EAI6010 Assignment Module 5: Face Mask Detection Microservice

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*\*\*\* Assignment submission by Arjun Shrivatsan  
\*\*\*\* EAI 6010 - Assignment No: Module 5 - Face Mask Detection Microservice  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Use Case

Face mask detection has become an essential tool in the wake of the COVID-19 pandemic. It enables public spaces, transportation services, and organizations to enforce safety measures effectively. This microservice aims to identify whether individuals are wearing a mask from an uploaded image. It simplifies integration into web or mobile applications and allows real-time mask compliance monitoring.

# Benefits of the Use Case

1. \*\*Public Health Safety\*\*: Automating mask detection helps enforce pandemic protocols and reduce virus spread.  
2. \*\*Real-Time Application\*\*: Easily integrable into surveillance systems, kiosks, or access control systems.  
3. \*\*Scalability\*\*: Deployed as a microservice, it can scale horizontally across distributed systems.  
4. \*\*Developer Friendly\*\*: Built using FastAPI, Dockerized, and API-based — making it developer-ready.  
5. \*\*Platform Independent\*\*: Containerization ensures platform independence and quick deployment.

# How We Made It a Service

The original model was created using OpenCV for face detection and a machine learning model for classification. To convert it into a service, the model logic was modularized using Python scripts. A FastAPI application was created with endpoints for image upload and prediction. The app was containerized using Docker, including all dependencies for reproducibility. Once deployed, this microservice can be invoked via REST API using HTTP POST requests.  
  
Deployment platforms like Render or DockerHub + Railway can be used to host the service continuously.

# Sample Input and Output

Endpoint: `/predict`  
Method: `POST`  
Input: JPEG or PNG image file  
Output:  
{  
 "prediction": "Mask",  
 "confidence": 0.95  
}

# References

Sajjad, M., Khan, S., & Muhammad, K. (2020). A robust deep learning-based face mask detection system. Journal of Healthcare Engineering, 2020. https://doi.org/10.1155/2020/8858987

FastAPI. (2024). FastAPI framework, high performance, easy to learn, fast to code, ready for production. https://fastapi.tiangolo.com/

Docker Inc. (2024). Docker Documentation. https://docs.docker.com/