AI-Powered Real-Time Emotion Recognition and Classification System

Project documentation

Documented by: Santosh Bista

Table of Content

Table of Figure

Abstract

The project presents a real-time emotion recognition and classification system integrated with a live video stream using OpenCV and Tkinter. DeepFace, a facial recognition and emotion detection library, powers the system. The application allows users to interact with a camera feed, recognize emotions in real-time, and capture snapshots. The system also provides the option for further classifiers such as Cat vs Dog and Face Recognition, although these modules are placeholders for future development

1 Introduction

1.1 Project Overview

The AI-Powered Real-Time Emotion Recognition and Classification System is designed to detect emotions in a live video stream using facial recognition techniques. This system leverages OpenCV for real-time video streaming, Tkinter for the graphical user interface (GUI), and DeepFace for emotion analysis. The project aims to provide an interactive user experience where the system recognizes facial emotions, provides an option to capture and save snapshots, and supports a modular classifier structure for future classification tasks such as Cat vs Dog and Face Recognition.

1.2 Objective

The main objective of this project is to:

- Capture real-time video feed from the user's webcam.
- Analyze emotions using DeepFace, which detects faces and classifies emotions like happiness, sadness, anger, etc.
- Provide the user with interactive options to classify images or faces, as well as to save snapshots of the current video frame.
- Enable further modules, like Cat vs Dog classification and Face Recognition, as placeholders for future enhancements.

2 Problem Statement

In many fields such as human-computer interaction, security, and psychology, detecting emotions from facial expressions is crucial for personalized experiences. However, real-time emotion detection systems are often complex, costly, or not easily accessible. This project seeks to create a cost-effective, user-friendly solution that integrates emotion detection into everyday computer applications using freely available tools and libraries. The project also lays the groundwork for integrating additional classifiers, such as Cat vs Dog and Face Recognition.

3 System Design and Architecture

3.1 System Flow

1. **User Interface:** The system uses tkinter to create a GUI that displays a live video feed from the webcam and offers options for emotion recognition, face recognition, and future classifiers.

2. **Video Capture**: The system captures video frames from the default webcam using OpenCV (cv2.VideoCapture).

3. **Emotion Detection**: When Emotion Recognition is selected, the system uses DeepFace to analyze the faces detected in each frame and classify the dominant emotion (happy, sad, etc.).

4. **Snapshot**: The system allows the user to capture and save a snapshot of the current frame with a timestamped filename.

5. **Exit and Further Classifiers**: The system offers options for exiting or extending the functionality to additional classifiers (currently placeholders for Cat vs Dog and Face Recognition).

3.2 System Architecture

The system is structured as follows:

1. **Main Application (SnapshotApp):**

- Initializes the Tkinter window.
- Displays the webcam feed on a canvas.
- Provides access to classifier options.

2. **Emotion Recognition Module (EmotionRecognitionApp):**

- Captures webcam frames, detects faces, and uses DeepFace for emotion analysis.
- Allows snapshot saving with the s key and system exit with the q key.

3. **Future Classifier Modules:**

   Placeholders for Cat vs Dog and Face Recognition modules.

4 Implementation

4.1 Libraries Used

The following libraries were used in this project:

- Tkinter: A Python library for GUI creation.
- OpenCV: A powerful library for computer vision tasks, including real-time video capture and face detection.
- Pillow (PIL): Used for image manipulation and integration into Tkinter.
- DeepFace: A Python library for facial recognition and emotion detection, using pre-trained models to analyze emotions from facial images.

4.2 Code Walkthrough

4.2.1 Main Class: SnapshotApp

The SnapshotApp class is the entry point of the application. It initializes the Tkinter GUI, handles the live video feed, and displays the classifier options.

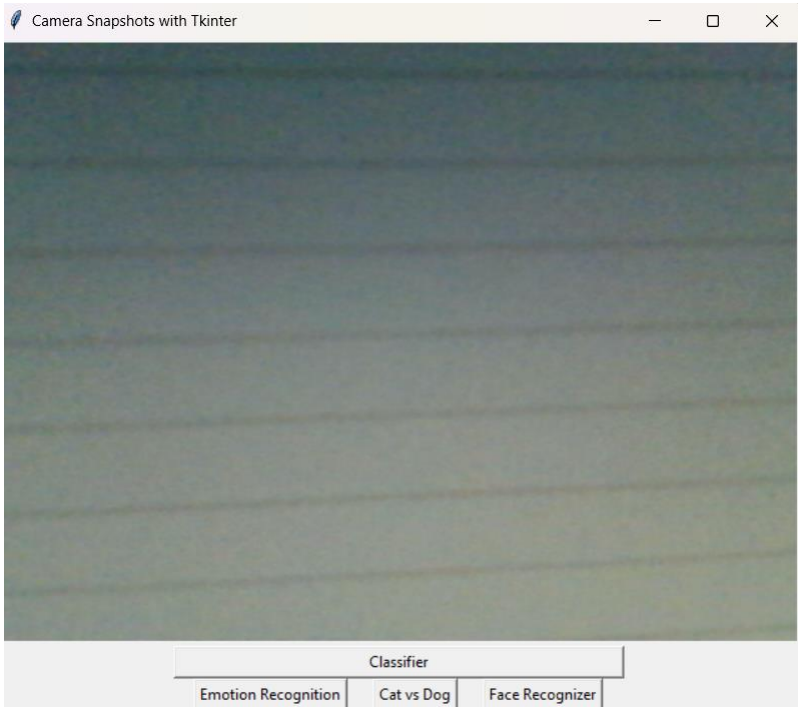

FIGURE 1: MAIN WINDOW

4.2.2 Emotion Recognition Class: EmotionRecognitionApp

The EmotionRecognitionApp class is responsible for capturing video frames, detecting faces, and performing emotion analysis using DeepFace

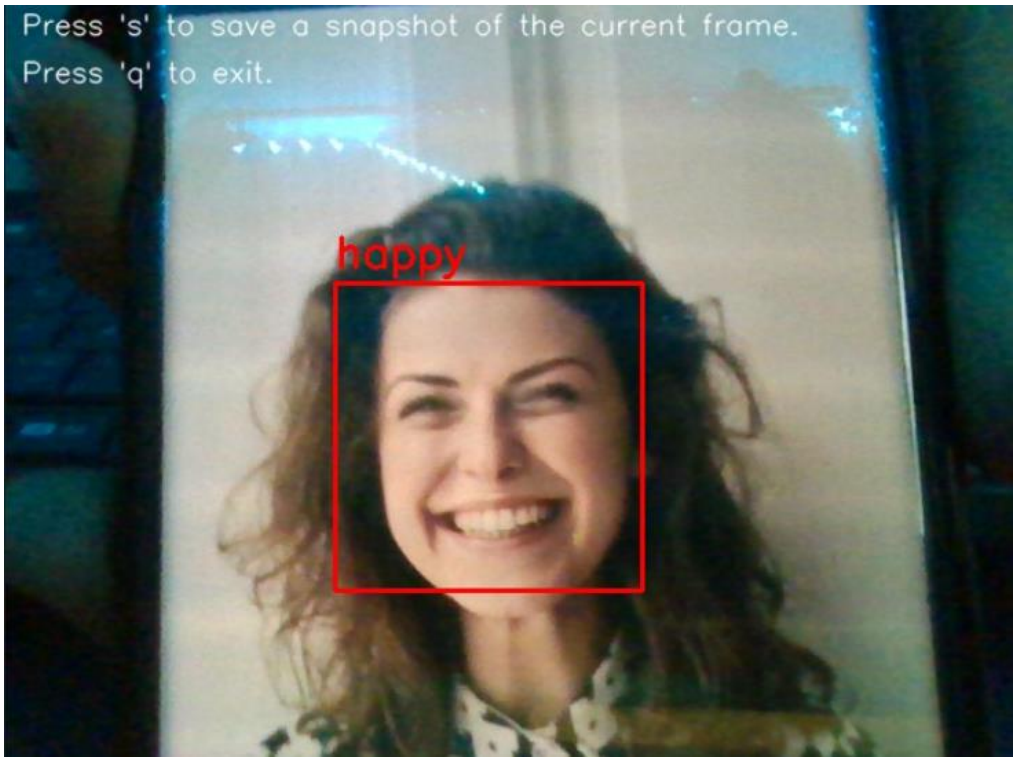

FIGURE 2: CLASSIFIER EMOTION-RECOGNIZER USED IN PIC [5]

4.3 Snapshot Capture

The snapshot capture functionality is implemented in the save_snapshot method.

5 Conclusion

This project successfully implements a real-time emotion recognition system that allows users to interact with a video feed, detect facial emotions, and capture snapshots. The system is highly modular, making it easy to extend with additional classifiers such as Cat vs Dog or Face Recognition. The project can be further enhanced by refining the emotion analysis model, adding more classification modules, and improving the graphical user interface.

6 Future Work

1. Cat vs Dog Module: Implement a classifier that uses a deep learning model to distinguish between images of cats and dogs.

2. Face Recognition: Add a module that identifies faces and associates them with known individuals.

3. GUI Improvements: Enhance the user interface with modern design elements, such as icons, tooltips, and better layout management.

7 References

[1]. https://opencv.org/

[2]. https://github.com/serengil/deepface

[3]. https://wiki.python.org/moin/TkInter

[4]. https://pillow.readthedocs.io/en/stable/

[5] [Online] https://www.pexels.com/photo/woman-in-collared-shirt-774909/ [Accessed on 03.12.2024]