



# Emotion Recognition App Documentation

<https://www.kaggle.com/datasets/samaneheslamifar/facial-emotion-expressions>

## Introduction

The Emotion Recognition App is designed to demonstrate real-time emotion detection from a webcam feed using computer vision and deep learning. This documentation provides detailed insights into the project structure, functionality, and usage.

## Project Structure

The project is structured as follows:

- **app.py**: Main script containing the application logic.
- **emotion\_model.h5**: Pre-trained weights for the emotion detection model.
- **requirements.txt**: List of required Python libraries.

## Functionality

### 1. Emotion Detection Model

The emotion detection model is implemented as a Convolutional Neural Network (CNN) using TensorFlow and Keras. The model architecture is defined in the `load_emotion_model` function. It consists of convolutional layers for feature extraction, max-pooling layers for downsampling, dropout layers for regularization, and dense layers for classification. The model is loaded with pre-trained weights from `emotion_model.h5`.

## 2. Streamlit Application

The Streamlit application is initialized using the `init_streamlit` function. This function configures the layout, sets up the sidebar with project information, and returns Streamlit components such as placeholders and buttons.

## 3. Real-time Emotion Recognition

The `run` function is the main loop of the application. It captures frames from the webcam, processes each frame, detects faces using Haarcascades classifier, and uses the emotion detection model to predict the emotion of each detected face. The predicted emotion is then displayed in real-time using Streamlit components.

## 4. Exit and Resource Release

The application can be exited by clicking the "Exit" button in the sidebar. The `release_resources` function is triggered, releasing the webcam resources and closing OpenCV windows.

## Usage

To use the Emotion Recognition App:

1. Install Python on your system.
2. Install required libraries using `pip install -r requirements.txt`.
3. Run the application using `python app.py`.
4. Observe the webcam feed and real-time emotion recognition display.
5. Click the "Exit" button to close the application.

## Technologies Used

- **OpenCV:** Used for image and video processing, including face detection.

- **TensorFlow/Keras**: Deep learning framework for building and training the emotion detection model.
- **Streamlit**: Web application framework for creating interactive and user-friendly interfaces.

## Conclusion

The Emotion Recognition App serves as a practical example of integrating computer vision and machine learning for real-world applications. It can be extended for various use cases, such as user experience enhancement, sentiment analysis, or mental health monitoring.

For more detailed information, refer to the source code and comments in [app.py](#).

---

This documentation provides an in-depth understanding of the Emotion Recognition App, its functionality, and how to use it. For any further inquiries or modifications, refer to the source code and comments.