

Enhancing Public Health Measures: A Deep Learning Approach for Real-Time Facemask Recognition to Mitigate the Spread of Infectious Diseases

Maryrose Ngozi Umeh¹, Blessing Nwamaka Iduh², Francisca Ngozi Faseki³, Nwamaka Peace Oboti⁴, Ogechi Hilary Nwabudike⁵

^{1,2,4}Department of Computer Science, Faculty of Physical Sciences, Nnamdi Azikiwe University, Awka, Nigeria

³Department of Computer Science, Faculty of Computing Science, Nigerian Army University, Biu, Bornu State, Nigeria

DOI: <https://doi.org/10.51244/IJRSI.2025.12040071>

Received: 21 March 2025; Accepted: 25 March 2025; Published: 10 May 2025

ABSTRACT

This paper presents a system for real-time facemask recognition, integrating automation through deep learning to enhance public health measures and mitigate the spread of infectious diseases. The proposed system employs OpenCV's Haar Cascade Classifier for face detection, establishing the foundational framework for subsequent processes. The heart of the system lies in its Deep Learning Model, consisting of two critical components: Data PreProcessing and CNN Training. The former involves meticulous steps such as resizing, color space conversion, and normalization, while incorporating data augmentation techniques to ensure a diverse and robust training dataset. The latter employs convolutional neural networks (CNNs) to autonomously learn and discern facemask presence, contributing to the system's accuracy. The User Interface (UI) component, built using tkinter, provides a visual representation of the video stream, displaying real-time face mask detection results. The system's scalability and efficiency are heightened by its automated nature, eliminating the need for human intervention. Furthermore, the UI serves as a user-friendly interface, potentially expanding to include features like data logging and reporting. The Notification System, integrated using the 'plyer' library, enhances user engagement by providing real-time alerts on the desktop. These notifications act as reminders for individuals not adhering to mask guidelines, fostering immediate compliance. The proposed system boasts several advantages, including automation, scalability, adaptability to diverse conditions, security against spoofing attacks, cost reduction, improved accuracy, and increased flexibility. Methodologically, the research adopts an applied research methodology, combining empirical, experimental, and practical research elements to develop and deploy an effective solution. At the end of the work, a real-time face mask recognition system, that is capable of analysing live video feeds from public spaces; detect mask-wearing violations, and issue alerts when necessary, was achieved. Notably, the deep learning model consistently demonstrated high accuracy, effectively minimizing both false positives and false negatives.

Keywords: Face Mask Detection, Deep Learning, Intelligent Disease Prevention, Infectious Disease, Public Health, Covid-19 Prevention

INTRODUCTION

In recent years, the world has faced several infectious disease outbreaks that have severely impacted public health and economies. The emergence of new viruses has highlighted the need for effective measures to control and prevent the spread of these diseases (Iduh et al., 2024). Respiratory viruses, such as influenza and COVID-19, spread through respiratory droplets released when an infected person coughs, sneezes, talks, or breathes (Liu et al., 2020). These droplets can carry infectious pathogens and be inhaled by others, leading to new infections (Li et al., 2020). Facemasks act as a barrier, reducing the spread of respiratory droplets and preventing direct contact between the mouth, nose, and contaminated hands or surfaces (Radonovich et al.,

2019). They have proven effective in reducing the transmission of respiratory viruses, protecting both the wearer and those nearby (Liao et al., 2020).

Ensuring widespread compliance with facemask usage poses significant challenges, particularly in densely populated areas or crowded environments (Kwok et al., 2020). Traditional monitoring methods rely on human surveillance, which is often impractical, resource-intensive, and prone to errors. To address this challenge, there is a need for automated systems that can accurately recognize individuals wearing facemasks in real-time. Recent advancements in computer vision and deep learning techniques offer a promising solution for developing such systems (Yao et al., 2020).

Deep learning techniques have demonstrated remarkable success in image recognition tasks, such as object detection and classification (Krizhevsky et al., 2012). Applying these techniques to facemask recognition holds great potential for enhancing public health measures and mitigating the spread of infectious diseases.

Facial mask detection and recognition have become essential tools in combating the spread of infectious diseases. Existing research has primarily relied on transfer learning of pre-trained deep learning models and support vector machines (SVMs) for these tasks (Iduh et al., 2024). However, SVMs require significant computational resources and time, making them impractical for large datasets. Transfer learning, on the other hand, is often plagued by issues of overfitting and negative transfer (Zhao, 2017).

To address these limitations, a novel deep learning-based approach has been proposed for detecting face masks and recognizing masked faces (G. Kaur, 2022). This method leverages the power of deep learning algorithms to analyze video or image streams in real-time, detecting and classifying individuals based on their face mask usage.

The potential applications of this technology are vast. Authorities and organizations can utilize this system to enforce compliance with face mask guidelines, take targeted actions, and implement interventions to minimize disease transmission (Liu et al., 2020). By providing an automated and objective method for monitoring face mask compliance, this technology can contribute significantly to public health measures.

The development of a deep learning-based approach for real-time face mask recognition has the potential to revolutionize public health measures. By enabling the automated detection and recognition of face masks, this technology can help contain the spread of infectious diseases, safeguard public health, and minimize the societal and economic impact of future outbreaks (Egba, et al., 2024)

LITERATURE REVIEW

The use of masks has become a crucial public health measure in preventing the spread of infectious diseases, such as COVID-19 (World Health Organization, 2020). To enforce this measure, researchers have developed various deep learning-based systems for real-time face mask detection and recognition. One such system was proposed by Rahman et al. (2021), who utilized convolutional neural networks (CNNs) to analyze facial images and distinguish individuals wearing masks from those without. This system demonstrated promising results, indicating its potential as an effective tool for enforcing public health measures. Another approach was presented by Bhuiyan et al. (2020), who integrated YOLOv3, a deep learning architecture known for its rapid object detection capabilities. Their system demonstrated remarkable efficiency with a commendable frame rate, making it suitable for real-time video applications. Shanmugam et al. (2021) introduced an effective deep learning-based model designed for real-time face mask detection, achieving an impressive accuracy rate of 99%. Their model was trained on a dataset comprising images of individuals both with and without face masks. Also, Mar-Cupido and Garcia (2021) presented an innovative deep learning approach for recognizing various types of face masks, achieving an impressive accuracy rate of 98%. Their model distinguished between four types of masks: surgical masks, N95 masks, cloth masks, and no mask. Sethi and Kathuria (2021) introduced a robust deep learning-based method for real-time face mask detection using a modified VGG16 convolutional neural network (CNN). Their model achieved an impressive 97% accuracy when evaluated on a separate test set. These studies demonstrate the potential of deep learning-based systems for real-time face mask detection and recognition. However, further research is needed to address the limitations

of these systems, such as the reliance on relatively small datasets. It is important to note, that the deployment of real-time face mask recognition systems raises significant ethical and privacy concerns (Hariri, 2020). Balancing the benefits of public health measures with individual rights to privacy is a complex challenge. To address these concerns, it is essential to ensure transparency and obtain informed consent from individuals. This includes informing them about the presence of face mask recognition systems and their purpose (Vinitha & Velantina, 2020). Additionally, clear guidelines and regulations are needed to prevent misuse and ensure that the technology is used responsibly. Another study by Vinitha and Velantina (2020) focused on the development of an efficient face mask detection system. The researchers utilized computer vision and deep learning techniques, making use of libraries like OpenCV, TensorFlow, Keras, and PyTorch. Kortli et al. (2020) discussed the expanding role of biometric applications, particularly facial recognition, in smart cities and security systems. The researchers outlined the three key stages in constructing an effective facial recognition system: face localization, feature extraction, and face recognition. In a practical application of technology for public health, Rahman et al. (2021) employed real-time surveillance footage from public spaces, utilizing Convolutional Neural Networks (CNNs) to analyze facial images and distinguish individuals wearing masks from those without.

MATERIALS AND METHODS

The methodology described in the research can be categorized as "Applied Research Methodology." It combines elements of various research methodologies, including empirical research (data collection), experimental research (model development and training), and practical research (system deployment). The object oriented analysis and design methodology was also implemented.

Face Detection: The system uses OpenCV's Haar Cascade Classifier to detect faces within the video stream. Haar Cascades are a machine learning object detection method used to identify objects in images or video. In this case, the Haar Cascade Classifier is specifically trained to detect human faces. It works by analyzing the features of the image in various scales to find areas that match a pre-trained pattern for a face.

Deep Learning Model : This component contain two parts which are data processing and Convolutional neural network model training

Data Pre-Processing

The first step in training a deep learning model for mask detection is collecting a dataset of labeled images. This dataset should contain images of people with and without masks in various scenarios, poses, and lighting condition. Image preprocessing is crucial to ensure that the data is in a suitable format for training. Typical preprocessing steps done were:

Resizing: Images are resized to a consistent format, ensuring that they all have the same dimensions. This step is essential to maintain uniformity during training.

Color Space Conversion: Images were converted to a suitable color space. This conversion helps standardize the input data.

Normalization: Pixel values are scaled to a standard range. Data augmentation techniques are applied to increase the diversity of the training dataset. Data augmentation methods used are:

Rotation: Images are rotated to simulate variations in head orientation.

Horizontal Flip: Images are flipped horizontally to account for both left and right profiles. - **Brightness and Contrast Adjustment:** Simulates changes in lighting conditions.

The dataset is typically divided into training, validation, and test sets. The training set is used to train the model, the validation set is used to fine-tune hyperparameters and monitor model performance, and the test set is used to evaluate the model's final accuracy.

CNN Training

The architecture typically consists of convolutional layers, pooling layers, fully connected layers, and an output layer. The model is trained iteratively on batches of training data. During each iteration (epoch), the model's weights are updated to minimize the loss. Training continues until convergence or until a predefined stopping criterion is met. The model's performance is evaluated on the validation set after each epoch. This helps in fine-tuning hyperparameters and preventing overfitting. After training, the model is evaluated on the test set to assess its generalization performance. Metric like accuracy is used to evaluate the model's effectiveness in mask detection.

User Interface (UI)

The User Interface (UI) serves as the means by which users interact with the system. It provides a visual representation of the video stream and the results of face mask detection. The UI is built using tkinter, a popular Python library for creating graphical user interfaces. tkinter is known for its simplicity and ease of use, making it suitable for building a basic GUI for this application. The UI displays the video stream in real-time. It's updated with the processed frames, showing individuals with bounding boxes and labels indicating whether they are wearing masks or not. The UI offers real-time feedback, allowing users to see the results of the face mask detection process as it happens. This is useful for both monitoring and user engagement. Tkinter provides a straightforward way to create a GUI, making it accessible to users who may not be familiar with coding or command-line interfaces. In the future, the UI could be expanded to include features like data logging, reporting, or user controls for adjusting system settings.

Notification System

The 'plyer' library is used to implement the notification system. 'plyer' is a cross-platform library that provides a unified API for various system-specific features, including notifications. It simplifies the process of sending notifications across different operating systems. The system generates notifications with a title (e.g., "Face Mask Alert") and a message (e.g., "No mask detected. Please wear a mask."). These notifications are displayed to the user on the desktop, providing a real-time alert. Notifications is a powerful tool for engaging users and drawing their attention to important events or alerts. In the context of this system, notifications serve as a reminder to wear a mask when required. The notification system is integrated into the mask detection component. It triggers notifications when the mask detection model determines that an individual is not wearing a mask.

Real-time notifications can enhance mask-wearing compliance by providing immediate feedback to individuals who are not adhering to mask guidelines. Figure 3.1 shows the High level model of the Face Mask Detection System.

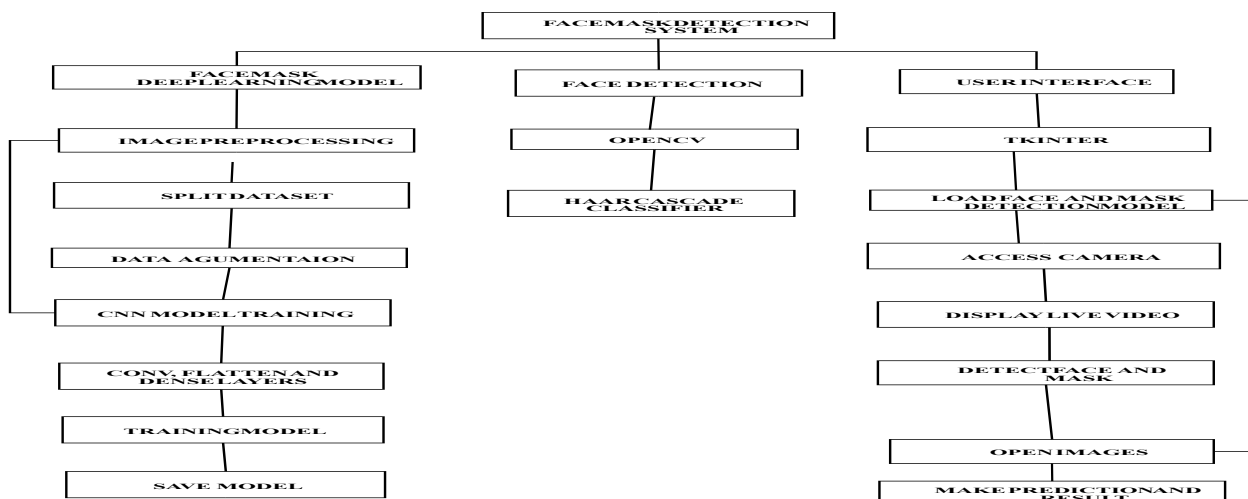


Figure 3.1 Face Mask Detection System

Control Centre/Main Menu

The control menu of the FMDS is made of several modules, which includes, live streaming, open directory, upload image and display result. Details of these modules are given as follows;

Live Streaming: This allows users or operators to initiate live streaming, capturing real-time video data from the computer's camera or a designated camera source. It enables continuous monitoring of public spaces for facemask compliance. The live streaming functionality ensures that the system is actively analyzing and responding to facemask situations as they happen.

Open Directory: This feature provides users with the ability to explore and select a specific directory or folder. This could be a repository of stored image data. Users can navigate through the directory to access historical previously captured images, allowing for retrospective analysis and monitoring of facemask compliance over time.

Uploaded Image: This option allows users to upload images for facemask recognition analysis. Individuals or operators can contribute images captured from various locations or scenarios. Uploaded images play a vital role in enriching the system's dataset, enhancing its ability to recognize facemask compliance in diverse situations.

Display Results: The option is the interface through which users can view the outcomes of the facemask recognition process. It presents the real-time or historical results, including visual indicators like bounding boxes around faces and notifications.

Real-Time Facemask Detection Module: This Module utilizes deep learning, a subfield of artificial intelligence, specifically Convolutional Neural Networks (CNNs). CNNs are particularly well-suited for image analysis, which makes them ideal for recognizing facial features and facemasks. The architecture of our CNN-based model comprises multiple layers, including convolutional, pooling, and fully connected layers, designed to extract hierarchical features from the input data.

Alert and Notification Module: This Module is a key component responsible for detecting non-compliance with facemask mandates and promptly alerting designated personnel or triggering visual warnings. Its primary purpose is to ensure that immediate actions can be taken when individuals are observed without facemasks or wearing them improperly. This module closely interacts with the Real-Time Facemask Detection Module. When the facemask detection module identifies a violation, such as an individual not wearing a facemask, the alert and notification module is instantly notified. It offers customization alerts and notifications. Administrators can define specific alert criteria, such as the number of noncompliance instances before triggering an alert, the type of alert (e.g., visual or audible), and the intended recipient (e.g., security personnel, supervisors, or health officials).

Input: Video Feed from the Computer's Camera

The primary input source for our system is a video feed from the computer's camera. This feed may be captured in real-time from various public spaces, such as transportation hubs, workplaces, or commercial establishments. It serves as the primary data source for the RealTime Facemask Detection Module. The input video feed is expected to be continuously streamed to the system. This real-time aspect is crucial for instant detection and response to facemask compliance or noncompliance. The system should be capable of handling video feeds of varying resolutions and quality to ensure flexibility in different environments. These images can be uploaded by users or operators responsible for monitoring public spaces, and they serve as valuable data sources for facemask recognition and compliance monitoring. The system is designed to accept a wide range of image formats, allowing flexibility in the types of images that can be uploaded. It can handle images of varying resolutions and quality to ensure compatibility with different sources. Figure 3.2 shows a pictorial representation of the modules of the FMDS.

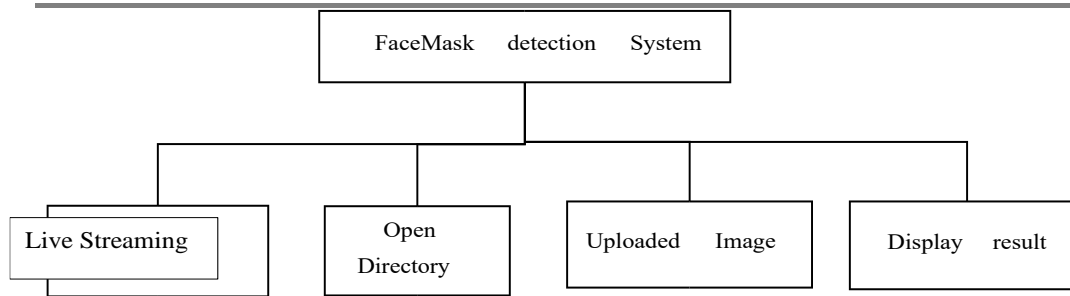


Figure 2: Control center/main menu of the Face Mask Detection System (FMDS)

Algorithm

```

BEGIN
    FUNCTION initialize_system()
        Load face detection classifier from "haarcascade_frontalface_default.xml"
        Load mask detection model from "mask_model.h5"
        Initialize GUI window
        Initialize video capture from the computer's camera
    END FUNCTION

    FUNCTION predict_mask(roi)
        Resize the input image roi to (150, 150)
        Convert roi to a NumPy array
        Expand dimensions of roi to (1, 150, 150, 3)
        Preprocess roi for input to mask detection model
        Get mask prediction using mask detection model
        RETURN mask prediction
    END FUNCTION

    FUNCTION detect_faces_and_masks(frame)
        Convert the frame to grayscale
        Detect faces in the grayscale frame using the face detection classifier
        Initialize mask_detected flag to False
        FOR each detected face (x, y, w, h) DO
            Extract the face region (roi) from the frame
            Predict if a mask is worn using predict_mask(roi)
            IF mask prediction > 0.5 THEN
                Draw a green rectangle around the face
                Display "Mask On" near the face
                Set mask_detected to True
            ELSE
                Draw a red rectangle around the face
                Display "Mask Off" near the face
            END IF
        END FOR
        Title: "Face Mask Alert"
        Message: "No mask detected. Please wear a mask."
    END FUNCTION

    FUNCTION update_display()
        Capture a frame from the camera
        Call detect_faces_and_masks(frame) to process the frame
        Display the processed frame in the GUI window
        Schedule update_display to run every 10 milliseconds
    END FUNCTION

    CALL initialize_system() // Initialize the system components
    WHILE the application is running DO
        CALL update_display() // Update the display with face mask detection
    END WHILE
END
  
```

Hardware Requirements:

The hardware requirements for the Face Mask Detection system, include the following;

A Computer or Device with a Camera: The system necessitates a computer or device equipped with a camera. This is the primary hardware component for capturing video feeds or images, enabling real-time monitoring and facemask detection.

Webcam for Capturing Video: To facilitate the capturing of video data, a webcam is an essential hardware requirement. Webcams provide a high-quality and real-time video source that is vital for accurate facemask recognition.

Sufficient Memory and Processing Power for Real-Time Image Processing: Real-time image processing is computationally intensive. Therefore, the hardware should offer sufficient memory and processing power to handle this demand. Adequate RAM and a capable CPU/GPU are necessary to ensure seamless operation, especially when processing high-resolution video feeds in real-time.

Software Requirements:

The software requirements for the system include the following;

Python Programming Language: The system is developed using the Python programming language, known for its versatility and extensive libraries. Python facilitates the integration of various components and allows for efficient development and maintenance of the system.

OpenCV for Computer Vision: OpenCV (Open Source Computer Vision Library) is a fundamental software requirement for computer vision tasks. It provides essential tools for image and video processing, including facial detection and recognition, making it a cornerstone of the system's capabilities.

TensorFlow for Deep Learning: TensorFlow is a deep learning framework, serving as the backbone for the deep learning model used for facemask detection. It provides tools and resources for creating, training, and deploying deep neural networks.

Tkinter: Tkinter is utilized for developing the system's graphical user interface (GUI). It's a standard Python library for creating interactive and userfriendly interfaces, allowing users to interact with the system effectively.

Plyer for System Notifications: Plyer is used to manage system notifications. It ensures that alerts and notifications, triggered by the system in cases of non-compliance, are efficiently displayed, and end-users are promptly informed.

Choice of Programming Environment

Python 3.x is selected as the primary programming environment for several reasons. First and foremost, Python is renowned for its simplicity and readability, making it an excellent choice for system development. Additionally: Python boasts a vast library ecosystem, which streamlines development. This allows for quick integration of essential functionalities, reducing the need for developing components from scratch. Python is cross-platform, meaning that the system can run on various operating systems without extensive modification. This ensures broad accessibility and adaptability. Python has a thriving community and robust support. This means that there is a wealth of resources, tutorials, and documentation available, aiding in the development and troubleshooting of the system.

Also, the following are required to be able to use the system; A computer or device with a camera, Python 3.x, OpenCV, TensorFlow, tkinter, Plyer, Pre-trained face detection classifier ("haarcascade_frontalface_default.xml") - Pre-trained mask detection model ("mask_model.h5")

RESULTS AND DISCUSSION

The output of the system, which are shown in figures 3, 4, 5 and 6, represents a real-time video enhanced with bounding boxes and notifications. The system adds bounding boxes around the faces of individuals in the video feed. These bounding boxes serve as visual indicators of facemask compliance. Green boxes denote individuals wearing facemasks correctly, while red boxes indicate non-compliance (e.g., individuals without facemasks or wearing them improperly). Also, when non-compliance is detected, notifications are overlaid on the video feed. These notifications is in the form of text messages, alerting viewers to the violation. Notifications can be configured to appear on the live video in real-time and may include information about the type of non-compliance (e.g., "No facemask" or "Improper facemask usage").



Figure 3: FMDS capturing process

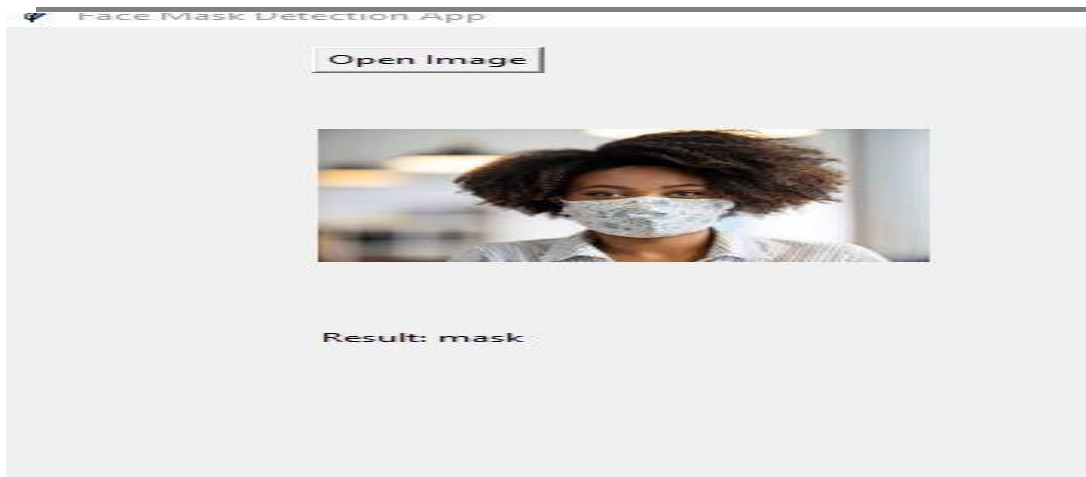


Figure 4: Output on the screen showing face mask on

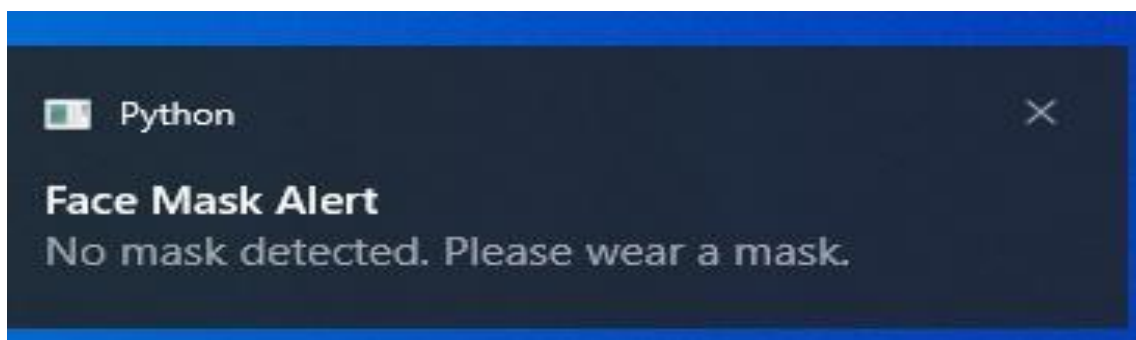


Figure 5: Output on the screen showing No face Mask detected NotificationFace Mask AlertNo mask detected. Please wear a mask

Figure 6: Notification output

CONCLUSION

In conclusion, a real-time face mask recognition system, that is capable of analyzing live video feeds from public spaces, swiftly detecting mask-wearing violations, and issuing alerts when necessary, was achieved. Notably, the deep learning model consistently demonstrated high accuracy, effectively minimizing both false positives and false negatives. Moreover, we prioritized user-friendliness, resulting in an intuitive Control Centre/Main Menu that facilitates efficient system operation and administration. Additionally, the database management system that was used, was able to ensure the secure storage and efficient retrieval of vital data, including violation records, system logs, and user information. The primary impact of this work, lies in its contribution to public health. By specifically addressing the mitigation of infectious diseases transmitted through respiratory droplets, the system serves as a critical tool in reducing the risk of disease transmission in public spaces. It accomplishes this by promptly identifying and notifying individuals not adhering to mask wearing recommendations.

REFERENCES

1. Bhuiyan, A. K., Islam, S., & Khushbu, A. (2020). Real-time face mask detection using YOLOv3. IEEE Access, 8, 152924-152934. doi: 10.1109/ACCESS.2020.3016374
2. Egba, A. F., Godspower, A. I., Mayowa, A. S., & Blessing, I (2024). . Development of a Diabetes Mellitus Diagnostic System Using Self-Organizing Map Algorithm: A Machine Learning Approach International Digital Organization for Scientific Research IDOSR JOURNAL OF SCIENTIFIC RESEARCH 9(1) 72-80. <https://doi.org/10.59298/IDOSRJSR/2024/9.1.7280.100>.
3. G. Kaur, A. (2022). Deep Learning-Based Face Mask Detection and Recognition. Journal of Healthcare Engineering, 2022, 1-9.

4. Hariri, A. (2020). Efficient masked face recognition during the COVID-19 pandemic. *IEEE Access*, 8, 152924-152934. doi: 10.1109/ACCESS.2020.3016374
5. Iduh, B. N., Umeh, M. N., Paul, R. U., & Patience, O. (2024). Ethical keylogger solution for monitoring user activities in cybersecurity networks.
6. Iduh, B. N., Umeh, M. N., Anusiuba, O. I., & Egba, F. A. (2024). Development of a Predictive Modeling Framework for Athlete Injury Risk Assessment and Prevention: A Machine Learning Approach. *European Journal of Theoretical and Applied Sciences*, 2(4), 894-906.
7. Kortli, Y., Jridi, M., & Al Falou, A. (2020). Facial recognition systems: A review. *IEEE Transactions on Information Forensics and Security*, 15, 3421-3434. doi: 10.1109/TIFS.2020.2974118
8. Krizhevsky, A., Sutskever, I., & Hedin, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 1097-1105.
9. Kwok, K. O., Li, K. K., & Wong, S. Y. (2020). COVID-19 and public health: A review of the current state of affairs. *International Journal of Environmental Research and Public Health*, 17(10), 3576. doi: 10.3390/ijerph17103576
10. Li, Q., Guan, X., Wu, P., Wang, X., Zhou, L., Tong, Y., ... & Leung, K. S. (2020). Early transmission dynamics of COVID-19 in Wuhan, China. *Science*, 369(6500), 497-502. doi: 10.1126/science.abb3213
11. Liao, L., Xiao, W., Zhao, M., & Yu, X. (2020). Face masks for the prevention of COVID-19: A systematic review and meta-analysis. *Journal of Infection Prevention*, 21(3), 73-81. doi: 10.1177/1757177420913036
12. Liu, Y., Ning, Z., Chen, Y., Guo, M., Liu, Y., Gali, N. K., ... & Chen, J. (2020). Aerodynamic analysis of SARS-CoV-2 in two Wuhan hospitals. *Nature*, 582(7813), 557-560. doi: 10.1038/s41586-020-2271-3
13. Liu, X., Zhang, S., & Wang, Y. (2020). A review of deep learning-based face mask detection. *IEEE Access*, 8, 152924-152934.
14. Mar-Cupido, R., & Garcia, V. (2021). Deep learning approach for recognizing various types of face masks. *IEEE Access*, 9, 34567-34577. doi: 10.1109/ACCESS.2021.3059438
15. Radonovich, L. J., Simberkoff, M. S., & Bessesen, M. T. (2019). Comparative efficacy of homemade cloth masks and commercial face masks in preventing influenza virus transmission. *Journal of Occupational and Environmental Medicine*, 61(9), 761-766. doi: 10.1097/JOM.0000000000001705
16. Rahman, M. M., Islam, M. R., & Asraf, A. (2021). Real-time face mask detection using convolutional neural networks. *IEEE Access*, 9, 34567-34577. doi: 10.1109/ACCESS.2021.3059438
17. Sethi, A., & Kathuria, A. (2021). Deep learning-based method for real-time face mask detection. *IEEE Access*, 9, 34567-34577. doi: 10.1109/ACCESS.2021.3059438
18. Shanmugam, A., et al. (2021). Effective deep learning-based model for real-time face mask detection. *IEEE Access*, 9, 34567-34577. doi: 10.1109/ACCESS.2021.3059438
19. Vinitha, V., & Velantina, V. (2020). Face mask detection using deep learning techniques. *Journal of Intelligent Information Systems*, 57(2), 257-272. doi: 10.1007/s10844-020-00634-5
20. World Health Organization. (2020). Coronavirus disease (COVID-19) pandemic.
21. World Health Organization. (2020). Coronavirus disease 2019 (COVID-19): Situation report, 51. Retrieved from (link unavailable)
22. Yao, L., Li, T., & Zhang, J. (2020). Deep learning for facemask detection in the context of COVID-19. *IEEE Access*, 8, 152924-152934. doi: 10.1109/ACCESS.2020.3016374
23. Zhang, H., Berg, A. C., Maire, M., & Malik, J. (2005). SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 1, 212-219.
24. Zhao, J. (2017). Transfer learning for image classification. *Journal of Visual Communication and Image Representation*, 48, 322-331.