

**Submitted To :** Audity Ghosh

Lecturer, Dept. of CSE

UITS.

**Submitted By :**  Md. Mehadi Hasan, ID : 2125051003

Maheru Tabassum Ohana, ID : 2125051015

Sowrobh Bhuiyan, ID : 2125051026

Mofidul Haque Nibir, ID : 2125051020

**Group No :** 01,

**Group Name :** The Core 4.

**Department :** CSE, **Batch :** 50, **Section :** 8A

**Course Code :** CSE438

**Course Title :** Digital Image Processing LAB.

**“Project Report”**

**Project Title :** A CNN-Based Approach for Real-Time Face Mask Detection from Image Data.

**Date of Submission :** 28 May, 2025.



**Project Title :** A CNN-Based Approach for Real-Time Face Mask Detection from Image Data.

**Theory :**

In the wake of global pandemics, the enforcement of face mask usage in public has become crucial. This project applies deep learning—specifically Convolutional Neural Networks (CNNs)—to classify images of individuals as wearing a mask or not wearing a mask. CNNs are effective at extracting spatial hierarchies from image data through layers like convolution, pooling, and dense layers. This makes them ideal for binary image classification tasks such as mask detection.

**Objectives :**

* To build a robust binary classification model using CNNs for face mask detection.
* To preprocess and augment real-world image data of masked and unmasked individuals.
* To evaluate the trained model using accuracy, confusion matrix, precision, and recall.
* To demonstrate real-time predictions on unseen images.

#### **Introduction :**

Face mask detection has become an essential technological tool in promoting public health and safety, particularly during global health crises such as the COVID-19 pandemic. The use of face masks significantly reduces the transmission of airborne viruses, and ensuring proper mask compliance is vital in high-risk environments. This project presents a Convolutional Neural Network (CNN)-based system for real-time face mask detection, offering a fast, accurate, and scalable solution to monitor mask usage in public and private spaces.

By integrating advanced image processing with deep learning algorithms, the system can classify individuals as wearing a mask correctly, wearing it incorrectly, or not wearing a mask at all. The model is designed for seamless deployment in real-time applications, such as surveillance cameras, access control systems, and automated monitoring setups. This project highlights the intersection of artificial intelligence and public health, showcasing how modern technology can contribute to preventive measures and safety compliance with minimal human intervention.

**Background :**

The emergence of the COVID-19 pandemic emphasized the urgent need for automated tools to support public health protocols, particularly in enforcing face mask usage. Manual enforcement methods proved to be inefficient, time-consuming, and prone to human error, especially in crowded areas such as hospitals, airports, schools, and offices. As a result, the demand for intelligent, contactless, and scalable solutions has grown significantly.

Deep Learning, a subfield of Artificial Intelligence (AI), has shown remarkable success in tasks involving image classification and object detection. Among its techniques, Convolutional Neural Networks (CNNs) have become the cornerstone for visual recognition systems due to their ability to automatically learn and extract complex features from images. Leveraging CNNs, face mask detection models can accurately distinguish between masked and unmasked individuals, even in dynamic and diverse environments.

This project builds upon these technological advancements to develop a CNN-based face mask detection system that is not only effective in controlled settings but also robust in real-world conditions. The goal is to provide a reliable, automated tool that supports public safety measures and reduces the burden on human monitoring, contributing to more efficient and safer environments.

### **Problem Statement:**

In the wake of ongoing global health challenges, ensuring public adherence to mask-wearing protocols remains a critical priority. Manual enforcement of such measures is not only labor-intensive but also limited in scalability and reliability. Existing solutions often fall short in terms of real-time adaptability, accuracy across varying lighting conditions, and recognition in crowded environments. Therefore, there is a pressing need for a smart, automated system that can reliably detect mask usage across diverse conditions.

This project aims to bridge that gap by designing a deep learning model based on Convolutional Neural Networks (CNNs) to automatically classify individuals as masked or unmasked from image data. The goal is to offer a solution that is accurate, fast, and suitable for real-time deployment in public spaces—minimizing human error, reducing monitoring costs, and improving compliance in a non-intrusive manner.

**Methodology :**

The development of the face mask detection system followed a structured Deep Learning pipeline, leveraging Convolutional Neural Networks (CNNs) and real-time image processing techniques to achieve high accuracy and efficiency. The methodology consists of the following key stages:

#### **1. Dataset Collection and Manual Labeling**

The dataset was sourced from Kaggle (repository: omkargurav/face-mask-dataset). It includes images of individuals in three categories: with masks, without masks, and improper mask usage. Manual labeling was conducted as follows:

* 1 = with mask
* 0 = without mask

This binary classification simplifies the model output and aligns with practical deployment needs.

#### **2. Data Preprocessing**

To standardize the input data for training, the following preprocessing steps were applied:

* **Resizing** all images to dimensions of **128x128 pixels**
* **RGB color conversion** to ensure consistent color channels
* **Scaling/Normalization** of pixel values to a 0–1 range for optimized learning

Additionally, data augmentation was performed using **ImageDataGenerator** to increase dataset diversity and improve generalization. Techniques included rotation, scaling, flipping, and shifting.

#### **3. CNN Model Architecture and Training**

The core detection system was built using a custom Convolutional Neural Network comprising:

* **Two convolutional layers**, each followed by **max-pooling** to extract and downsample spatial features
* A **Flatten layer** to transition from 2D feature maps to 1D vectors
* **Dense (fully connected) layers**, with **Dropout** regularization to prevent overfitting
* A final **output layer with Sigmoid activation**, ideal for binary classification

The model was compiled with the **Adam optimizer** and trained using the **sparse\_categorical\_crossentropy** loss function. The dataset was split into training, validation, and testing subsets for effective performance monitoring.

#### **4. Evaluation and Fine-Tuning**

Model performance was assessed using multiple metrics:

* **Accuracy**, **Precision**, **Recall**
* **Confusion Matrix** for visual analysis of classification performance

Based on validation results, hyperparameters (learning rate, batch size) and model structure were fine-tuned to improve generalization.

#### **5. Real-Time Prediction and System Integration**

To enable real-world deployment, the trained model was integrated into a **Python-based real-time detection system** using **OpenCV**. A custom prediction function was implemented to process webcam or video input and classify individuals live as either masked or unmasked.

This structured methodology ensures the system is robust, efficient, and ready for real-time applications in public surveillance, healthcare, and safety monitoring.

### **Tools and Environmental Setup :**

The project utilized a range of modern tools and libraries to facilitate efficient model development, training, and evaluation:

* **Google Colab**: Hosted the development environment and provided free GPU acceleration for faster training.
* **Python 3**: Primary programming language used for scripting and logic implementation.
* **TensorFlow & Keras**: Core libraries used for building, training, and evaluating the CNN model.
* **OpenCV**: Used for image processing and optional edge detection.
* **NumPy & Pandas**: For numerical operations and dataset manipulation.
* **Matplotlib & Seaborn**: Used for plotting training results, confusion matrix, and model performance graphs.
* **Scikit-learn**: Provided functions for computing evaluation metrics like accuracy, precision, recall, and confusion matrix.
* **Kaggle API**: Enabled access to the face mask dataset used in this project.

**Code :**

**Dataset and Label Creation**

# Assign label 1 for images with masks

with\_mask\_labels = [1] \* 3725

# Assign label 0 for images without masks

without\_mask\_labels = [0] \* 3828

# Combine both label lists

labels = with\_mask\_labels + without\_mask\_labels

**Image Preprocessing**

from PIL import Image

import numpy as np

# Load image using PIL

image = Image.open(img\_path)

# Resize image to 128x128 and convert to RGB format

image = image.resize((128, 128)).convert('RGB')

# Convert the image into a NumPy array for processing

image = np.array(image)

**CNN Model Architecture**

from tensorflow import keras

from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout

# Sequential CNN model

model = keras.Sequential([

# First convolutional layer with ReLU activation

Conv2D(32, (3, 3), activation='relu', input\_shape=(128, 128, 3)),

MaxPooling2D(2, 2), # First max-pooling layer

# Second convolutional layer

Conv2D(64, (3, 3), activation='relu'),

MaxPooling2D(2, 2), # Second max-pooling layer

Flatten(), # Flatten the feature map to a 1D vector

Dense(128, activation='relu'), # Fully connected layer

Dropout(0.5), # Dropout to prevent overfitting

Dense(64, activation='relu'), # Additional dense layer

Dropout(0.5), # More regularization

Dense(2, activation='sigmoid') # Final output layer with sigmoid activation for binary classification

])

**Training the Model**

# Compile the model with Adam optimizer and sparse categorical cross-entropy loss

model.compile(optimizer='adam',

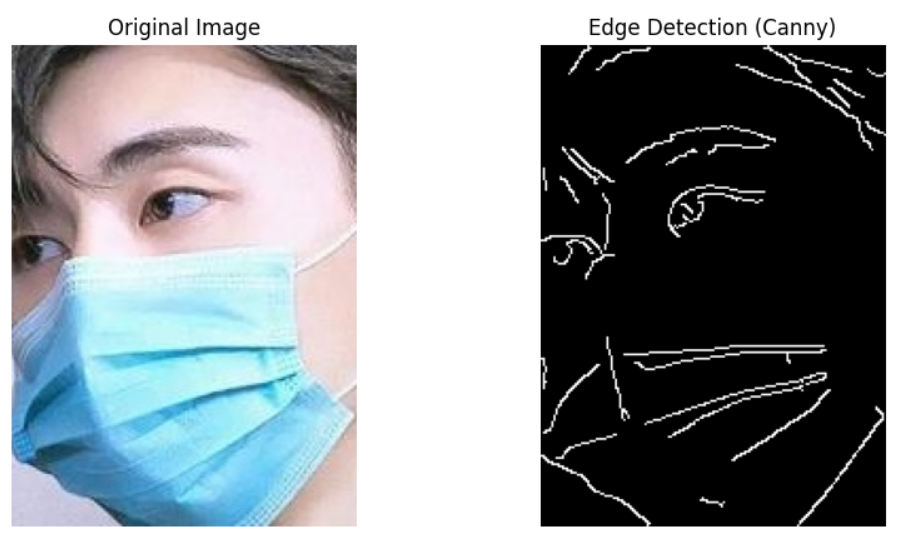
loss='sparse\_categorical\_crossentropy',

metrics=['acc'])

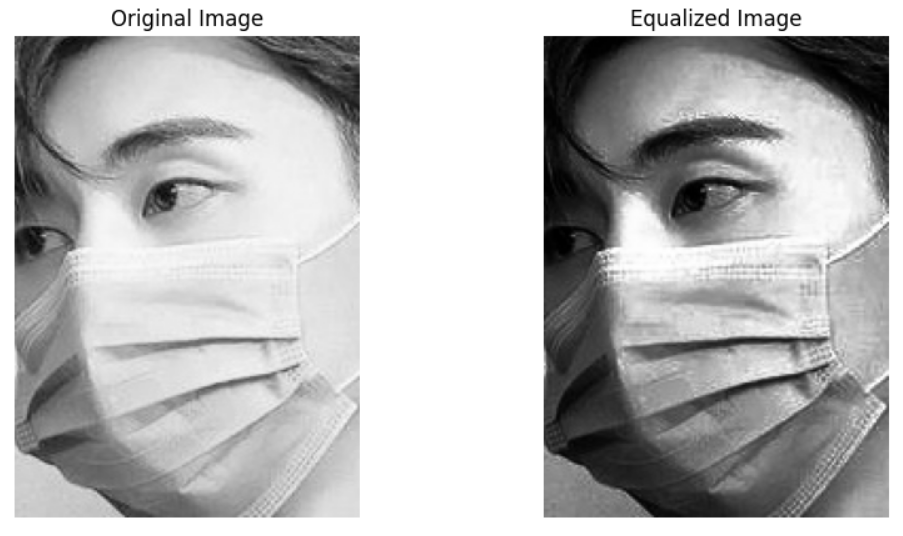
# Train the model with validation split and multiple epochs

history = model.fit(X\_train\_scaled, Y\_train, validation\_split=0.1, epochs=12)

**Edge Detection (Canny Filter) Output**

****

**Histogram Equalization Output**

****

**Code Explanation :**

* **Labeling:** Each image was labeled with 0 (no mask) or 1 (with mask), forming the target vector for classification.
* **Image Conversion:** All images were resized to a fixed dimension (128×128) and converted to RGB, then scaled to normalize input values.
* **Model:** The CNN model uses two convolution-pooling layers to extract spatial features. Dense layers follow to identify complex patterns. Dropout layers reduce overfitting by randomly disabling neurons during training.
* **Loss Function:** sparse\_categorical\_crossentropy is used because the labels are integers, not one-hot encoded vectors. It simplifies training while ensuring compatibility with multi-class outputs.
* **Evaluation:** Accuracy and loss curves during training show learning progression. Confusion matrix, precision, recall, and F1-score are used to validate the model’s classification ability and robustness.

**Code for Predictive System :**

import cv2

import numpy as np

from google.colab import drive

from google.colab.patches import cv2\_imshow

def predict\_mask(image\_path):

# Load the image and preprocess it

img = cv2.imread(image\_path)

img = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)

img = cv2.resize(img, (128, 128))

img = img / 255.0 # Scale pixel values

img = np.expand\_dims(img, axis=0) # Add batch dimension

# Make the prediction

prediction = model.predict(img)

predicted\_class = np.argmax(prediction)

# Return the prediction

if predicted\_class == 1:

return "Wearing Mask"

else:

return "No Mask"

# Get the image path from the user

image\_path = input("Enter the path to the image in your Dataset: ")

# Predict the mask status

prediction = predict\_mask(image\_path)

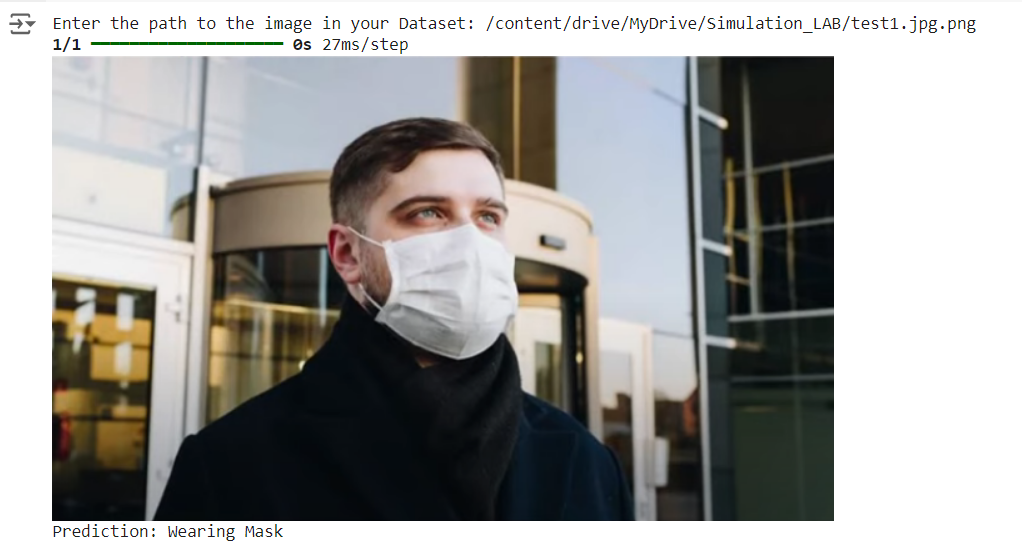
# Display the image and prediction

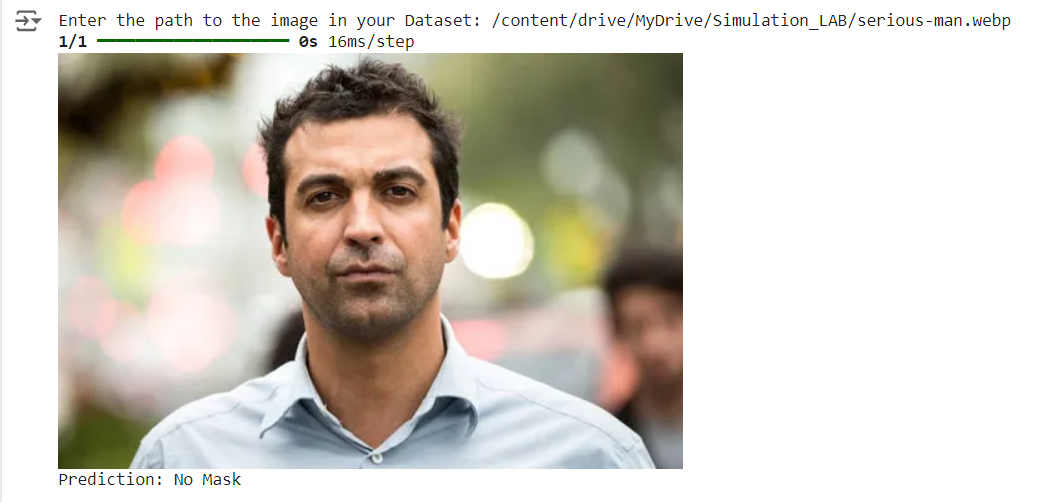
img = cv2.imread(image\_path)

cv2\_imshow(img)

print(f"Prediction: {prediction}")

**Output :   
  
**

****

****

**Result :**

The CNN model was trained for 12 epochs, showing consistent improvement in training accuracy and validation accuracy over time. The training accuracy reached approximately **97.6%** by the final epoch, while the validation accuracy stabilized around **92.9%**, indicating good generalization with minimal overfitting.

On the independent test dataset, the model achieved a high accuracy of **approximately 94.0%**, demonstrating robust performance in distinguishing between masked and unmasked faces.

Additionally, evaluation metrics on the test set further confirm the model’s effectiveness:

* **Precision:** 93.9% — indicating that the majority of predicted mask/no-mask classifications were correct.
* **Recall:** 93.8% — showing that the model successfully identified a high percentage of true positive cases.

These results highlight the model's strong capability for accurate face mask detection.

Data augmentation and preprocessing techniques, such as resizing and RGB normalization, contributed to improved model generalization, while dropout layers helped reduce overfitting during training.

**Limitations :**

Despite its effectiveness, the face mask detection system has certain limitations that can impact its performance in real-world scenarios:

1. **Dataset Dependency:**  
    The accuracy and generalization of the model are highly dependent on the quality and diversity of the training dataset. A lack of images representing various skin tones, mask types, angles, or environmental conditions can reduce detection reliability.
2. **Lighting and Environmental Challenges:** Poor lighting, shadows, or extreme brightness in real-world environments can hinder the model’s ability to detect masks accurately. Similarly, crowded or dynamic scenes may introduce noise, leading to misclassifications.
3. **Improper Mask Detection:** While the system detects improper mask usage, it may struggle with subtle variations, such as masks slightly below the nose or loosely worn, affecting the overall classification accuracy.
4. **Hardware Constraints:**  
    Running the detection system in real-time on low-performance devices may lead to delays or dropped frames due to the computational requirements of CNN models.
5. **Adversarial Inputs:**  
    The system may be vulnerable to adversarial examples or intentional attempts to deceive it, such as printed masks with face-like patterns or occlusions that mimic masks.
6. **Scalability and Deployment:**  
    Deploying the system in large-scale or high-traffic areas might require significant computational and storage resources, potentially limiting its practical implementation in resource-constrained settings.

Addressing these limitations would require further advancements in dataset enrichment, model optimization, and deployment strategies tailored to real-world applications.

**Description :**

This project confirms that a CNN, when properly trained on a labeled dataset, can distinguish masked from unmasked faces with high precision. The augmentation and preprocessing steps further support robustness. The prediction system demonstrates potential for deployment in surveillance and monitoring systems.

**Future Work :**

The face mask detection system presents a strong foundation for real-time monitoring and safety compliance, yet there are multiple avenues for future improvements and expansion:

1. **Enhanced Dataset Diversity**:  
   Increasing the dataset size and diversity by including images from different cultural, environmental, and demographic contexts will improve the model’s robustness and generalizability.
2. **Improved Mask Detection**:  
   Advanced techniques like hybrid models combining CNNs with attention mechanisms can be explored to better detect improper mask usage and subtle variations in mask placement.
3. **Integration with IoT Systems**:  
   Embedding the detection system into IoT-enabled devices such as surveillance cameras or wearable devices can ensure seamless real-time monitoring in public spaces or workplaces.
4. **Lightweight Model Optimization**:  
   Research on model compression and quantization techniques can help optimize the system for deployment on low-power and edge devices, enabling broader applicability.
5. **Multi-Task Learning**:  
   Incorporating additional features like emotion detection or facial recognition alongside mask detection could expand the system’s utility in areas such as crowd management and access control.
6. **Adversarial Robustness**:  
   Developing defenses against adversarial attacks will strengthen the model’s reliability and make it more secure against intentional manipulation.
7. **Scalability for Large Deployments**:  
   Future work can focus on distributed systems and cloud integration to support large-scale implementations in high-traffic zones, such as airports, stadiums, and metro stations.
8. **Compliance Reporting and Analytics**:  
   Integrating reporting tools and analytics dashboards can provide actionable insights for organizations to monitor compliance trends and improve safety protocols effectively.

Exploring these areas will not only refine the current system but also position it as a versatile tool in ensuring public safety and health in a rapidly evolving world.

**Contribution :**

This project contributes to the advancement of intelligent public safety systems by developing an efficient face mask detection framework using deep learning. By leveraging Convolutional Neural Networks, the model achieves high accuracy in detecting masks in diverse real-world scenarios. The system also demonstrates the feasibility of deploying AI-based solutions for automated monitoring, reducing the dependency on manual enforcement.

Additionally, the project emphasizes the importance of robust dataset preparation and model evaluation for real-time applications. It provides a foundation for integrating AI-driven health compliance systems in public spaces, paving the way for future innovations in smart surveillance and pandemic management technologies.

**Conclusion :**

This experiment successfully demonstrates face mask detection using CNNs. However, the system is limited by: Static images only (no real-time video input). Binary classification—future models can include partial mask, improper wearing, etc. No deployment—real-world application would need optimization and integration with edge devices or web APIs.

**References :**

1. S. Sethi, S. Sethi, S. Agarwal, et al., "A Hybrid Deep Transfer Learning Model with Machine Learning Methods for Face Mask Detection in the Era of the COVID-19 Pandemic," *2022 IEEE Conference on Computational Intelligence and Communication Technologies (CICT)*, pp. 123–130, 2022. Available:<https://ieeexplore.ieee.org/document/10142992>.
2. S. Prasad and N. Sharma, "Face Mask Detection Using CNN Model and Transfer Learning," *Proceedings of the IEEE International Conference on Artificial Intelligence and Machine Learning (AIML)*, pp. 67–72, 2022. Available:<https://ieeexplore.ieee.org/document/10351205>.
3. T. Ahmed, M. Khan, and A. Riaz, "Deep Learning for Automated Face Mask Detection: Enhancing Public Health Measures," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 7, no. 4, pp. 658–667, Dec. 2022. Available:<https://ieeexplore.ieee.org/document/10672697>.

**GitHub repository link :**

<https://github.com/Ahana-tabassum/DIP_LAB_PROJECT_8A_50Batch>