# Project Report: Face Mask Detection with Live Alert System

## 1. Introduction

The COVID-19 pandemic highlighted the importance of wearing face masks in public spaces to prevent the spread of the virus. Manual enforcement of mask-wearing can be tedious and error-prone. This project aims to automate the detection of whether a person is wearing a mask using computer vision and deep learning techniques. The system runs in real-time using webcam input and can optionally be accessed via a web interface.

## 2. Abstract

This project implements a **real-time face mask detection system** using a Convolutional Neural Network (CNN). It integrates OpenCV for webcam access and face detection using Haar cascades. TensorFlow/Keras is used for model training, while Flask powers a simple web interface for live streaming. The model is trained on a dataset of facial images with and without masks. Once trained, the system can predict in real-time whether a detected face is wearing a mask or not, displaying a visual alert (colored bounding boxes and labels).

## 3. Tools Used

- **Programming Language:** Python
- **Libraries:** OpenCV, TensorFlow, Keras, NumPy, Matplotlib, Pandas, scikit-learn
- **Web Framework:** Flask (for web app)
- **Face Detection:** Haar Cascade Classifier (`haarcascade_frontalface_default.xml`)
- **IDE:** VS Code
- **Version Control:** Git & GitHub

## 4. Steps Involved in Building the Project

### Step 1: Data Preparation

- Downloaded datasets of images of people with and without masks from Kaggle.
- Organized into two folders: `with_mask/` and `without_mask/`.

### Step 2: Image Preprocessing

- Resized images to 100×100 pixels.

- Normalized pixel values (scaled between 0 and 1).
- Labeled data: 0 for with-mask, 1 for without-mask.

**Step 3: Model Building & Training**

- Used Keras to build a CNN with convolutional, pooling, dropout, and dense layers.
- Trained the model for 5 epochs with validation split.
- Achieved over **93% accuracy** on validation set.
- Saved the trained model as `mask_detector_model.h5`.

**Step 4: Real-Time Detection Script**

- Integrated Haar cascade face detection using OpenCV.
- Loaded trained model and predicted mask status on webcam input.
- Displayed bounding boxes with labels ("With Mask 😷" / "No Mask 😷") in real-time.

**Step 5: Flask Web App**

- Created a Flask server (`app.py`) to stream webcam video with live prediction overlay.
- Developed a simple HTML template to view predictions in a browser.

**Step 6: Deployment & Documentation**

- Created `README.md` with setup instructions.
- Added `requirements.txt` for environment setup.
- Pushed codebase to GitHub.

---

## 5. Conclusion

This project demonstrates the application of computer vision and deep learning in solving real-world problems. The face mask detection system effectively identifies whether individuals are wearing masks in real-time. It is scalable and can be integrated into CCTV systems, kiosks, or entrances in public places. With slight modifications, the project can be expanded to detect other types of PPE (Personal Protective Equipment) as well.