**Project Title: Emotion Detection from Uploaded Images**

**Objective:**

You will develop a comprehensive system that enables users to upload an image through a Streamlit application and accurately detect and classify the emotion present in the image using Convolutional Neural Networks (CNNs). This project aims to develop and design, implement, and optimize a complete solution that integrates machine learning, computer vision, and user interface design.

**Project Scope:**

**• End-to-End Development:** You will be responsible for all aspects of the project, from setting up the image upload interface to training the CNN model and delivering a polished, user-friendly application.

**• Real-World Relevance:** The project should address potential applications in areas such as healthcare, education, and customer service, where emotion detection is valuable.

**Key Components:**

**1.** User Interface Development:

• Design a Streamlit application that allows users to easily upload images. Do not allow user to upload any other file except image. (Check format, size)

• Focus on creating an intuitive, responsive interface with clear instructions for the user.

**2. Facial Detection Implementation:**

**• Resize the image:** Implement facial detection in the uploaded images using **pre trained models** and **your own model.**

• Explore options to enhance detection accuracy, precision,recall and F1 score, such as refining detection thresholds or combining methods.

**3. Facial Feature Extraction:**

**•** Use tools like Dlib or Mediapipe to extract key facial landmarks.

• Analyze how the accuracy of landmark detection impacts the emotion classification process.

**4. Emotion Classification:**

**•** Train and fine-tune a CNN model using datasets such as [FER-2013](https://www.kaggle.com/datasets/damnithurts/fer2013-dataset-images) available in the Kaggle.

To get the dataset you can use the torchvision library as well. torchvision.datasets.FER2013

• Experiment with different CNN architectures and training techniques to maximize the model’s accuracy.

**5. Performance and Optimization:**

**•** Evaluate the model’s performance with metrics like accuracy, precision, and recall.

**•** Implement optimizations to ensure the system runs efficiently and delivers results in real-time.

**Expected Results:**

**•** A fully functional application where users can upload images and receive accurate emotion classifications.

**•** A detailed report covering the system’s design, implementation, performance analysis, and potential applications.

• An exploration of the ethical considerations related to emotion detection, including privacy concerns and bias mitigation.

**Tools and Technologies:**

**• Programming Language:** Python

**• Frameworks and Libraries:** Streamlit, OpenCV, TensorFlow/Keras or PyTorch

**• Datasets:** [FER-2013](https://www.kaggle.com/datasets/damnithurts/fer2013-dataset-images)

**Deliverables:**

**1. Application:** A Streamlit-based web application for emotion detection.

**2. Codebase:** Well-documented Python scripts for all components of the project.

**3. Trained Models:** Pre-trained and fine-tuned CNN models for emotion classification.

**4. Project Report:** A detailed document covering system design, methodology, experimental results, and conclusions.

**5. Ethical Analysis:** A discussion on the ethical implications of emotion detection technology, including user privacy and bias mitigation strategies.

Project Guidelines:

**• Independent Work:** You are expected to complete all project components independently, demonstrating a strong understanding of the underlying technologies.

**• Regular Milestones:** The project will be divided into milestones with regular progress checks to ensure steady advancement.

**• Final Presentation:** You will present your project to faculty members, showcasing the application and discussing your approach, findings, and any challenges faced during development.