Project Report: Binary Emotion Classification using Deep Learning

Group Members

- **22102A0007**: Parth Satvekar
- **22102A0026**: Jaskirat Singh
- **22102A0023**: Aditya Mamluskar
- **22102A0021**: Soham Jambhwadekar

Introduction

In recent years, advancements in artificial intelligence and machine learning have significantly transformed various sectors, including healthcare, entertainment, and social media. One area of great interest is emotion recognition, where algorithms can identify and classify human emotions based on visual cues. This project focuses on developing a web application that classifies facial expressions into two categories: happy and sad.

Objectives

The primary objective of this project is to create a user-friendly web application that allows users to upload images of faces. The application employs a deep learning model to predict whether the facial expression in the image is happy or sad. The specific objectives include:

- Building a robust image classification model using TensorFlow and Keras.
- Developing a Flask backend to handle file uploads and predictions.
- Ensuring the application provides accurate predictions in real-time.

System Architecture

The architecture of the application consists of a Flask-based backend that interacts with the user through a web interface. Users can upload images, which are processed by a trained model that classifies the facial expressions.

1. **Frontend**: The user interface enables users to upload images and view results. It is designed to be intuitive and responsive.
2. **Backend**: The Flask application manages file uploads, invokes the model for predictions, and returns the results to the frontend.
3. **Model**: A convolutional neural network (CNN) trained on a dataset of labeled images is employed to classify the emotions.

Methodology

The project follows a systematic approach, encompassing the following stages:

1. **Data Collection**: Images of happy and sad faces were sourced from publicly available datasets.
2. **Preprocessing**: The images were resized and normalized to prepare them for training.
3. **Model Training**: A convolutional neural network was trained using TensorFlow. The model was evaluated on a validation set to assess its accuracy.
4. **Integration**: The trained model was integrated into the Flask application, enabling it to process incoming images and return predictions.

Implementation

The implementation of the project involves several key components:

1. **Image Upload**: The web application allows users to upload images in various formats.
2. **Prediction Logic**: Upon receiving an image, the application preprocesses it and feeds it to the trained model, which returns a classification result.
3. **Response Handling**: The results are then displayed to the user, indicating whether the facial expression is classified as happy or sad.

Results

The application was tested with various images to evaluate its performance. The model demonstrated a high level of accuracy in distinguishing between happy and sad expressions, providing real-time feedback to users.

Conclusion

This project successfully demonstrates the potential of deep learning in emotion recognition. The developed web application not only provides a practical tool for classifying emotions but also serves as a foundation for future enhancements, such as incorporating additional emotions or using video input for real-time emotion detection.

Future Work

Future enhancements could include:

- Expanding the model to classify additional emotions such as anger, surprise, or fear.
- Implementing a real-time video analysis feature to capture emotions during live interactions.
- Enhancing the user interface for improved accessibility and usability.