Facial Emotion Recognition

Fatima Rahil

May 2024

1 Introduction

The aims of a facial emotion recognition project using AI are to accurately detect and classify facial expressions, enable real-time processing for applications like user interfaces and sentiment analysis, ensure robustness across diverse conditions, and address privacy and ethical concerns. Additionally, the project aims to foster natural human-like interactions, explore cross-domain applications, and integrate with other technologies for enhanced performance and user experience. This report provides a detailed overview of the project's objectives, methodology, dataset, results, and challenges encountered during execution.

Project Objective

The basic goal of the project is to create and train a machine learning model that can accurately classify emotions. The study aims to increase the model's performance and robustness to various sources of bias by utilizing sophisticated approaches like data augmentation, and preprocessing changes.

Dataset Description

For this project, a dataset comprising 14,248 images was utilized, with an equitable distribution between different emotions images obtained from Kaggle. Kaggle, a prominent platform for data science and machine learning enthusiasts, provided a pivotal resource for this project. With its extensive repository of datasets, competitions, and collaborative forums, Kaggle facilitated access to a diverse array of AI-generated images.

Workflow

The workflow of the project can be outlined as follows:

- **Dataset Collection and Curation:** Images were collected from Kaggle, due to the platform's vast collection of high-quality datasets, ensuring a rich and diverse selection for the project's needs.

- **Data Preprocessing:** Preprocessing steps included normalization using mean and standard deviation values, and grayscale conversion to enhance the model's robustness against variations in color dynamics.
- **Model Training:** A sequential CNN model was built using the Keras framework, stacking convolutional layers for feature extraction and pooling layers for spatial reduction. The model architecture enabled hierarchical feature learning from input photos, which aided in facial emotion identification tasks.
- **Model Improvement:** Data augmentation and pre-processing adjustments were implemented to address identified biases and improve model performance.
- **Evaluation:** Performance metrics including accuracy, confusion matrix, and performance across different models were used to evaluate the model's performance.

Methodologies and Approaches

The methodologies and approaches utilized in this project were meticulously chosen to address the challenges of distinguishing AI-generated images from real photographs. Each approach is detailed below:

- **Sequential CNN Model Analysis:** In our sequential CNN model, layers were systematically arranged to progressively extract complex features from input images, enhancing the model's ability to discern subtle facial expressions. Through iterative convolutional and pooling layers, the model learned hierarchical representations, aiding in accurate emotion classification. This architectural design facilitated efficient feature learning, crucial for robust facial emotion recognition.
- **Data Augmentation:** The dataset was enhanced with techniques such as rotation, shifting, shearing, zooming, and horizontal flipping. It determines the number of additional images required for each class to fulfil a set target count, and then uses Keras to generate augmented images for each. Finally, the original and augmented data are merged, the dataset is shuffled, and the augmented training images are returned with labels. This augmentation approach aids in overcoming data shortage and imbalance issues, hence improving the robustness and generalizability of the facial emotion recognition model.
- **Preprocessing Adjustments:** Several preprocessing modifications were made to improve model resilience. Photos were standardized by shrinking to 224x224 pixels, turning to grayscale, and applying Gaussian blur before loading and preprocessing photos from specified pathways to ensure uniformity and quality. This systematic methodology prepared the dataset

for model training, resulting in improved facial emotion identification accuracy.

- **ResNet50:** ResNet50, a popular convolutional neural network architecture, was used in the model training process. The goal of this inclusion was to use ResNet50's depth and skip connections to improve feature learning and classification accuracy.

Challenges Encountered

- **Limited Resources Due to Kaggle's Online Notebook:** One of the main challenges faced during this project was the limited computational resources available through Kaggle's online platform. Kaggle provides free access to GPUs and CPUs, but these resources are limited in terms of available memory (RAM) and processing time. This constraint often required careful management of memory and computational efficiency, especially when working with large datasets or complex models. As a result, optimizing data preprocessing steps, model architecture, and training procedures to fit within these limitations was necessary to ensure the project could be executed without hitting the platform's usage limits or running into out-of-memory errors.

- **Imbalanced Dataset:** Another significant challenge was dealing with an imbalanced dataset. This imbalance can lead to a model that is biased towards the majority class, resulting in poor generalization performance on the minority classes. To address this, techniques such as data augmentation, resampling methods (e.g., under sampling the majority class or oversampling the minority class), and adjusting class weights in the model training process were considered and implemented to help balance the influence of each class during model training.

- **Inaccurate Base CNN Model:** The initial convolution neural network (CNN) model used as the base for this project showed inadequate performance, likely due to its inability to capture the complexity of the task or sub-optimal architecture choices. This challenge required revisiting the model design, including exploring different architectures, adding regularization techniques like dropout, and tuning hyper-parameters. Ensuring that the model architecture was suitable for the specific characteristics of the dataset was crucial to enhance performance and achieve more accurate results.

- **Challenges with Transfer Learning:** The implementation of transfer learning was significantly hindered by the aforementioned limited computational resources available on Kaggle's platform. Attempts to utilize pre-trained models like ResNet50, MobileNetv2, which are typically large due to their extensive training on datasets like ImageNet, consistently resulted in out-of-memory errors due to the limited GPU and RAM capacities of the free tier. This limitation necessitated reliance on smaller, custom-built models or simplified architectures, which, while fitting within resource constraints, potentially compromised the depth of feature extraction and overall accuracy achievable with more complex pre-trained networks. As a result, the project focused more on optimizing model architecture and data processing to balance performance against the available computational resources.

Results

The methodologies and approaches utilized in this project were meticulously chosen to address the challenges of distinguishing AI-generated images from real photographs. Each approach is detailed below:

- **Model Performance:** Accuracy and resilience were increased by the integration of sophisticated architectures, especially when processing photos with vivid or uncommon colors.
- **Evaluation Matrices:**

	Precision	Recall	F1-score	Support
0	0.42	0.47	0.44	105
1	0.73	0.73	0.73	299
2	0.52	0.65	0.58	322
3	0.57	0.41	0.47	315
4	0.58	0.56	0.57	99
Accuracy		0.58		1140
Macro-Avg	0.56	0.56	0.56	1140
Micro-Avg	0.59	0.58	0.58	1140

Table 1: Evaluation Metrics

References

1. Chollet, François. "The Sequential Model." *Keras.io*, 12 Apr. 2020, https://keras.io/guides/sequential$_{m}odel/.\backslash FacialEmotionRecognitionImageDataset." Kaggle, http$
2. Pramoditha, Rukshan. "Coding a Convolutional Neural Network (CNN) Using Keras Sequential API." *Medium*, 28 June 2022, https://towardsdatascience.com/coding-a-convolutional-neural-network-cnn-using-keras-sequential-api-ec5211126875.
3. Team, Keras. "Keras Documentation: Keras Layers API." *Keras.io*, https://keras.io/api/layers/.
4. TensorFlow. "Convolutional Neural Network (CNN) — TensorFlow Core." *TensorFlow*, 2019, https://www.tensorflow.org/tutorials/images/cnn.