Emotion Detection using Human Computer Interaction

Deewakar Goud, Rituraj Kumar, Shreyas Ladhe
BOTAI
IIIT Vadodara - International Campus Diu

***Abstract*—Emotion recognition plays a pivotal role in advancing Human-Computer Interaction (HCI), enabling systems to better understand and respond to human emotions. With the increasing reliance on AI-driven interfaces, emotion recognition systems have found applications in mental health assessment, virtual reality, intelligent tutoring systems, and customer sentiment analysis. This project focuses on developing an advanced emotion recognition system that utilizes deep learning techniques for analyzing facial expressions and audio signals. Two specialized neural network architectures were employed: a Convolutional Neural Network (CNN) for extracting and classifying features from facial images, and a Time-Distributed CNN coupled with Long Short-Term Memory (LSTM) layers for processing log-mel spectrograms derived from audio signals. The integration of these methods aims to enhance the accuracy and reliability of emotion detection, paving the way for more intuitive and responsive HCI systems.**

I. Introduction

Emotion recognition plays a pivotal role in advancing Human-Computer Interaction (HCI), enabling systems to better understand and respond to human emotions. With the increasing reliance on AI-driven interfaces, emotion recognition systems have found applications in various domains, including mental health assessment, virtual reality, intelligent tutoring systems, and customer sentiment analysis. This project focuses on developing an advanced emotion recognition system that leverages deep learning techniques for analyzing facial expressions and audio signals.

II. Problem Statement

Develop AI models to recognize human emotions from voice, facial expressions, and body language to enhance user experience in Human-Computer Interaction (HCI).

III. Dataset

In this project, two publicly available datasets were utilized to train and evaluate the multimodal emotion recognition system:

- **RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song)** The RAVDESS dataset is a benchmark dataset widely used for audio-based emotion recognition. It contains 24 professional actors (12 male and 12 female) performing emotional speech and songs. The dataset includes:
  1) **Speech data:** 1,440 audio files with 8 emotions (neutral, calm, happy, sad, angry, fearful, surprise, and disgust), each emotion recorded in two levels of intensity (normal and strong), along with a neutral expression.
  2) **Song data:** 1,012 audio files covering the same range of emotions.

  Each file is recorded with high-quality 48 kHz, 16-bit resolution. The RAVDESS dataset provides balanced samples across genders and emotions, making it ideal for training a robust voice emotion recognition model. The distinctiveness of the emotional prosody in speech enables effective feature extraction through the log-mel spectrograms processed by the Time Distributed CNN-LSTM architecture.
- **Facial Expression Recognition Challenge Dataset(FER Challenge)** This dataset is designed for facial emotion recognition and contains labeled images of faces in various emotional states. Key features include:
  1) **Size:** 35,887 grayscale images, each of 48x48 pixels, distributed across training, validation, and test splits.
  2) **Labels:** Seven emotion classes: anger, disgust, fear, happiness, sadness, surprise, and neutral.
  3) **Diversity:** The dataset covers a wide variety of facial expressions across different individuals, lighting conditions, and noise levels, making it a robust choice for training models.

  This dataset offers well-structured facial features, enabling Convolutional Neural Networks (CNNs) to effectively capture spatial hierarchies and patterns in the data. Its diversity ensures generalization to real-world applications, contributing significantly to the visual emotion recognition pipeline of the project.

IV. Model Architecture

A. *Convolutional Neural Network (CNN)*

CNNs are specifically designed to process grid-like data structures such as images, leveraging spatial hierarchies in the data.

Layers in the CNN Architecture

- **Input Layer:** The input to the CNN is an image resized to 48×48 pixels. Each pixel's intensity serves as a feature, and grayscale images reduce the computational complexity compared to RGB.
- **Convolutional Layer:** This layer applies filters (kernels) to extract spatial features like edges, textures, and

shapes from the input.The convolution operation can be expressed mathematically as:

$$y_{ij} = \sigma\left(\sum_{m,n} x_{(i+m)(j+n)} \cdot W_{mn} + b\right),$$

where x is the input, W is the kernel, b is the bias, and σ (ReLU in most cases) introduces non-linearity. This layer detects local features, which are passed to subsequent layers for higher-level abstractions.

- **Activation Function (ReLU):** The Rectified Linear Unit (ReLU) is applied element-wise to the feature map:

$$f(x) = \max(0, x)$$

This layer ensures non-linearity, enabling the network to learn complex patterns.

- **Pooling Layer (Max Pooling):** This layer reduces the spatial dimensions of the feature maps while retaining the most significant information:

$$y_{ij}^{\text{pool}} = \max_{m,n} \left\{ y_{(i+m)(j+n)} \right\}$$

This layer downsamples the feature maps, reducing computational complexity and ensuring translation invariance.

- **Fully Connected Layer:** The output of the convolutional layers is flattened and passed through dense layers, which connect all input neurons to all output neurons. This layer integrates spatial features into a global representation for classification.
- **Output Layer (Softmax):** The final layer produces probabilities for each emotion class using the softmax function:

$$p(y = k \mid x) = \frac{\exp(z_k)}{\sum_{j=1}^{K} \exp(z_j)}$$

This layer assigns a probability score to each emotion class.

B. *Time Distributed CNN-LSTM*

This architecture combines CNNs for feature extraction from spectrogram windows with LSTMs to learn temporal dependencies in audio sequences.

Layers in the CNN-LSTM Architecture

- **Input Layer:** The audio signal is represented as log-mel spectrograms, segmented into fixed-size windows to capture temporal features.
- **Time Distributed Convolutional Layers:** These layers process each spectrogram window independently, similar to a standard CNN, but in a time-distributed manner to maintain the sequence order. This layer extracts local features (e.g., pitch and energy) for each time window.
- **Dropout Layer:**A dropout mechanism randomly sets a fraction of neurons to zero during training:

$$\text{Dropout}(x) = \begin{cases} 0, & \text{with probability } p, \\ x, & \text{with probability } 1 - p. \end{cases}$$

This later prevents overfitting by introducing randomness during training.

- **LSTM Layers (Long Short-Term Memory):** LSTM layers model the temporal dependencies between features extracted from each time window.

$$i_t = \sigma(W_i \cdot x_t + U_i \cdot h_{t-1} + b_i),$$
$$f_t = \sigma(W_f \cdot x_t + U_f \cdot h_{t-1} + b_f),$$
$$o_t = \sigma(W_o \cdot x_t + U_o \cdot h_{t-1} + b_o),$$
$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tanh(W_c \cdot x_t + U_c \cdot h_{t-1} + b_c),$$
$$h_t = o_t \cdot \tanh(c_t),$$

where i_t, f_t, and o_t are the input, forget, and output gates, c_t is the cell state, and h_t is the hidden state. This layer retains important temporal information and discards irrelevant data.

- **Fully Connected Layer:** The LSTM's final hidden state is passed to a dense layer for emotion classification.
- **Output Layer (Softmax):** As in the CNN model, the softmax function provides the final emotion probabilities.

V. Methodology

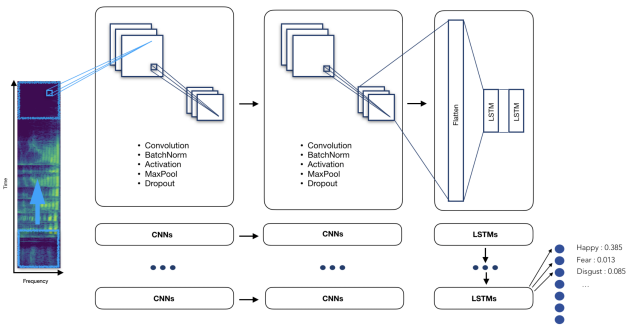
A. *Audio Emotion Recognition*

This section details the methodologies applied for data preprocessing, model architecture, and practical deployment.

*1) **Dataset Preprocessing**:* The preprocessing phase involved multiple steps to ensure the input audio was appropriately structured for model training and inference:

- **Voice Recording**: We captured audio samples using the `PyAudio` library, saving them in WAV format. The sampling rate was set to 16 kHz to balance fidelity and computational efficiency.
- **Feature Extraction**: Using the `Librosa` library, we converted raw audio signals into log-mel spectrograms. These spectrograms captured the frequency domain information essential for emotion detection. The extraction parameters included 128 mel bands and a maximum frequency cutoff of 4000 Hz for optimal resolution.
- **Framing**: Audio signals were framed into fixed-size windows using our custom `frame` function, ensuring compatibility with the time-distributed CNN structure.

*2) **Model Design**:* We developed a Time-Distributed Convolutional Neural Network (CNN) integrated with LSTM layers to leverage both spatial and temporal features of audio data:



- **Convolutional Layers**: Extracted spatial features from the log-mel spectrograms. Each convolutional layer was paired with batch normalization, ELU activations, and max-pooling for efficient feature learning.
- **LSTM Layer**: Captured sequential dependencies in the framed audio data, which are critical for recognizing emotion transitions over time.
- **Fully Connected Layer**: Output the final prediction probabilities for seven emotion classes (*Angry*, *Disgust*, *Fear*, *Happy*, *Neutral*, *Sad*, and *Surprise*) using a softmax activation.

*3) **Model Deployment**:*

- The trained model was loaded using TensorFlow and applied in real-time scenarios via a Flask web application.
- During inference, the `predict_emotion_from_file` function:
  1) Processed audio recordings by framing and normalizing them.
  2) Computed log-mel spectrograms and fed them to the CNN-LSTM model.
  3) Predicted emotion probabilities for each temporal segment of the audio.

*4) **Post-Processing and Visualization**:*

- Predicted emotions were saved and visualized as time-series data. This allowed us to analyze emotion trends over the audio duration.
- The results were integrated into dashboards built using `Altair`, enabling easy interpretation of individual and comparative emotion distributions.

This methodology ensured robust and real-time emotion recognition, leveraging advanced deep learning techniques for practical applications.

B. Facial Emotion Recognition

Below, we outline the practical implementation steps taken during the project, including data preparation, model integration, and system deployment.

*1) **Pre-trained Model and Tools**:* We used a pre-trained deep learning model stored in `model.h5` for emotion classification and the Haar Cascade classifier (`haarcascade_frontalface_default.xml`) for face detection. These tools were integrated into our system using the Keras and OpenCV libraries.

*2) **Dataset and Model Training**:* The pre-trained model was developed using a labeled dataset of facial expressions, resized to dimensions of 48×48 pixels, and normalized to enhance training efficiency. The dataset contained seven emotion categories: *Angry*, *Disgust*, *Fear*, *Happy*, *Neutral*, *Sad*, and *Surprise*. For training, images were converted to grayscale, scaled to the range $[0, 1]$, and expanded to include the necessary dimensions for the model.

*3) **Real-Time Detection Process**:*

- **Face Detection:** Each input frame from a webcam was converted to grayscale to facilitate efficient detection using Haar Cascade. The face regions were detected using the `detectMultiScale()` method, which outputs bounding boxes for identified faces.
- **Preprocessing:** The detected face regions of interest (ROIs) were resized to 48×48 pixels and normalized to match the format of the input dataset.
- **Emotion Classification:** The processed ROIs were passed through the pre-trained model, which outputs probabilities for each of the seven emotion categories. The category with the highest probability was selected as the detected emotion.
- **Visualization:** The system drew bounding boxes around detected faces and annotated them with the corresponding emotion label, displayed in real time on the video feed.

*4) **System Deployment**:* The real-time detection system was implemented in Python and deployed using OpenCV for video capture and frame processing. The application was tested on live video feeds, ensuring accurate detection and classification performance.

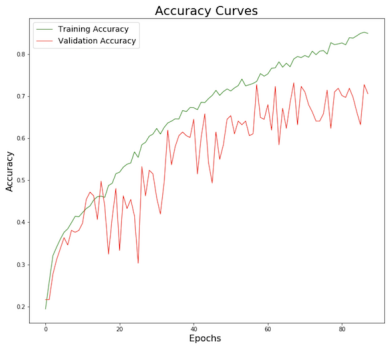
*5) **Error Handling**:* For cases where no faces were detected, the system displayed a `No Faces` message, ensuring robustness and user feedback even in challenging scenarios.

*6) **Output**:* The system outputs a real-time annotated video feed, marking detected faces and overlaying the identified emotion label. This provides immediate insights into the emotional state of individuals in the frame.

VI. Results

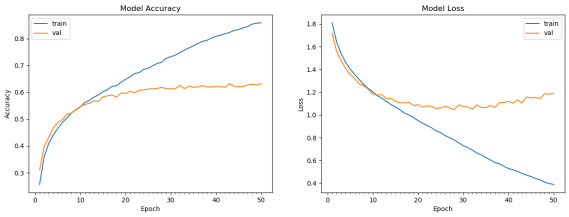
The performance of the facial and audio emotion recognition models was analyzed based on the training and validation accuracy curves obtained during the training phase. These results highlight the effectiveness of the implemented architectures and provide insights into their strengths and limitations.

A. Audio Emotion Recognition



This result illustrates the accuracy and loss curves for the audio emotion recognition model over 50 epochs. The training accuracy progressively improved, surpassing 85%, while the validation accuracy stabilized around 65%.

B. *Facial Emotion Recognition*



The figure displays the accuracy curves for the facial emotion recognition model across 80 epochs. The training accuracy steadily increased, reaching approximately 80% by the end of the training phase. However, the validation accuracy exhibited fluctuations throughout the epochs, stabilizing around 60–65%.

The divergence between training and validation accuracy indicates potential overfitting, where the model performs well on the training data but struggles to generalize to unseen validation data. This could be attributed to the limited size or diversity of the dataset, or to insufficient regularization techniques. Despite these challenges, the model demonstrates a promising baseline for facial emotion recognition tasks.

VII. Challenges

During the development and implementation of the facial and audio emotion recognition models, several challenges were encountered:

- **Data Imbalance:** The dataset used for training had an uneven distribution of emotions, leading to bias in the model's predictions toward majority classes. This hindered the accurate classification of minority emotions.
- **Overfitting:** As observed in the facial emotion recognition model, the training accuracy was significantly higher than the validation accuracy, indicating overfitting. This challenge arose due to the limited dataset size and high model complexity.
- **Feature Extraction:** Both audio and facial emotion recognition required careful preprocessing and feature extraction. Extracting meaningful features from raw data, especially in the audio domain, was a complex and time-intensive process.
- **Model Generalization:** Ensuring that the models generalize well to unseen data was a persistent challenge. While the audio emotion recognition model demonstrated better generalization, the facial model struggled in this aspect.
- **Computational Resources:** Training deep learning models required substantial computational power and memory. Prolonged training times for larger models further complicated experimentation and hyperparameter tuning.

VIII. Future Work

The current work provides a solid foundation for emotion recognition using facial and audio modalities. However, there are several avenues for future exploration to enhance the system's performance and applicability:

- **Multimodal Emotion Recognition:** Combining facial and audio modalities into a unified framework could improve recognition accuracy by leveraging complementary information from both data sources.
- **Advanced Data Augmentation:** Applying sophisticated data augmentation techniques, such as generative adversarial networks (GANs), could help address the data imbalance and improve the model's ability to generalize.
- **Transfer Learning:** Leveraging pre-trained models, such as those trained on large facial and audio datasets, could significantly reduce training time and improve performance on smaller datasets.
- **Real-Time Implementation:** Optimizing the models for deployment in real-time emotion recognition systems would involve reducing computational complexity while maintaining accuracy.
- **Broader Emotion Spectrum:** Expanding the dataset to include a wider range of emotions and cultural diversity would improve the robustness and inclusiveness of the models.

By addressing these challenges and pursuing these directions, the project can be extended to create a more robust, accurate, and versatile emotion recognition system for practical applications.

IX. Conclusion

In this project, we developed and evaluated models for emotion recognition using both facial expressions and audio signals. The results demonstrated the feasibility of leveraging deep learning techniques for emotion recognition, with each modality offering unique strengths. The facial emotion recognition model achieved promising training accuracy but faced challenges in validation performance, highlighting issues of overfitting and data limitations. Conversely, the audio emotion recognition model exhibited better generalization, showcasing its potential for real-world applications.

References

[1] AI-Based Emotion Recognition: Promise, Peril, and Prescriptions for Prosocial Path, Papers with Code, 2022. [Online]. Available: https://paperswithcode.com/task/emotion-recognition. [Accessed: Oct. 3, 2024].

[2] D. Singh, et al., "Deep Learning-Based Facial Emotion Recognition for Human–Computer Interaction," *Neural Computing and Applications*, vol. 33, no. 10, pp. 5127-5145, 2021. [Online]. Available: https://link.springer.com/article/10.1007/s00521-021-06012-8. [Accessed: Dec. 2, 2024].

[3] Y. Zhang, et al., "Emotion Recognition for Human-Robot Interaction: Recent Advances," *Frontiers in Computer Science*, vol. 2, pp. 1-10, 2020. [Online]. Available: https://www.frontiersin.org/journals/computer-science/articles/10.3389/fcomp.2024.1359471/full. [Accessed: Dec. 2, 2024].

[4] Facial Expression Recognition Challenge Dataset, "Facial Expression Recognition Challenge," Kaggle. [Online]. Available: https://www.kaggle.com/datasets/debanga/facial-expression-recognition-challenge. [Accessed: Dec. 2, 2024].

[5] S. Livingstone and F. Russo, "The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS)," *Zenodo*, 2018. [Online]. Available: https://zenodo.org/record/1188976. [Accessed: Dec. 2, 2024].