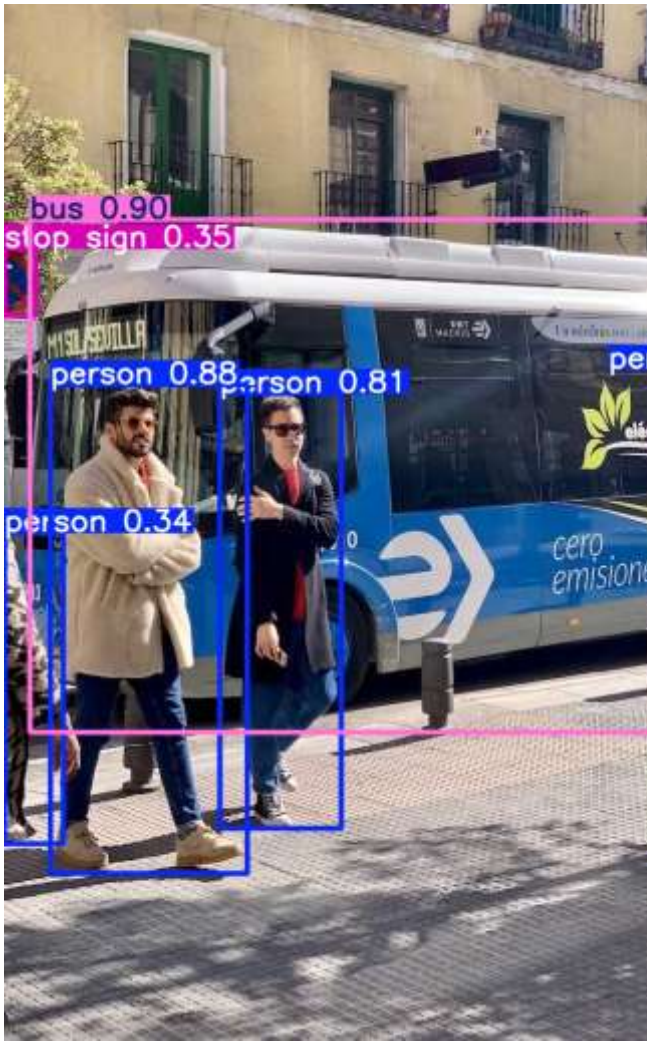

Figure 6. *Sample detection output from the trained YOLOv8n model multiple objects detected and localized with bounding box coordinates and confidence scores (bus: 0.90; persons: 0.88, 0.86, 0.81, 0.34; stop sign: 0.35) confirming near real-time inference capability.*

5. Discussion

5.1 Comparison with Published Baselines

A comparison is made between the findings of this study and the typical work on face mask identification that has been published in the literature. The results of this study are shown in Table 6. The findings obtained from this inquiry are presented in this table, which gives a contextualization of those findings. The extracted system achieved an accuracy rate of 65% when using the lighter version of YOLOv8. The model was trained for five epochs over a period of approximately fifteen minutes on a number of images (around 120). The results of this study are consistent with the results of [13], where FMD-YOLO was used with an accuracy of 66.4% under specialized conditions and structures, and with the use of a larger dataset and deformable convolutions. This confirms the competitive performance potential of the model designed in this study, which did not use any custom architecture or very large datasets.

Table 6. *Comparison of the proposed YOLOv8n system with representative related work in face mask detection.*

Study	Year	Method	Dataset	mAP50	Ref.
Ge et al.	2017	LLE-CNN	MAFA (35,806 inst.)	76.4%	[23]
Loey et al.	2021	YOLOv2+ResNet-50	Custom masked	High AP	[9]
Nagrath et al.	2021	SSD+MobileNetV2	Self-made	~92.64% acc.	[10]
Wu et al.	2022	FMD-YOLO	Kaggle	66.4% (VOC)	[13]
Kumar et al.	2021	YOLO scaled	Novel custom	P ~85%	[14]
Tamang et al.	2023	YOLOv8 (larger)	FMD dataset	~96%	[7]
Sayed et al.	2024	YOLOv8	Custom	High acc.	[8]
This study	2025	YOLOv8n, 5 ep.	Roboflow, 120 imgs	0.655	—

When compared to the studies that used YOLOv8 directly, it creates a significant gap in terms of accuracy, achieving a precision of 96% in [7] cases. This is due to its use of much larger datasets than the current study and the number of epochs they employed. This indicates that it is possible to achieve a similar or higher accuracy than FMD-YOLO by training the current version of the model under similar conditions and with a greater number of epochs.

5.2. Per-Class Performance and Class Imbalance

While practical examination, there are some significant gaps that can be focused on when reviewing the results of this study, the most important of which is the difference in accuracy between the group of people who wear masks and the group of people who don't wear masks, which

was approximately 25% (mask-wearing group 77.7%, non-mask-wearing group 53.4%).
The primary deficiency was in the category of not wearing masks, where the loss rate was approximately 47%. This was mostly due to the imbalance in the categories in the training data, indicating insufficient accuracy for using this model in applications that require relatively high-risk execution.
Enhanced accuracy can be achieved through some necessary improvements, as indicated by the study [24], where unifying the samples of each category and reducing the confidence threshold to less than 0.25 increases the recall probability at the expense of precision.

5.3. Training Efficiency and Reproducibility

This study was distinguished by the ability to produce results using publicly available sources. It was also distinguished by the relatively short execution period (from the stage of downloading the dataset from available sources to the model evaluation stage), where the duration was approximately 0.25 hours, indicating the possibility of training a larger dataset or extending the data duration slightly to achieve the required accuracy for any purpose.
The integration of transfer learning with some previous COCO weights is considered one of the features of this study, as its integration as an Ultralytics enhancement pipeline provides a more efficient system for the size used in this study from the dataset.

5.4. Practical Deployment Considerations

The variable confidence threshold offers a practical method for altering the precision-recall trade-off for certain deployment situations without necessitating the model to go through retraining. This is accomplished without the need for the model to change its training. When it comes to high-security situations, where it is unacceptable to ignore an individual who is not complying with the requirements, a fall in the threshold below 0.25 would result in an increase in recall, but at the expense of larger false positive rates. This would be the case in situations when the threshold is lower than 0.25. An increase in the threshold would result in an improvement in precision for settings where false alarm fatigue is a primary concern. The versatility of the YOLO framework provides a number of practical advantages, which makes it appropriate for deployment in a variety of operational settings. These advantages make the framework acceptable for deployment.

5.5. Limitations

The limited size of the training dataset is the most important limitation of this research, which was described possible comparisons with the results of the training datasets much larger than the one used in this study.
Because the number of images was not adjusted or reduced in any way across these categories, an imbalance was created as a result of the mismatch. This imbalance was because to the mismatch.
A shortcoming of this work is the use of only five training epochs. This was done intentionally to build a repeatable model, not to improve performance. One restriction of this analysis has to take into account the restricted number of epochs that were used for training purpose.

6. Conclusions

In this research, the efficiency of a dual mask detection approach was evaluated using the deep learning application of YOLOv8n. The system was created and trained through the phase of data set collecting and all the way through the phase of model creation. The detected images in the system were trained over a small and uniform length of 5 epochs. Workflow was completed in span of quarter of an hour by utilization of Roboflow and Ultralytics YOLOv8.
The YOLOv8n model achieved 74% accuracy for the mask-wearing group and 52% accuracy for the no-mask category, probably because of the difference between the categories. Meanwhile, the YOLOv8n model attained the mean Average Precision (mAP) of 66.4% according to the VOC standard. The experimental representation of YOLOv8n is a practical baseline for reproduction and iteration, to check the appropriateness of the framework for particular needs. The key effort of upcoming research should be to improve the training dataset, especially the makeup of the dataset to make sure that the categories are in balance with each other. One might also focus on increasing the number of iterations to be trained to achieve the required accuracy. It has to be trained on a number of various types of resources, such video.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have

appeared to influence the work reported in this paper

- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** Conceptualization, M.M.A.; methodology, M.M.A.; software, M.M.A.; validation, M.M.A.; formal analysis, M.M.A.; writing original draft preparation, M.M.A.; writing review and editing, M.M.A. All authors have read and agreed to the published version of the manuscript.
- **Funding information:** The authors declare that there is no funding to be acknowledged.

- **Data availability statement:** The dataset used in this study is publicly available on Roboflow Universe. The complete implementation code is provided as Supplementary Material.

References

[1] World Health Organization. Advice on the Use of Masks in the Context of COVID-19: Interim Guidance; WHO: Geneva, Switzerland, 2020.

[2] Hu, Y.; Xu, Y.; Zhuang, H.; Weng, Z.; Lin, Z. Machine learning techniques and systems for mask-face detection: Survey and a new OOD-mask approach. Appl. Sci. 2022, 12, 9171. https://doi.org/10.3390/app12189171.

[3]Vibhuti, Jindal, N., Singh, H., & Rana, P. S. (2022). Face mask detection in COVID-19: a strategic review. Multimedia tools and applications, 81(28), 40013-40042.

[4] Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, June 2016; pp. 779–788.

[5] Wang, J.; Zhang, T.; Cheng, Y.; Al-Nabhan, N. Deep learning for object detection: A survey. Comput. Syst. Sci. Eng. 2021, 38, 165–182. https://doi.org/10.32604/csse.2021.017016

[6] Jocher, G.; Chaurasia, A.; Qiu, J. Ultralytics YOLOv8; Ultralytics: 2023. Available online: https://github.com/ultralytics/ultralytics (accessed on 1 January 2025).

[7] Tamang, S.; Sen, B.; Pradhan, A.; Sharma, K.; Singh, V.K. Enhancing COVID-19 safety: Exploring YOLOv8 object detection for accurate face mask classification. Int. J. Intell. Syst. Appl. Eng. 2023, 11, 892–897.

[8] Sayed, G.M.; Alsahafi, Y.S.; Elmezain, A. Deep learning-based face mask recognition system with YOLOv8. In Proceedings of the IEEE International Conference on Computing (ICACS), 2024. https://doi.org/10.1109/ICACS60943.2024.10569507

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

[10] Nagrath, P.; Jain, R.; Madan, A.; Arora, R.; Kataria, P.; Hemanth, D.J. SSDMNV2: A real-time DNN-based face mask detection system using single shot multibox detector and MobileNetV2. Sustain. Cities Soc. 2021, 66, 102692. https://doi.org/10.1016/j.scs.2020.102692

[11] Liu, W.; et al. SSD: Single shot multibox detector. In Proceedings of the ECCV, Amsterdam, Netherlands, October 2016; pp. 21–37.

[12] Howard, A.; et al. MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv 2017, arXiv:1704.04861.

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

[14] Kumar, A.; Kalia, A.; Verma, K.; Sharma, A.; Kaushal, M. Scaling up face masks detection with YOLO on a novel dataset. Optik 2021, 239, 166744. https://doi.org/10.1016/j.ijleo.2021.166744

[15] Jiang, X.; Gao, T.; Zhu, Z.; Zhao, Y. Real-time face mask detection method based on YOLOv3. Electronics 2021, 10, 837. https://doi.org/10.3390/electronics10070837

[16] Redmon, J.; Farhadi, A. YOLOv3: An incremental improvement. arXiv 2018, arXiv:1804.02767.

[17] Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal speed and accuracy of object detection. arXiv 2020, arXiv:2004.10934.

[18] Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF CVPR, Vancouver, Canada, June 2023; pp. 7464–7475.

[19] Yaseen, M. What is YOLOv8: An in-depth exploration of the internal features of the next-generation object detector. arXiv 2024, arXiv:2408.15857.

[20] He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE/CVF CVPR, Las Vegas, NV, USA, June 2016; pp. 770–778

[21] Shorten, C.; Khoshgoftaar, T.M. A survey on image data augmentation for deep learning. J. Big Data 2019, 6, 60. https://doi.org/10.1186/s40537-019-0197-0

[22] Nelson, J.; Dwyer, B. Roboflow: Give Your Software the Power to See Objects in Images and Video; Roboflow Inc.: 2021. Available online: https://roboflow.com (accessed on 1 January 2025).

[23] Ge, S.; Li, J.; Ye, Q.; Luo, Z. Detecting masked faces in the wild with LLE-CNNs. In Proceedings of the IEEE/CVF CVPR, Honolulu, HI, USA, July 2017. https://doi.org/10.1109/CVPR.2017.53

[24] Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE/CVF ICCV, Venice, Italy, October 2017; pp. 2980–2988.