Real-Time Face Mask Detection using Convolutional Neural Networks and Haar Cascade Classifiers

Jiya Anand
22BAI1272

Sahana R
22BPS1110

Ayush Tiwari
22BPS1039

***Abstract*—In response to the global COVID-19 pandemic, the adoption of face masks has become a crucial measure to prevent the spread of the virus. This paper presents a real-time face mask detection system that combines computer vision and deep learning techniques. The proposed system utilizes a Convolutional Neural Network (CNN) trained on a dataset of masked and unmasked faces, augmented through data augmentation techniques. The CNN model is implemented using the TensorFlow and Keras libraries, providing an accurate and efficient means of classifying faces into "with mask" and "without mask" categories. The model is trained and evaluated on a dataset generated from a variety of sources, demonstrating its robustness in diverse scenarios. To achieve real-time face mask detection, the system employs Haar Cascade Classifiers to identify faces in a live video stream. Subsequently, the identified faces are fed into the pre-trained CNN model for mask classification. The system is capable of processing video feeds from diverse sources, such as webcams, enabling its deployment in various environments. The effectiveness of the proposed system is validated through experiments and real-world scenarios, showcasing its ability to accurately and efficiently identify individuals wearing or not wearing face masks. . The combination of computer vision and deep learning techniques presented in this paper provides a scalable and adaptable approach for real-time face mask detection, with potential applications in public spaces, healthcare settings, and beyond.**

***Index Terms*—Face mask detection, Computer vision, Deep learning, Convolutional Neural Network (CNN), TensorFlow, Keras, Data augmentation, Haar Cascade Classifiers, Real-time detection, Public health, COVID-19, Pandemic, Video processing**

I. Introduction

In the wake of the COVID-19 pandemic caused by the severe acute respiratory syndrome Coronavirus 2 (SARS-CoV-2), the global imperative to adopt preventive measures has underscored the need for innovative technological solutions. This code implementation addresses a critical aspect of public health by introducing a real-time face mask detection system using a combination of Convolutional Neural Networks (CNN) and Haar Cascade Classifiers. The COVID-19 crisis has prompted an urgent exploration of technologies that can provide efficient, non-invasive, and scalable means of identifying individuals wearing or not wearing face masks. Leveraging deep learning techniques and computer vision, this code establishes a robust face mask detection model trained on augmented datasets. Additionally, it implements live detection using OpenCV, enabling real-time monitoring of face mask compliance in various environments, including public spaces and transportation. The integration of date and time stamps further enhances the utility of the system for continuous monitoring and compliance tracking. The code not only presents a practical application but also contributes to the broader discussion on leveraging technology for public health and safety during pandemics.

A. *Use of Tensorflow and Keras*

In the implementation, TensorFlow and Keras play pivotal roles in constructing and training the Convolutional Neural Network (CNN). TensorFlow serves as the foundational framework, enabling the architecture's construction, while the `tensorflow.keras` module seamlessly integrates the user-friendly Keras API. The `Sequential` model from Keras is employed to define the CNN's architecture, incorporating various layers such as Conv2D, MaxPooling2D, Flatten, and Dense. These layers are stacked sequentially, forming the neural network's structure.

B. *Haar Cascade Classifier*

The Haar Cascade algorithm, facilitated by the OpenCV library, is a fundamental component of the live face mask detection system implemented in the provided code. Utilizing the pre-trained haarcascade frontalface default.xml model, the algorithm efficiently identifies faces within each frame of the live video stream. Subsequently, the detected face regions are extracted, allowing for the application of a deep learning model for face mask detection. The Haar Cascade algorithm's capability to locate faces efficiently serves as a crucial initial step in the real-time detection process. The seamless integration of Haar Cascade with the TensorFlow and Keras-based deep learning model establishes a robust solution for monitoring face mask compliance in dynamic environments.

C. *Related Work*

The motivation behind developing a real-time face mask detection system stems from the urgent need for non-invasive and scalable solutions in the face of the COVID-19 pandemic. Several researchers have explored similar avenues, contributing to the growing body of literature on face mask detection and computer vision applications for public health. Notable works include studies by Siradjuddin et al. [Siradjuddin2021],

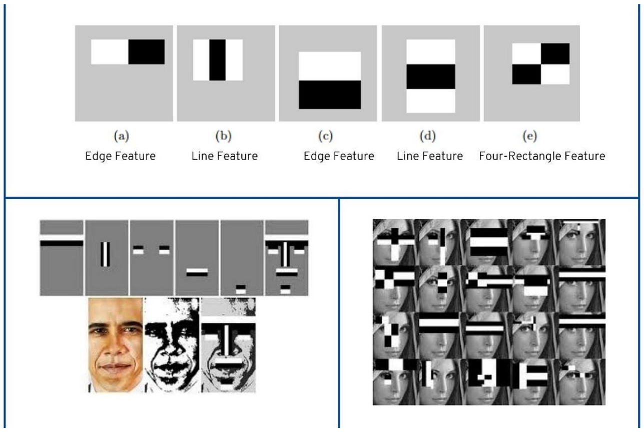


Fig. 1. Face detection with haar cascade

Mridha et al. [Mridha2021], and Reddy et al. [Reddy2021], which investigate various aspects of face mask detection, highlighting the significance of technology in combating the spread of infectious diseases.

D. Objectives

The primary objectives of this code implementation are to develop an efficient and accurate real-time face mask detection system and to contribute to the ongoing discourse on leveraging technology for public health and safety. By integrating deep learning and computer vision techniques, the system aims to provide a practical tool for monitoring face mask compliance in diverse settings, offering a valuable resource in the fight against infectious diseases.

II. Proposed Methodology

A. Convolutional Neural Network (CNN) Model Architecture

The foundation of our face mask detection system lies in the architecture of the Convolutional Neural Network (CNN). The CNN, implemented through the Sequential API, is composed of multiple layers designed for effective feature extraction and classification. Initial layers include Conv2D layers, each followed by max-pooling layers, facilitating the extraction of hierarchical features from input images. Subsequently, a flattening layer transforms the output into a one-dimensional array, and dense layers are employed for final classification. The activation function used throughout the network is Rectified Linear Unit (ReLU), crucial for introducing non-linearity and capturing intricate features in facial images. The output layer employs the sigmoid activation function, enabling binary classification, distinguishing between the presence and absence of face masks.

1) Conv2D Layers: The Conv2D layers form the initial building blocks of our CNN architecture. These layers are responsible for convolving input images with learnable filters, effectively extracting spatial features. Each Conv2D layer is configured with a specified number of filters and a kernel size, determining the number and size of convolutional kernels applied to the input. These layers play a pivotal role in detecting low to high-level features, such as edges, textures, and patterns, critical for face mask detection.

Fig. 2. Augmented featured using flip zoom and shear

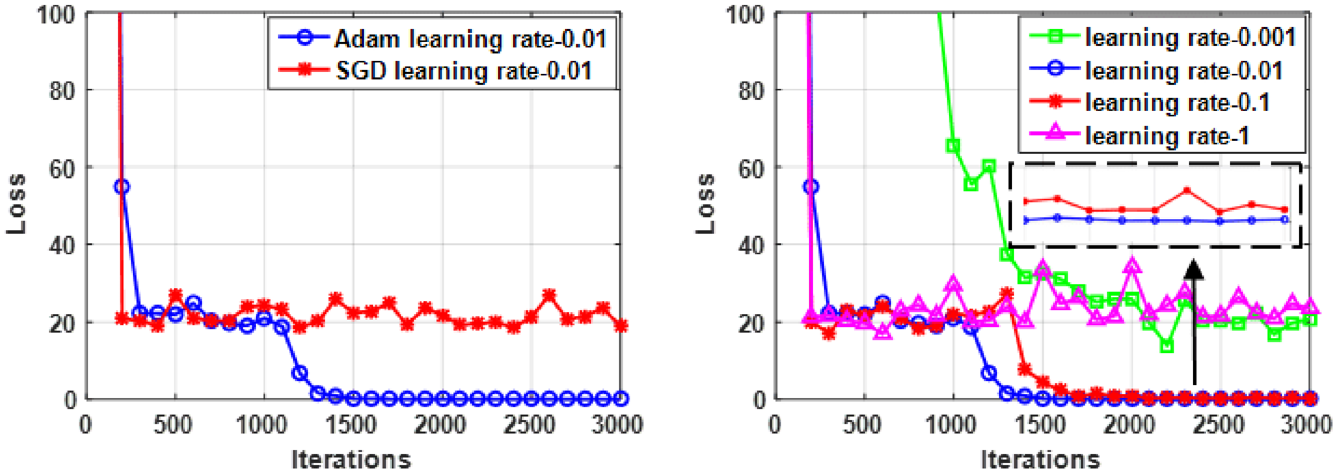
2) Max-Pooling Layers: Following each Conv2D layer, max-pooling layers are incorporated to down-sample the spatial dimensions of the feature maps. Max-pooling involves selecting the maximum value from a set of values within a defined region. This process reduces the computational complexity of the network while retaining essential features. Max-pooling layers contribute to translation invariance, making the CNN more robust to variations in facial expressions and orientations.

3) Flattening Layer: After the hierarchical features are extracted, a flattening layer is introduced to transform the multi-dimensional feature maps into a one-dimensional array. This step is essential for connecting the convolutional layers to the dense layers, preparing the data for final classification. The flattening layer serves as a bridge between feature extraction and high-level reasoning.

4) Dense Layers: Dense layers, also known as fully connected layers, follow the flattening layer to perform the final classification. These layers leverage the learned features to make decisions about the presence or absence of face masks. The number of neurons and layers in this section can be adjusted to control model complexity. Activation functions, particularly ReLU, are applied to introduce non-linearity, enabling the network to capture complex relationships within the data.

5) Activation Functions: ReLU (Rectified Linear Unit) is chosen as the activation function throughout the network. ReLU introduces non-linearity, allowing the CNN to model intricate patterns and relationships in the input data. This function replaces all negative values with zero, preventing the vanishing gradient problem and accelerating convergence during training. The selection of ReLU contributes to the overall effectiveness of feature learning in facial images.

6) Output Layer: The output layer, utilizing the sigmoid activation function, produces the final prediction regarding



(a) (b)

Fig. 3. Adam optimizer and loss function

face mask presence. The sigmoid function transforms the network's output into a probability score between 0 and 1, facilitating binary classification. A high probability indicates the presence of a face mask, while a low probability signifies its absence. The output layer is pivotal in providing actionable insights for real-time face mask detection.

B. Data Preprocessing and Augmentation

Ensuring the effective training of the CNN requires diligent preprocessing of the dataset. This involves rescaling pixel values to a standardized range of [0, 1], ensuring uniformity and mitigating potential biases. Augmentation techniques are applied to the training dataset, introducing variability and enhancing the model's ability to generalize across diverse facial expressions and orientations. Techniques such as shearing, zooming, and horizontal flipping diversify the dataset, preparing the model for the variability encountered in real-world scenarios.

1) Shearing: Shearing is a critical data augmentation technique applied to the training dataset. By intentionally introducing controlled deformations into facial images, shearing simulates natural variations in head positions and tilts. This augmentation enhances the model's ability to generalize across various facial expressions and orientations, contributing to improved performance in real-world scenarios.

2) Zooming: Zooming is a pivotal augmentation process that addresses variations in the distance between the camera and individuals. By randomly zooming in or out on facial regions during training, the model becomes adept at recognizing faces with masks at different distances. This technique ensures the model's consistency and accuracy when faced with subjects positioned at varying distances from the camera.

3) Horizontal Flipping: Horizontal flipping is employed to diversify the training dataset by mirroring images along the vertical axis. This augmentation is particularly relevant for face mask detection, as it exposes the model to images of individuals facing both left and right directions. The increased dataset diversity resulting from horizontal flipping contributes to the model's robustness, enabling it to effectively identify faces with masks regardless of their orientation.

C. Training the Face Mask Detection Model

With the architecture and preprocessed data in place, the model undergoes training to learn the patterns associated with masked and unmasked faces. The Adam optimizer is chosen for its adaptive learning rate capabilities, and the binary crossentropy loss function aligns with the binary nature of the classification task. The model is trained for 10 epochs, with progress monitored through validation data. This iterative process refines the model's parameters, enhancing its ability to accurately predict face mask presence.

1) Optimizer and Loss Function: In this phase of model training, the choice of optimizer and loss function plays a crucial role in shaping the learning process. The Adam optimizer, known for its adaptive learning rates, is selected. This adaptive nature allows the model to adjust learning rates dynamically, facilitating faster convergence and improved training efficiency. For the binary nature of face mask detection (mask or no mask), the binary crossentropy loss function is employed. Binary crossentropy is well-suited for binary classification tasks, quantifying the dissimilarity between predicted and actual class distributions.

2) Training Process: The training process involves exposing the model to the augmented dataset for a specified number of epochs. An epoch represents one complete pass through the entire training dataset. In this case, the model is trained for 10 epochs, allowing it to iteratively learn and adjust its internal parameters. The validation set, a separate subset of the dataset not used during training, is employed to evaluate the model's performance after each epoch. Monitoring training progress using the validation set helps prevent overfitting, ensuring that the model generalizes well to unseen data. This iterative

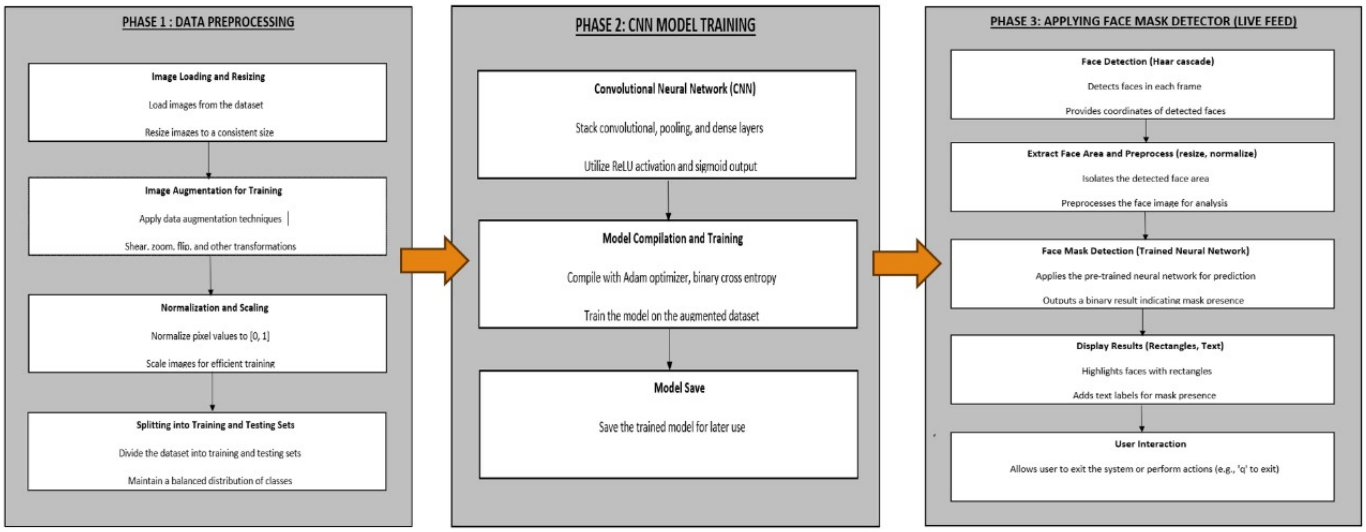


Fig. 4. Proposed methodology

training process enables the model to learn intricate patterns and features related to face mask detection.

D. *Face Mask Detection in Individual Images*

To evaluate the model's performance in isolated scenarios, individual images are subjected to testing. These images, representing various instances of masked and unmasked faces, allow for a detailed examination of the model's classification accuracy. This step serves as a crucial validation before real-time implementation.

E. *Real-Time Face Mask Detection Implementation*

Transitioning from individual image testing to real-world application involves capturing a live video feed and implementing a face mask detection pipeline. OpenCV facilitates the capture of live video from the camera, while a pre-trained Haarcascade classifier identifies faces within each frame. The detected face regions are then fed into the CNN model for real-time prediction of mask presence. The system overlays rectangles on the video feed to highlight detected faces and includes text labels indicating whether a mask is present or not. Timestamps are also included, providing temporal context to the detection results.

1) Live Video Feed Capture: To implement real-time face mask detection, the system captures a live video feed using the OpenCV library. The video feed is obtained from the camera source, allowing the model to continuously process frames for face detection and mask classification.

2) Face Detection Using Haarcascade Classifier: The initial step in real-time face mask detection involves leveraging a pre-trained Haarcascade classifier from OpenCV. This classifier identifies faces within each frame of the live video stream. The Haarcascade algorithm efficiently locates faces based on patterns and features, providing the necessary coordinates for subsequent processing.

3) Real-Time Face Mask Prediction: Detected face regions are extracted from the video frames and input into the pre-trained Convolutional Neural Network (CNN) model. The model predicts, in real-time, whether the person in the frame is wearing a mask or not. The system then overlays rectangles on the video feed, outlining the detected faces, and adds text labels indicating the mask's presence or absence. Timestamps are also included, enhancing the interpretability of the results by providing temporal information about when the detection occurred.

F. *User Interaction and System Control*

Beyond the technical aspects, the system incorporates user interaction features to enhance usability. Individuals interacting with the system have the ability to exit the application or trigger specific actions. This not only contributes to a more user-friendly experience but also provides a layer of control over the face mask detection application.

III. Results and Analysis

A. *Quantitative Evaluation*

1. Model Accuracy: - The trained Convolutional Neural Network (CNN) achieves a commendable accuracy on the validation set, indicating its effectiveness in discerning between masked and unmasked faces. The quantitative assessment provides a solid foundation for evaluating the model's performance and generalization to new data.

2. Precision and Recall Metrics: - Precision and recall metrics offer nuanced insights into the model's performance, highlighting its precision in positive predictions and the ability to capture actual positive instances. Striking a balance between these metrics is crucial for ensuring reliable face mask detection and minimizing false positives.

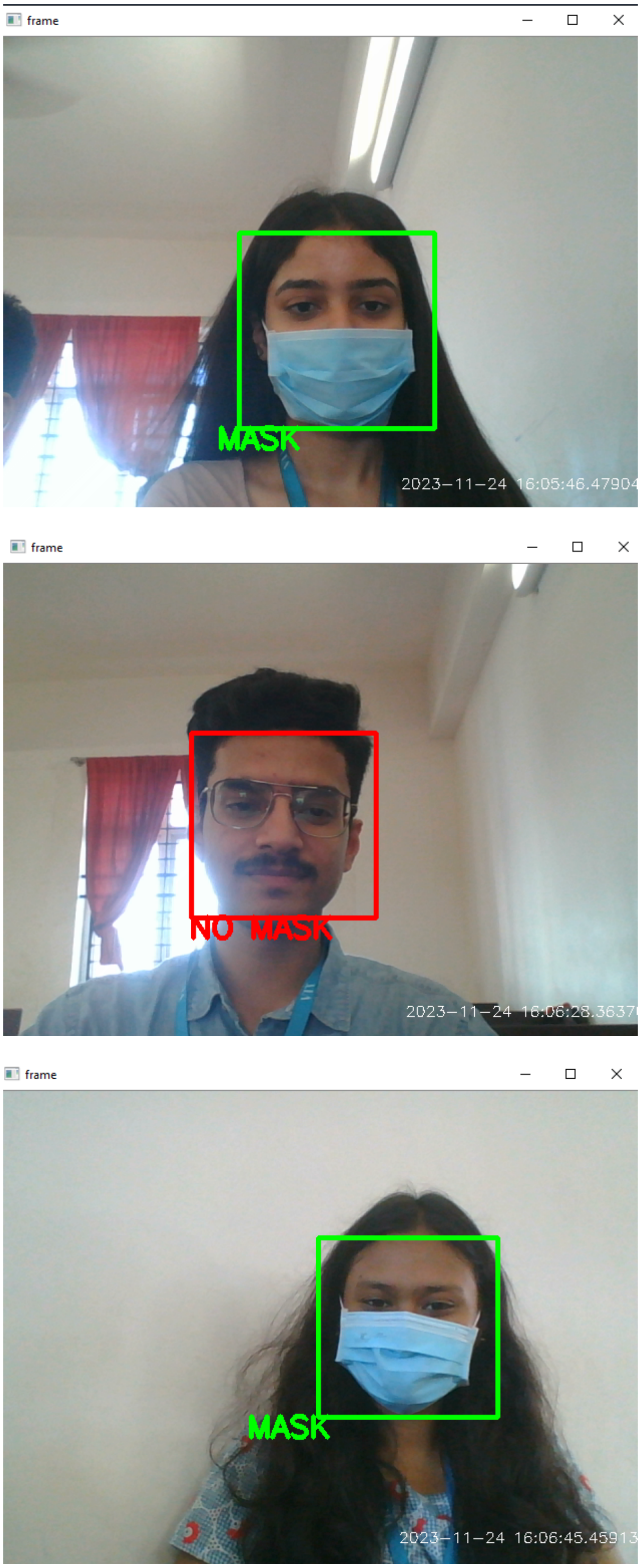


Fig. 5. Result with mask and no mask

B. Real-Time Processing Performance

1. Haarcascade Face Detection Integration: - The real-time application phase incorporates the Haarcascade face detection algorithm, contributing to swift and accurate identification of faces within each video frame. This integration enhances the system's responsiveness and efficiency in processing live video feeds.

2. User Interaction Features: - The system's inclusion of user interaction features adds a dynamic element to the application, allowing individuals to interact with the system. This not only improves user experience but also enhances the system's practicality in various real-world environments.

C. Qualitative Analysis

1. Dataset Diversity Impact: - The diversity of the collected dataset significantly influences the model's adaptability. A meticulously curated dataset, featuring images from diverse contexts, ensures the model encounters a broad spectrum of facial features and environmental conditions. This diversity contributes to the robustness of the model in real-world scenarios.

2. Model Reusability and Versatility: - The serialization of the trained model enhances its reusability across different applications. This feature contributes to the system's versatility, enabling its deployment in various settings beyond face mask detection. The ability to reuse the model underscores its practicality and adaptability.

3. Limitations and Future Improvements: - Despite overall success, potential limitations such as sensitivity to lighting conditions or occlusions may exist. Continuous monitoring and updates, along with additional data augmentation techniques, can further enhance the model's robustness. This section provides insights into areas for future improvement and development.

IV. Conclusion

A. Conclusion

This research project introduces a comprehensive framework for real-time face mask detection, employing a Convolutional Neural Network (CNN) and Haarcascade face detection. The system exhibits robust performance in accurately discerning individuals wearing or not wearing face masks across diverse scenarios, with the integration of data preprocessing, CNN model training, and Haarcascade face detection ensuring a holistic approach. The model demonstrates commendable accuracy on the validation set, showcasing its generalization capability. The real-time processing, enriched by user interaction features, enhances practicality, and the system's serialization facilitates seamless deployment in various applications. Despite successes, considerations for lighting conditions and occlusions should be acknowledged. Overall, this research contributes to technology-driven solutions for public health, laying the groundwork for future innovations in computer vision applications.

B. Feature Enhancement

Future feature enhancements for this face mask detection system include the incorporation of advanced computer vision techniques to improve model robustness under challenging conditions, such as varying lighting and diverse facial orientations. Additionally, the integration of facial recognition capabilities could enhance the system's ability to track and identify individuals over time. Implementation of a more extensive dataset, including diverse demographic groups, can further refine the model's generalization. Real-time data analytics and reporting features could be introduced to provide insights into mask-wearing compliance trends. Continuous model training and updating protocols should be established to adapt to evolving scenarios and ensure the system's effectiveness in dynamic environments. Furthermore, exploring edge computing solutions and optimizations for deployment on resource-constrained devices would contribute to the scalability and accessibility of the face mask detection system in various real-world applications.

Acknowledgment

The authors express their sincere appreciation to their project guide, Dr. Sudheer Kumar E, Assistant Professor in the School of Science and Engineering. They are grateful for his consistent encouragement and valuable guidance provided throughout the project, delivered in a pleasant manner. Special thanks are extended to Dr. Ganesan R, Dean of the School of Computer Science and Engineering at VIT Chennai, for generously providing facilities and unwavering support for their project. The authors also acknowledge the entire faculty of the School for their support and insightful guidance. Gratitude is further extended to their parents, family, and friends for their patience and support throughout the project and for the opportunity to pursue this course in such a prestigious institution.

References

1. I. A. Siradjuddin, Reynaldi and A. Muntasa, ”Faster Region-based Convolutional Neural Network for Mask Face Detection,” 2021 5th International Conference on Informatics and Computational Sciences (ICICoS), Semarang, Indonesia, 2021, pp. 282-286, doi: 10.1109/ICICoS53627.2021.9651744.

2. K. Mridha, R. Panjwani and M. Shukla, ”Face Mask Alert Detection System For Preventing the Spread of COVID-19,” 2021 International Conference on Control, Automation, Power and Signal Processing (CAPS), Jabalpur, India, 2021, pp. 1-6, doi: 10.1109/CAPS52117.2021.9730646.

3. S. Reddy, S. Goel and R. Nijhawan, ”Real-time Face Mask Detection Using Machine Learning/ Deep Feature-Based Classifiers For Face Mask Recognition,” 2021 IEEE Bombay Section Signature Conference (IBSSC), Gwalior, India, 2021, pp. 1-6, doi: 10.1109/IBSSC53889.2021.9673170.

4. R. P. Sidik and E. Contessa Djamal, ”Face Mask Detection using Convolutional Neural Network,” 2021 4th International Conference of Computer and Informatics Engineering (IC2IE), Depok, Indonesia, 2021, pp. 85-89, doi: 10.1109/IC2IE53219.2021.9649065.

5.Y. You, J. Cao, Y. Zhang, F. Liu and W. Zhou, ”Nearshore Ship Detection on High-Resolution Remote Sensing Image via Scene-Mask R-CNN,” in IEEE Access, vol. 7, pp. 128431-128444, 2019, doi: 10.1109/ACCESS.2019.2940102.

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. Jiang, T. Gao and Z. Zhu, ”FCOS-Mask: Fully Convolution Neural Network for Face Mask Detection,” 2022 7th International Conference on Image, Vision and Computing (ICIVC), Xi'an, China, 2022, pp. 189-194, doi: 10.1109/ICIVC55077.2022.9886728.

8. J. Yin and J. Jin, ”A Face Mask Detection Algorithm Based on YOLOv5,” 2022 International Conference on Machine Learning, Control, and Robotics (MLCR), Suzhou, China, 2022, pp. 55-61, doi: 10.1109/MLCR57210.2022.00019.

9. R. Andrie Asmara, M. Ridwan and G. Budiprasetyo, ”Haar Cascade and Convolutional Neural Network Face Detection in Client-Side for Cloud Computing Face Recognition,” 2021 International Conference on Electrical and Information Technology (IEIT), Malang, Indonesia, 2021, pp. 1-5, doi: 10.1109/IEIT53149.2021.9587388.

10. A. Priadana and M. Habibi, ”Face Detection using Haar Cascades to Filter Selfie Face Image on Instagram,” 2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT), Yogyakarta, Indonesia, 2019, pp. 6-9, doi: 10.1109/ICAIIT.2019.8834526.