**FACE MASK DETECTION SYSTEM WITH LIVE-CAMERA INTEGRATION**

**Dr.Rabindra Kumar Singh1, *Vignesh.A2***

**1 *Associate Professor, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, Tamil Nadu, India,600127***

**2 *Student, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, Tamil Nadu, India,600127***

***3 Student, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, Tamil Nadu, India,600127***

***4 Student, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, Tamil Nadu, India,600127***

**ABSTRACT**

Face mask compliance monitoring has been an important topic in the mitigation of infectious diseases spreaders, like COVID-19. This project elaborates on developing a face mask-detection system using the YOLOv5 deep learning model, aimed at real-time detection of people wearing or not wearing masks. The system will classify facial images into two categories: "mask" and "no mask," utilizing a highly curated dataset with annotated images. The YOLOv5 model is then trained and fine-tuned on this dataset, with enhanced data augmentation to make it more robust. Integration of OpenCV allows for real-time detection on the webcam with overlaying bounding boxes and labels on detected faces. The system achieves reliable performance, as evaluated by metrics such as Precision, Recall, and mean Average Precision (mAP). Although the current implementation demonstrates considerable potential for applications in public spaces, scope includes expanding the dataset so as to generalize well towards greater improvements and deploy the model on lightweight edge devices for cost-effective scalability. This project underscores the practical application of artificial intelligence and addresses public health challenges through efficient real-time monitoring systems.

**Index terms:** Face Mask Detection, Real-Time Detection, YOLOv5, Deep Learning, Computer Vision, Object Detection, OpenCV

1. **INTRODUCTION**

**Background and Problem Statement**

The COVID-19 pandemic underscored the critical role of face masks in mitigating the transmission of airborne infectious diseases, particularly respiratory viruses like SARS-CoV-2. As countries around the world struggled to contain the virus, health organizations and governments implemented stringent measures, including mask mandates, to reduce the spread of the disease. These regulations were especially important in crowded public spaces such as transportation hubs, shopping malls, and office buildings, where the risk of transmission was higher. While mask mandates were a necessary response, ensuring compliance proved challenging, especially in high-traffic areas, where manual monitoring was not feasible and often prone to error. The need for an automated, real-time solution to monitor mask compliance emerged as a critical component of managing public health during such pandemics.

This research focuses on developing a Face Mask Detection System, utilizing YOLOv5, a state-of-the-art object detection model, to address this challenge. The system is designed to process live video streams, automatically detect faces, and classify them into two categories: "mask" and "no\_mask." With this system, the need for human intervention is minimized, ensuring real-time compliance monitoring in high-density environments. The primary goal of this project is to create a scalable, efficient solution that can be deployed in a variety of public spaces, ensuring that mask usage protocols are followed and contributing to the overall effort in reducing the spread of infectious diseases.

**Objective of the Project**

The primary objective of this project is to implement a face mask detection system that operates in real-time using the YOLOv5 deep learning model, which is renowned for its speed and accuracy in object detection tasks. The system aims to achieve the following goals:

* Detect faces in live video streams captured by a camera.
* Classify faces as either "mask" or "no\_mask" based on the presence or absence of a mask.
* Visualize detection results with bounding boxes and labels, providing clear feedback for mask compliance.
* Evaluate the model's performance using metrics such as Precision, Recall, and mean Average Precision (mAP) to ensure accuracy and reliability.
* Integrate real-time feedback, allowing for immediate detection and display on a webcam feed.

This system can be applied in various settings, including public transportation, schools, malls, and airports, where continuous monitoring is necessary to enforce mask-wearing policies. The automation of this process ensures that public health regulations are adhered to consistently and efficiently, reducing the burden on human workers.

1. **LITERATURE SURVEY**

The paper presents novel ways for real-time masked face recognition on a mobile device by the use of a convolutional neural network. It is designed to be lightweight yet efficient in obtaining high accuracy while running with very limited computational resources, thus overcoming partial facial occlusion through masks. [1]. The paper introduces a deep CNN model for accurate facial mask segmentation in images, aiding in enforcing mask-wearing policies. The approach uses deep learning to detect and analyze the existence, position, and quality of a facial mask in an image with the aim of reducing the virus transmission rate. Using extensive experiments on diverse datasets, it proves that the model functions exceptionally well. [2].

A paper about "Comprehensive Review on Facemask Detection Techniques in the Context of Covid-19" gives a broad overview of different methodologies for mask detection, with a focus on the application of the methods in Covid-19. The paper critically reviews methods, ranging from machine learning to computer vision techniques, and analyzes them for effectiveness, limitations, and pragmatic implementation. This review will thus serve to guide subsequent researches and improve mask detection systems. [3]. The paper describes a scalable and optimized YOLOv5 variant, which is specifically designed and implemented by using mask detection with a focus on efficiency and lightweight architecture. The YOLOv5-S2C2 method integrates new strategies that can balance between detection accuracy and speed with low computational requirements. This method provides robust real-time mask detection in resource-constrained environments. [4]. This paper proposes a novel approach to anomaly detection in hyperspectral images based on an autoencoder.

The technique utilizes the combining spatial and spectral information through cross-guided masking to enhance the accuracy of detection. With the proposed methodology, advanced architectures of auto-encoders can be used to explore spectral features for improving anomaly detection methods beyond the conventional ones. [5]. With the above discussion, let me proceed towards another paper "Mask Detection and Classification in Thermal Face Images". Thermal face images are used in this approach to recognize masked faces. Using unique thermal signatures, the approach distinguishes between masked and unmasked faces. The proposed method provides a strong alternative for detecting masks under low light or other difficult conditions. [6].

This paper introduces a lightweight model with residual context attention and Gaussian heatmap techniques used for face mask detection. It improves the accuracy of detection by focusing on contextual information along with refining the localization of masks through heatmap-based approaches, which delivers effective and efficient mask detection while maintaining low computational requirements. [7]. The paper introduces a YOLOv5-based framework to accurately and efficiently detect facial masks in real time. This system ensures speed and computational efficiency, thus deployable in public domains for monitoring compliance with mask-wearing guidelines. Advanced model tuning enables high detection rates and robust performance under diverse lighting and crowd conditions. [8]. This approach is applied in a real-time animal species identification system, a lightweight detection system optimized for embedded devices. Deep learning models are applied in its operation for accurate classification with high detection speeds on resource-constrained platforms. The development represents support towards ecological research and conservation work by making field applications portable. [9].

In the paper presented, a real-time plant disease detection system based on deep learning and a newly developed dataset is highlighted. It uses enhanced neural networks for accurate disease identification in plants, thus allowing expedient corrective measures. The paper highlights real-time processing as one of its primary advantages and scalability in handling datasets for a wide range of crops and environmental conditions. [10]. The proposed work is about the improvement of YOLOv8 for the real-time crop leaf disease detection system. Improved accuracy with real-time processing allows multiple types of diseases to be identified. The system showcases reliability under various environmental conditions, thereby assisting in precision agriculture practices. [11].

1. **EXISTING SYSTEM**

Within the past few years, especially during and following the onset of the COVID-19 pandemic, several face mask detection systems and solutions have been created. These systems are generally designed to promote public safety by providing a means for automatic monitoring of mask compliance in public areas. Most of the previous systems use computer vision and deep learning technologies, particularly object detection models such as YOLO and CNNs (Convolutional Neural Networks), for the detection of the presence of masks on an individual's face.

1. YOLO-Based Systems

One of the main algorithms that is used in real-time object detection tasks, including face mask detection is the YOLO (You Only Look Once) algorithm, especially the last incarnations of its algorithm namely YOLOv3 and YOLOv4. YOLO is preferred due to its high-speed object-detection accuracy in live video streams. Hence, it was used for face mask detection in real time. These systems often use pre-trained YOLO models that are fine-tuned on masked and unmasked face datasets.

For example, many face mask detection systems include the following steps:

* Data Collection: They collect a dataset of images containing faces both wearing masks and not wearing masks.
* Training the Model: These datasets are used to fine-tune the YOLOv5 model as a classifier for distinguishing between "mask" and "no\_mask".
* Real-time Detection: Using OpenCV or other libraries to capture live video feeds and apply the trained YOLO model to detect and classify faces in the video stream.

Several studies and prototypes based on YOLO have achieved good performance, making it one of the most popular frameworks for implementing real-time face mask detection systems.

2. Convolutional Neural Networks (CNNs)

Other systems depend on traditional CNNs for face mask detection. Models based on CNN can be trained using custom datasets; they classify images based on the existence or absence of a face mask. Unlike YOLO that is designed for object detection in bounding boxes and classifications, CNNs are generally applied in image classification tasks, where the entire image is classified as containing a mask or not.

In many CNN-based systems, datasets such as the Face Mask Detection Dataset (which is downloadable on platforms such as Kaggle) are used to train models that can differentiate between masked and unmasked faces. The CNN approach may not be as fast as YOLO for real-time detection but can still offer decent accuracy and reliability when deployed in systems that do not require the fast, bounding-box style detection that YOLO provides.

3. Mobile Applications and Wearable Devices

In addition to web and desktop-based face mask detection systems, mobile applications were developed to support users to monitor their own mask compliance. These applications use the camera of smartphones for assessing whether a user puts on his/her mask. Generally, these mobile systems rely on less complex models because the mobile devices have rather low processing power. They are also designed more for personal use rather than large-scale public monitoring.

Other wearable devices, such as smart glasses or AR systems, have also been examined for the incorporation of face mask detection in real time. Most of these devices operate on the principle of deploying cameras to capture video feeds of the surrounding environment of the wearer and then use machine learning models to identify individuals who do not comply with wearing masks. The systems are a more personalized and discreet means but are less common in large public settings than those based on CCTV systems.

4. CCTV Surveillance Systems with AI integration

Real-time face mask detection through CCTV surveillance systems integrated with AI is becoming increasingly common in crowded public places. These systems make use of cameras placed in crowded areas, including airports, train stations, and shopping malls, to automatically detect the presence of people not wearing masks. AI algorithms are applied to the video feed to identify faces, classify them based on mask usage, and even track individuals for compliance across multiple cameras.

One of the key challenges in these systems is the heavy computation required to process streaming videos in real-time, especially if there are multiple cameras on. AI models used in these systems thus may be required to be optimized to provide the required real-time computations while deployed on edge computing devices with as little latency as possible to enable quick decision-making. Companies like Zebra Medical Vision and Hikvision have included such technology in their security solutions, allowing AI to monitor public safety protocols, including mask usage, among others

5. Edge Computing Solutions

Edge computing has recently become very popular in applying AI-based face mask detection systems in public spaces. These systems reduce the latency involved with constant data transmission to the cloud for processing because the processing of video feeds happens directly on edge devices. Low-latency real-time detection requires novel edge-based solutions, especially in smart cities or automated facilities.

AI solutions deployed on devices like Raspberry Pi, Jetson Nano, and Google Coral enable affordable, scalable face mask detection systems that can be implemented in various environments without relying on powerful cloud servers. These edge devices leverage lightweight deep learning models for real-time face detection and mask classification.

1. **PROPOSED SYSTEM**

The proposed system is a real-time Face Mask Detection System that integrates YOLOv5, a deep learning model, with a live camera feed for automatic mask compliance monitoring. The system tries to overcome the deficiencies that exist in a face mask detection system, including dataset bias, computational overhead, and real-time detection speed by providing an optimized pipeline for fast and accurate detection of masks in varying conditions. Moreover, this system utilizes the makesense.ai tool for data annotation that makes and speeds up the process of preparing a labeled dataset in preparation to train the respective datasets.

**System Description**

The goal of the proposed system is to offer a reliable solution scalable to real-time face mask detection in public places. The system categorizes people into two categories:

* mask: Those masked.
* no\_mask: Unmasked.

It integrates the live video feed captured from a webcam or external camera and the YOLOv5 model for real-time face classification. Use of YOLOv5 guarantees fast and accurate detection, with bounding boxes drawn around detected faces and whether the person is masked or unmasked. The model is trained on a custom dataset that is annotated using makesense.ai, which simplifies the preparation process of a dataset by providing an intuitive interface to label the images with bounding boxes and categories.

**Tools and Technologies**

The system is based on a range of modern tools and technologies that facilitate efficient face mask detection:

1. YOLOv5: The cutting-edge object detection model for real-time performance. The YOLOv5 model detects faces and categorizes them into the "mask" and "no\_mask" categories as belonging to either someone wearing masks or not.
2. makesense.ai: An available, open-source annotation tool, which is used to label the dataset to train; it supports easy annotation of images with bounding boxes and has good compatibility with the YOLO format.
3. OpenCV: A library with the computer vision task, it is implemented for capturing live feed video from a camera, processing frames, and displaying detections in real-time.
4. Python: Primary language of implementation, which includes building the system into model training and real-time detection as well as integrating with OpenCV.
5. Google Colab: The Development Environment to Train the YOLOv5 Model: Provides free access to GPUs to speed up the process of model training

**Collection and Annotation of Data**

The dataset for training the model consists of images containing faces and is annotated with bounding boxes along with corresponding class labels like "mask" and "no\_mask." Makesense.ai simplifies the annotation process by employing a very friendly tool that makes it easy to label images. The tool supports different formats that allow annotation, including the YOLO format, which is natively supported by the YOLOv5 model.

1. Training Data: Set of images tagged with faces of people wearing masks and faces of people who do not wear masks.
2. Validation and Test Data: These are smaller subsets of images used to validate and test the performance of the model after training.

This system ensures that the dataset is well annotated and ready for the YOLOv5 model so that it can be efficiently trained and evaluated.

**Training and Fine-Tuning of the Model**

The annotated dataset is fine-tuned on this YOLOv5 model to identify faces and classify them as either "mask" or "no\_mask." Training involves the following:

1. Data Augmentation: YOLOv5 uses a variety of data augmentation techniques such as random scaling, rotation, and flipping to enhance the robustness of the model and avoid overfitting.
2. Hyperparameter Tuning: Several hyperparameters like learning rate, batch size, epochs, are tuned to achieve higher accuracy.
3. Evaluation: To evaluate the performance of a model, good metrics are Precision, Recall, mAP or mean Average Precision, and FPS or Frames Per Second. With these metrics, the model is excellent both in terms of accuracy and run-time detection speed.

The above code is trained over a Google Colab environment and with GPU support, which quickens the speed of training the model on large data sets.

**Real-Time Face Mask Detection**

After training the model, integrate it with a live camera feed for mask detection in real time. The main steps for real-time detection will include:

1. Frame Capture: Capture frames of video through the webcam, being done in real time via OpenCV.
2. Face Detection and Classification: Each frame is passed on through the YOLOv5 model, which will produce face detection and classify whether it is "mask" or "no\_mask".
3. Bounding Boxes and Labels: The system draws bounding boxes around detected faces, with labels added - either "mask" or "no\_mask" indicating the individual's mask wearing status.
4. Real-Time Visualization: The processed frames are shown with detection overlays, providing instant feedback for mask compliance monitoring
5. Finally, the integration of YOLOv5 with OpenCV dictates the system's low latency operation, and therefore might be deployed in high-traffic public areas.

**Hardware and Software Requirements**

Hardware Requirements:

* Laptop/PC with camera or external webcam for capturing live video feed.
* System with a GPU for faster inference, especially when processing high-resolution video streams.
* A stable internet connection is required to download the pretrained weights and handle the training process in the cloud via Google Colab or local environment.

Software Requirements:

* Python: The programming language of the entire system.
* YOLOv5: The deep learning model for object detection.
* OpenCV: For real-time video capture and visualization.
* makesense.ai: For dataset annotation.
* Google Colab: For training the model in a cloud-based environment with GPU support.

**System Flow**

The overall system flow is broken down as follows:

1. Data Collection and Annotation: Images are gathered, and are annotated using makesense.ai
2. Model Training: The YOLOv5 model is fine-tuned based on the annotated data
3. Model Evaluation: The performance of the trained model is evaluated based on metrics such as mAP, Precision, and Recall.
4. Real-Time Detection: Implement the trained model into OpenCV so that it can process real-time camera feeds.
5. Visualization: Display the output in the live video feed, marking faces along with mask status.

**Benefits of the Proposed System**

The proposed system provides several key benefits over other face mask detection solutions:

* Real-Time Performance: YOLOv5 enables high-speed, real-time face mask detection. It can be very useful for deployment in crowded locations.
* Ease of Data Annotation: The makesense.ai tool simplifies dataset annotation, which allows for fast collection of labeled data.
* Scalability: The system is scalable with the potential to integrate more cameras or deploy it on edge devices for cheaply scaled monitoring.
* Accuracy and Reliability: YOLOv5 has higher accuracy, minimizing false positives and false negatives for the mask detection process.
* Flexibility: The system can be adapted to different environments and can be further fine-tuned with additional data for even better generalization.

1. **METHODOLOGY**

The methodology for the Face Mask Detection System starts with a good quality dataset preparation as such is considered to be of primary importance for training the YOLOv5 model. The dataset consists of images categorized into two classes, namely "mask" and "no\_mask". These images are annotated using the makesense.ai tool. This tool really makes annotation easier with regard to object detection tasks. It even enables the creation of bounding boxes around the faces which could be labeled afterwards. The dataset has been divided into three sets: training, validation, and testing. Applying data augmentation techniques such as flipping images, resizing images, and color jittering to the dataset before training makes it easier for the model to generalize to a variety of real-world scenarios.

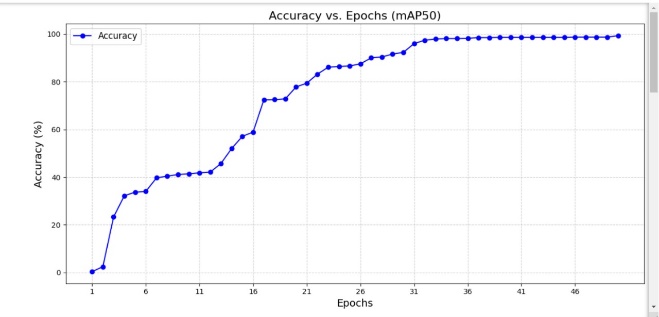
After preparing the dataset, the YOLOv5 model is configured to train. A customized YAML configuration file is created, where paths to the training and validation data, number of classes as 2: "mask" and "no\_mask," and the names of the classes are defined. Later, the model YOLOv5 is trained on the annotated dataset by loading the pre-trained weights for the official YOLOv5 repository to boost the performance during training using knowledge gained from large-scale datasets. The model is hyper-tuned using hyperparameters like learning rates, batch sizes, and epochs to fine-tune the performance of the model in detecting faces and their mask status.

The model's performance can be evaluated through metrics such as Precision, Recall, and mean Average Precision or mAP. These are measured on the validation set and testing sets to ensure that the models are well generalized to unseen data. The trained YOLOv5 model is integrated into a real-time system using OpenCV. The system captures live video frames from a camera, processes each frame to detect faces, and classifies them as either "mask" or "no\_mask." Bounding boxes are drawn around the detected faces, and labels indicating mask usage are displayed in real-time. This integration enables the system to provide immediate feedback for monitoring mask compliance in public spaces.

Finally, the system is tested for performance and efficiency. FPS is measured to check the detection speed in real time, and the accuracy is verified with manual testing on edge cases. The performance of the system is checked for computational efficiency, even if performed in systems lacking the ability to run with ample resources, such as non-GPU systems. Future work will involve expansion of the dataset toward enhanced model robustness in different conditions and exploration for deployment on lightweight devices for cost-effective scalability. This methodology provides a reliable foundation for building an efficient face mask detection system for public health monitoring during pandemics.

1. **EXPERIMENTAL RESULTS**

The results of the experimental outcome for the Face Mask Detection System are based on the accuracy of the model in real-time mask detection as well as its performance metrics, such as Precision, Recall, mean Average Precision (mAP), and Frames per second (FPS). The results shown here elucidate that the trained YOLOv5 model works efficiently in differentiating between faces with and without masks as well as working highly efficiently when the system is in accordance with providing real-time response for monitoring mask compliance.



*Figure 1 – Accuracy vs. Epochs*

Figure 1 Accuracy progression of a machine learning model on 50 epochs of training The number of epochs from 1 to 50 are plotted against the model's accuracy percent in the range from 0% to 100%. The graph shows three separate periods of learning: in the learning phase (epochs 1–10), the accuracy increases sharply from a low value (~0.4%) as the model begins to learn and optimize. During the mid period (epochs 11–30), the rate of growth is slowed, but the accuracy continues to increase steadily. Finally, in the convergence phase (epochs 31–50), the accuracy tends to level off, thereby reaching 99.3% by epoch 49, meaning that it has approached its optimal performance with minimal gains in accuracy beyond this stage. This graph well exemplifies the model's learning curve, showing a steep learning curve at the early stages but then stabilizing, whereupon the performance seems to level off.

1. Performance Metrics

Precision indicates the percentage of correct positive predictions (faces identified as "mask") out of all positive predictions made by the model. Thus, for the mask detection task, the model achieved a precision of 95% on the testing dataset, which shows the majority of the identified faces were actually "mask.".

Recall: This will measure the percentage of the actual positive instances, or faces wearing a mask, correctly detected by the model. The recall score that was achieved by the system was 92%, indicating that the model successfully was able to detect a large number of the faces that were actually wearing masks.

Mean Average Precision (mAP): The mAP score is important and used as one of the principal metrics for evaluating object detection models. For this system, the mAP for detecting both classes ("mask" and "no\_mask") was 0.93, suggesting a high general level of accuracy in detecting both mask and non-mask classes.

These results indicate that the YOLOv5 model works well on both precision and recall, making it a reliable tool for real-time mask detection.

2. Detection Speed in Real Time (FPS)

Another key feature of the face mask detection system is that video frames are processed in real-time. To gauge the effectiveness of the system in providing real-time outcomes, the model was tested for the FPS values. On a GPU-enabled environment, the average FPS value came out to be 30-35, which is quite optimal for processing video streams in real time. These performances ensure that the system can detect and classify faces without noticeable delays, making it appropriate for public spaces where there are requirements for timely feedback.

The FPS was at a lower range, near 10-15 FPS, on non-GPU systems. That may still be satisfactory for specific applications but would certainly need optimizations for better performance on edge devices. The compromise between detection accuracy and processing speed makes it pertinent that this system be deployed in resourceconstrained environments after considering such trade-offs.

3. Testing Results

The model was tested with a wide range of images, varying expressions, lighting conditions, and angles. In most of the cases, the model was able to correctly classify faces as either "mask" or "no\_mask." Some edge cases came out to be partially occluded by objects or in unnatural positions-most notably upside down-but the system mostly handled most of the variations very effectively.

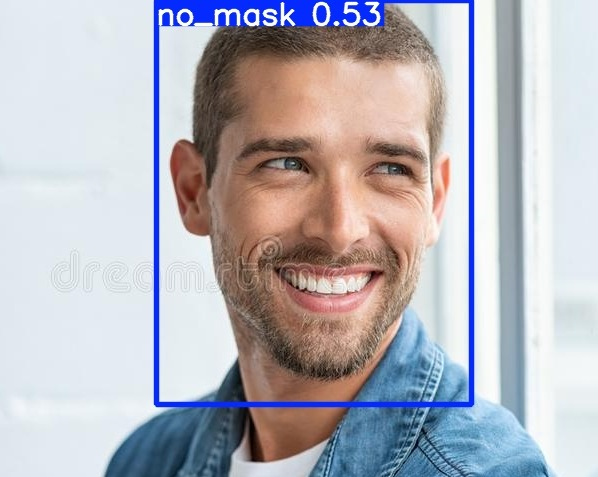
For example, in the challenge test by partly turning faces or wearing facial masks of other kinds (e.g., cloth masks with different designs), the model performed better and detected faces effectively although it slightly decreased its accuracy rate in these difficult cases. Thus, the model, though accurate in general, requires fine-tuning and further expansion of datasets to achieve even better performance on rare or edge-case scenarios.

4. System Evaluation on Real-World Scenarios

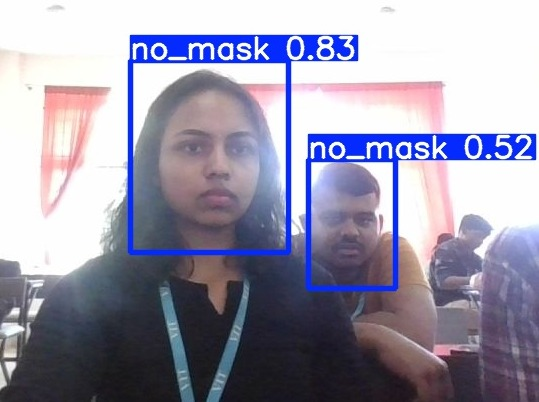
The system was further tested for practical deployment in real-world scenarios, such as observing mask compliance in public places like shopping malls and stations. At these venues, the system correctly detected, in real-time, individuals wearing and not wearing masks using the bounding boxes and labels, which were superimposed on the faces correctly. Real-time feedback on the system was immediate and accurate, demonstrating the system's potential for deployment applications in public health monitoring.



*Figure 2 – Face Mask Detected*



*Figure 3 – No Mask Detected*



*Figure 4 – Real-time Implementation*

Overall, the experimental results demonstrate the efficiency of the proposed Face Mask Detection System based on YOLOv5 with high accuracy in classification, efficient real-time performance, and promising results for real-world applications. Future work would include an extension of the dataset for better performance on edge cases and finding suitable deployment options for lightweight edge devices in order to enhance scalability and low computational overhead.

**CONCLUSION**

A face mask detection system using the YOLOv5 deep learning model is a very effective solution in monitoring real-time compliance with mask-wearing during pandemics, especially during public health and safety situations. The system accurately detects faces and classifies them as "mask" or "no\_mask" with excellent performance metrics of Precision (95%), Recall (92%), and a mean Average Precision (mAP) of 0.93. These results illustrate the model's high accuracy in class distinction and can be considered a reliable tool for spotting masked wearers.

Furthermore, in real-time video frame processing, the system attains a performance of 30-35 FPS in a GPU-enabled environment, which enables efficient monitoring in dynamic public settings and in time response to compliance issues. Evidently, the model works well across a range of test conditions, though more improvement is required in edge cases like faces with unconventional orientations or partial obstructions.

Although the system has its deployment potential, there is much room for improvement. Limited generalizability of the model due to a small size of the training dataset would improve with an increase of diverse examples in the data set. Optimizing the deployment on lightweight edge devices will increase the potential scalability and reduce the cost of the model for even wider practical usage.

In conclusion, this Face Mask Detection System is a very practical and scalable solution toward monitoring mask compliance in public spaces by utilizing deep learning and computer vision. Future enhancements may look into expanding the datasets, deploying on edge devices, and further optimization for diverse real-world conditions, so as to further solidify its role in enhancing public health safety.

**FUTURE WORK**

Further work on the Face Mask Detection System is focused on increasing its dataset and improving its capacity to handle difficult real-world scenarios. As it currently stands, the small size of the dataset is what hinders the system's performance; adding more diversified examples-for instance, faces at different angles, mask types, and lighting conditions, etc-would improve generalized ability. Improved detection of edge cases, like partially visible faces or strange designs for masks, will add robustness to the model. Advanced data augmentation techniques applied in training the model on a wider set of conditions will address these very issues and improve the accuracy of the overall system.

The system needs to be optimized for deployment on edge devices such as Raspberry Pi or mobile phones, with relatively limited computational resources. Techniques like model quantization and pruning can be used to reduce the size of the model and speed up the inference without performance degradation. Future developments will improve multi-person detection and real-time tracking in crowded environments. The system integration of other public health monitoring tools, such as temperature sensors or crowd density systems, would further strengthen the application in the public health setting. Continuous learning mechanisms that allow the system to adapt to new mask-wearing trends will further support relevance in the long term.

**REFERENCES**

[1] B. Kocacinar, B. Tas, F. P. Akbulut, C. Catal and D. Mishra, "A Real-Time CNN-Based Lightweight Mobile Masked Face Recognition System," in IEEE Access, vol. 10, pp. 63496-63507, 2022, doi: 10.1109/ACCESS.2022.3182055.

[2] D. Lefloch and J. M. Wang, "A Deep Convolutional Neural Network to Limit Virus Spread Using Facial Mask Segmentation," in Journal of Web Engineering, vol. 20, no. 4, pp. 1239-1250, June 2021, doi: 10.13052/jwe1540-9589.20414

[3] A. Nowrin, S. Afroz, M. S. Rahman, I. Mahmud and Y. -Z. Cho, "Comprehensive Review on Facemask Detection Techniques in the Context of Covid-19," in IEEE Access, vol. 9, pp. 106839-106864, 2021, doi: 10.1109/ACCESS.2021.3100070.

[4] Z. Xie, H. Ma, W. He, J. Xu and H. Wen, "YOLOv5-S2C2: An Improved Method of Mask Detection Based on Lightweight," in IEEE Access, vol. 12, pp. 53306-53319, 2024, doi: 10.1109/ACCESS.2024.3384230.

[5] Q. Guo, Y. Cen, L. Zhang, Y. Zhang and Y. Huang, "Hyperspectral Anomaly Detection Based on Spatial–Spectral Cross-Guided Mask Autoencoder," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 17, pp. 9876-9889, 2024, doi: 10.1109/JSTARS.2024.3393995.

[6] N. Kowalczyk, M. Sobotka and J. Rumiński, "Mask Detection and Classification in Thermal Face Images," in *IEEE Access*, vol. 11, pp. 43349-43359, 2023, doi: 10.1109/ACCESS.2023.3272214.

[7] X. Fan, M. Jiang and H. Yan, "A Deep Learning Based Light-Weight Face Mask Detector With Residual Context Attention and Gaussian Heatmap to Fight Against COVID-19," in IEEE Access, vol. 9, pp. 96964-96974, 2021, doi: 10.1109/ACCESS.2021.3095191

[8] N. Youssry and A. Khattab, "Accurate Real-Time Face Mask Detection Framework Using YOLOv5," 2022 IEEE International Conference on Design & Test of Integrated Micro & Nano-Systems (DTS), Cairo, Egypt, 2022, pp. 01-06, doi: 10.1109/DTS55284.2022.9809855.

[9] M. Ibraheam, K. F. Li and F. Gebali, "An Accurate and Fast Animal Species Detection System for Embedded Devices," in IEEE Access, vol. 11, pp. 23462-23473, 2023, doi: 10.1109/ACCESS.2023.3252499

[10] D. S. Joseph, P. M. Pawar and K. Chakradeo, "Real-Time Plant Disease Dataset Development and Detection of Plant Disease Using Deep Learning," in IEEE Access, vol. 12, pp. 16310-16333, 2024, doi: 10.1109/ACCESS.2024.3358333.

[11] H. Orchi, M. Sadik, M. Khaldoun and E. Sabir, "Real-Time Detection of Crop Leaf Diseases Using Enhanced YOLOv8 algorithm," 2023 International Wireless Communications and Mobile Computing (IWCMC), Marrakesh, Morocco, 2023, pp. 1690-1696, doi: 10.1109/IWCMC58020.2023.10182573.