14th Project Innovation Contest 2025

21st ICDCIT

International Conference on Distributed Computing and Internet Technologies (ICDCIT)

THEME : IMAGE PROCESSING PROJECT

TOPIC : FACIAL EXPRESSION RECOGNITION

DEVELOPERS :

KAUSTABH SHIT - 2205131

RANITA TRIPATHY - 2205150

ACKNOWLEDGEMENT

We would like to extend our deepest gratitude to everyone who contributed to the successful completion of this project.

First and foremost, we are thankful to our Software Engineering teacher, **Dr. Prasant Kumar Pattnaik**, for their invaluable guidance, encouragement, and support throughout the course of this project. Their constructive feedback and insightful suggestions have greatly enhanced our work.

We would also like to express our appreciation to our team for their dedication, collaboration, and hard work. The combined efforts and unique contributions of each individual made this project a success. Whether it was research, design, development, or problem-solving, every team member played an essential role.

Finally, we would like to thank our families and friends for their continuous support and understanding during this project.

Their patience and encouragement were vital as we worked through challenges and long hours.

This project has been a great learning experience, and it wouldn't have been possible without the contributions of everyone involved.

INTRODUCTION

Facial Expression Recognition (FER) is a field within computer vision and artificial intelligence (AI) that involves detecting and interpreting human facial expressions from digital images or video streams. The goal of FER systems is to automatically identify and classify emotions or affective states such as happiness, sadness, anger, surprise, fear, disgust, and neutral expressions. These systems can be applied in various real-world applications, ranging from human-computer interaction to healthcare, marketing, and security.

FER project works to design, implement, and evaluate a system that can accurately detect and classify facial expressions in real-time or from static images. The project involves collecting or utilizing an existing dataset of facial expressions, building a machine learning model (such as a CNN), and fine-tuning the model to improve accuracy and performance. This could include tackling specific challenges such as handling facial occlusions, lighting variations, and ensuring real-time processing for practical applications.

In summary, Facial Expression Recognition is an exciting and rapidly advancing field with significant implications across various industries. A successful FER project will contribute to creating more human-aware technologies, improving user experiences, and potentially enhancing various sectors like healthcare, security, and human-computer interaction.

OVERVIEW OF PROJECT

FER is designed to detect emotions either from an image file or live camera feed using the FER (Facial Expression Recognition) library and OpenCV. It allows the user to choose between live emotion detection via a camera or by loading a static image.

Overview of the functionality:

1. **Detect Emotion in Image :**
   (detect_emotion_in_image)

   - **Parameters:**
     - image_path: The path to the image file.
     - Width and height: Optional arguments for resizing the image.
   - **Description:**
     - The function reads the image using cv2.imread().
     - It then uses the FER model (detector = FER()) to detect the dominant emotion in the image with detector.top_emotion().
     - The detected emotion is displayed on the image using cv2.putText().
     - The image with the detected emotion is shown in a window via cv2.imshow(), and the window closes when any key is pressed.

2. **Detect Emotion in Live Camera Feed :**
   (detect_emotion_in_live_camera)

   - **Description:**
     - This function captures real-time video from the webcam using cv2.VideoCapture(0).
     - For each frame, the FER model detects the dominant emotion, which is displayed on the frame using cv2.putText().

  - The webcam feed is displayed in a window, which updates in real-time. Pressing the 'q' key will terminate the camera feed.
  - Once finished, the camera is released with cap.release() and the display window is destroyed with cv2.destroyAllWindows().

3. Main Program : (main)

- **Description:**
  - The user is prompted to choose between detecting emotions from a live camera feed or an image file.
  - Based on the user's choice:
    - If the user selects "1" for the live camera feed, the detect_emotion_in_live_camera() function is called.
    - If the user selects "2", they are prompted to input the path to an image file, and detect_emotion_in_image() is called if the file exists.

LANGUAGES & TOOLS

THINGS TO NOTE :

- FER library should be installed. You can install it using **pip install fer**.
- OpenCV (cv2) should also be installed via **pip install opencv-python.**

SUMMARY :

The code is written in **Python** and utilizes the following libraries:

I. **OpenCV (cv2)** : Used for image and video processing, including reading images, capturing video from the webcam, and displaying the output.

II. **FER (FER)** : A Python library for facial expression recognition that identifies emotions from faces in images or live video streams.

III. **OS** : Used to interact with the file system, such as checking if a file exists.

The code handles two scenarios:

- **Live Camera Emotion Detection :** Detects and displays the dominant emotion in real-time from the webcam feed.
- **Image Emotion Detection :** Reads an image from a file, detects the dominant emotion, and displays it on the image.

CONTRIBUTION

GitHub	Silent18Killer (Kaustabh Shit)	Tripathy-ranita (Ranita Tripathy)
Contribution	Python Code, OpenCV, Camera Setup, Image Processing	PPT, Project Research, FER, Code implementation

CONCLUSION

The FER model may not always return accurate results, and you should ensure that the library and dependencies are installed properly.

The image display and camera feed will not close until the user quits manually or the display window is closed.

FER allows users to detect facial emotions from either a live webcam feed or a static image file. It uses the FER library to identify emotions and overlays the detected emotion on the image/video using OpenCV. The user interface is simple and offers two input methods, with real-time emotion detection provided via the webcam feed. The system handles errors like invalid file paths and unsupported menu choices to provide a smooth user experience.