Face Mask Detection System

An AI-powered face mask detection system using deep learning and computer vision to promote COVID-19 safety compliance.

🎯 Overview

This project uses Convolutional Neural Networks (CNNs) to detect whether individuals in images are wearing face masks. The system combines face detection (Haar Cascades) with deep learning classification to provide real-time analysis with visual feedback.

✨ Features

- **Real-time Detection**: Identifies multiple faces in a single image
- **High Accuracy**: CNN-based classification with confidence scores
- **Visual Feedback**: Annotated images with bounding boxes and labels
- **Compliance Monitoring**: Calculates mask-wearing compliance rates
- **User-Friendly Interface**: Clean Streamlit web application
- **Detailed Statistics**: Per-image analysis with summary metrics

📁 Project Structure

```
Face-Mask_Detection/
├── src/                          # Source code
│   ├── __init__.py              # Package initialization
│   ├── config.py                # Configuration settings
│   ├── detector.py              # Core detection logic
│   ├── ui.py                    # UI components
│   └── utils.py                 # Utility functions
├── models/                       # Model files
│   └── face_mask_detection_model.h5
```

```
├── notebooks/                    # Jupyter notebooks
│   └── Educational_Notebook.ipynb
├── data/                         # Sample data (optional)
├── docs/                         # Documentation
│   └── ARCHITECTURE.md           # Architecture documentation
├── app.py                        # Main application
├── requirements.txt              # Dependencies
├── .gitignore                    # Git ignore file
└── README.md                     # This file
```

🚀 Quick Start

Prerequisites

- Python 3.8 or higher
- pip package manager

Installation

1. **Clone the repository**

```
git clone
cd Face-Mask_Detection
```

2. **Install dependencies**

```
pip install -r requirements.txt
```

3. **Run the application**

```
streamlit run app.py
```

4. **Open in browser**
    - The app will automatically open at `http://localhost:8501`

🎓 Usage

1. **Launch the application** using the command above
2. **Upload an image** with one or more faces
3. **View results** including:
    - Annotated image with bounding boxes
    - Mask/No Mask labels with confidence scores
    - Statistics (total faces, masked, unmasked)
    - Compliance rate

🏗️ Architecture

The system consists of three main components:

1. **Face Detection**: Haar Cascade classifier identifies faces
2. **Preprocessing**: Detected faces are normalized (128x128 RGB)
3. **Classification**: CNN model predicts mask presence

For detailed architecture information, see ARCHITECTURE.md.

📊 Model Details

- **Architecture**: Convolutional Neural Network (CNN)
- **Input Shape**: 128x128x3 (RGB images)
- **Output**: Binary classification (Mask/No Mask)
- **Framework**: TensorFlow/Keras

🔧 Configuration

Key settings can be modified in `src/config.py`:

- `MODEL_INPUT_SIZE`: Input image dimensions
- `FACE_DETECTION_SCALE_FACTOR`: Face detection sensitivity
- `CONFIDENCE_THRESHOLD`: Minimum confidence for predictions

📚 Educational Resources

Check out the Educational Notebook for:

- Deep learning fundamentals
- CNN architecture explained
- Step-by-step implementation guide
- Model training process

🛠️ Tech Stack

- **Deep Learning**: TensorFlow, Keras
- **Computer Vision**: OpenCV
- **Web Framework**: Streamlit
- **Data Processing**: NumPy
- **Language**: Python 3.8+

🤝 Contributing

Contributions are welcome! Please feel free to submit a Pull Request.

📝 License

This project is licensed under the MIT License - see the LICENSE file for details.

🙏 Acknowledgments

- Face detection using OpenCV Haar Cascades
- Deep learning framework: TensorFlow/Keras
- Web interface: Streamlit

Contact

For questions or feedback, please open an issue on GitHub.

Note: This project was developed for educational and safety monitoring purposes. Ensure compliance with local privacy laws when deploying in production environments.