**Face Mask Detection with Live Alert System**

**Name**: Yagyesh Bhardwaj  
**Tools Used**: Python, OpenCV, TensorFlow, Keras, VS Code  
**Date**: 7th July, 2025 – 25th July, 2025  
**Project Type**: Machine Learning (AI)

**Submission Date:** 26th July, 2025

### What is this project about?

This project helps check if a person is wearing a **face mask** or not. It works in real-time using the **webcam**. If someone is **not wearing a mask**, it shows a warning or red box.

This system can be used in schools, offices, shops, or public places to keep everyone safe.

### Dataset (Images)

I used a small dataset of images:

* 171 pictures with **mask**
* 100 pictures **without mask**

All images were resized to 100x100 pixels. I used 80% of the data to train the model and 20% to test it.

### How it works (Step by Step)

1. **Face Detection**: First, it finds faces using OpenCV (Haar Cascade).
2. **Prediction**: Then, it sends the face to the trained AI model.
3. **Check**: The model tells if the face has a **mask** or **no mask**.
4. **Result**: Shows green box for mask, red box for no mask.

### Model Design and Training

* I made a CNN (Convolutional Neural Network).
* It learns to understand mask vs no mask.
* I used layers like Conv2D, MaxPooling2D, Flatten, Dense.
* I trained it until it got **accuracy more than 80%(Epoch 15)** .

### Results

* If I wear a mask: green box shows.
* If I don’t wear a mask: red box with alert.
* It works in real-time using the webcam.

### Problems faced

* Webcam was slow sometimes.
* Model didn’t work well with poor light.
* Small number of images made training a bit hard.

### Future Improvements

* Add sound alert or voice message
* Use more images for better accuracy
* Make it work faster using a better model
* Add email or message alert if someone has no mask

### Final Words

This was a great learning project. I learned how to:

* Use Python and OpenCV
* Train a machine learning model
* Use webcam to detect things live

This project can be made even better in the future.

### References

* opencv.org
* tensorflow.org
* kaggle.com