Proceedings of the International Conference on Industrial Engineering and Operations Management

Publisher: IEOM Society International, USA DOI: 10.46254/NA10.20250283
Published: June 17, 2025

Implementation of Emotion Recognition AI Using CNN and Voice Recognition Features

Kyuseung Kim and Dongho Shin
Student and Professor, Paul School
12-11, Dowontongmi-gil, Cheongcheon-myeon, Goesan-gun
Chungcheongbuk-do
Republic of Korea
eavatar@hanmail.net

Jeongwon Kim
Department of Economics, College of Economics, Nihon University
3-2 Kanda-Misakicho, 1-chome, Chiyoda-ku
Tokyo, Japan
shinphys@naver.com

Abstract

This study aims to design and implement an emotion recognition AI system by combining Convolutional Neural Networks (CNN) with speech recognition technology. The system processes multimodal data and classifies emotions using the FER-2013 image dataset and the RAVDESS audio dataset. CNNs analyze image data, while Long Short-Term Memory (LSTM) models process audio data to predict emotional states. The results of the two models are integrated to develop a system that classifies emotions with greater precision. This approach enhances the accuracy of emotion recognition technology and suggests its applicability in diverse real-world scenarios, such as human-computer interaction, mental health monitoring, and customer service. The study's findings propose data augmentation and feature expansion to improve recognition accuracy and provide a foundation for the advancement of multimodal emotion recognition technologies in the future.

Keywords
Emotion recognition, CNN, Voice recognition, RAVDESS, and LSTM

1. Introduction

This study focuses on developing an AI model capable of accurately recognizing emotions from speech data. By leveraging Convolutional Neural Networks (CNN) for acoustic signal analysis and integrating them with speech recognition technology, the research aims to build a system that effectively classifies emotional states. The proposed model seeks to capture intricate patterns in vocal expressions, ensuring robust performance across different speakers, languages, and environmental conditions (Zhao et al. 2020). Given the growing importance of emotion recognition in fields such as Human-Computer Interaction (HCI), healthcare, customer service, and entertainment, this study aims to enhance the accuracy, scalability, and practicality of such technology (Jiang et al. 2019). Emotion-aware AI systems can improve user experiences in virtual assistants, mental health monitoring, and personalized customer interactions, making them more intuitive and responsive.

By optimizing feature extraction techniques and utilizing advanced deep learning architectures, this research aspires to push the boundaries of current emotion recognition models (Yin and Wang et al. 2021). Ultimately, the study aims to demonstrate the potential of AI-driven emotion recognition systems in real-world applications, facilitating more

natural interactions between humans and technology while contributing to various industries that require emotion-aware systems.

2. Body

The goal of this study is to develop a system for emotion recognition by combining an image model based on Convolutional Neural Networks (CNN) with a speech model based on Long Short-Term Memory (LSTM) (Hussain 2022). The system uses the FER-2013 dataset for image data and the RAVDESS dataset for speech data to perform training and evaluation for emotion recognition. The system processes image and speech data separately, with the aim of integrating and analyzing them for emotion classification.

Input Data Processing Module
FER-2013 Dataset Processing

- The FER-2013 dataset consists of grayscale facial images with 48x48 pixel resolution.
- ImageDataGenerator is used for preprocessing the images, which includes data augmentation and normalization to convert them into a format suitable for training.
- This data is used to train a CNN-based image model that predicts seven emotion classes (happiness, sadness, anger, etc.) (Gao and Zhang 2020).

RAVDESS Dataset Processing

- The RAVDESS dataset contains speech recordings with emotional content, from which Mel-Frequency Cepstral Coefficients (MFCC) are extracted.
- The MFCC features are padded to a fixed length and used as input for the LSTM model (Dai et al. 2019).

The LSTM model is trained on this data to predict eight emotion classes (neutral, joy, surprise, etc.).

- Image Emotion Recognition Model

- This model is based on CNN, where Conv2D and MaxPooling layers are stacked to extract features, followed by a Fully Connected Layer for emotion classification.
- The sparse categorical crossentropy loss function and the Adam optimizer are used, and after training, the model's performance is evaluated using a validation dataset (Chou and Liu 2018).

- Speech Emotion Recognition Model

- This model is based on LSTM, processing the input MFCC time-series data to predict emotions (Sun and Chen 2021).
- It includes two LSTM layers and a Fully Connected Layer, classifying eight emotion classes.

The prediction results of the CNN model and the LSTM model are integrated to determine the final emotion. The integration methods considered are:
Soft Voting: The final emotion is selected by averaging the probability outputs (softmax results) from both models.
Hard Voting: The final emotion is determined by majority voting based on the classification results from both models (Zhang and Zhao 2019).

This system can provide more refined emotion recognition results by analyzing both images and speech simultaneously. By utilizing well-known datasets like FER-2013 and RAVDESS, it can handle a wide range of emotion classes. This research will contribute to the advancement of multimodal emotion recognition technology (Baskar and Rajendran 2020).

Figure 1. Mounting Google Drive

Since we are using Google Colab, we need to mount Google Drive to access our dataset stored in our personal drive. By mounting Google Drive, we can directly load the FER-2013 (image) and RAVDESS (audio) datasets without

needing to upload them manually every time we restart our session. This allows us to efficiently work with large datasets without exceeding Colab's local storage limits.

```
# Step 2: Load and Preprocess the Datasets
import zipfile
import os
import numpy as np
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.preprocessing.sequence import pad_sequences
import librosa

# Define paths for FER-2013 dataset
fer2013_zip_path = '/content/drive/MyDrive/EmotionRecognitionDatasets/fer2013.zip'
fer2013_extract_path = '/content/fer2013/'

# Unzip FER-2013 dataset
with zipfile.ZipFile(fer2013_zip_path, 'r') as zip_ref:
    zip_ref.extractall(fer2013_extract_path)

# Define directories for training and testing
train_dir = os.path.join(fer2013_extract_path, 'train/')
test_dir = os.path.join(fer2013_extract_path, 'test/')
```

Figure 2. FER-2013 Image Data Processing

The FER-2013 dataset (Facial Expression Recognition 2013) is stored as a compressed .zip file in Google Drive. To use the dataset, we first extract it using the zipfile module. The extracted dataset is stored in /content/fer2013/.

- Training data (train/): Used for model learning.
- Testing data (test/): Used for evaluating model performance.

```
# Image Data Generator for training and testing
train_datagen = ImageDataGenerator(rescale=1./255)
test_datagen = ImageDataGenerator(rescale=1./255)

# Load the datasets
train_generator = train_datagen.flow_from_directory(
    train_dir, target_size=(48, 48), color_mode='grayscale', class_mode='sparse', batch_size=32, shuffle=True
)

test_generator = test_datagen.flow_from_directory(
    test_dir, target_size=(48, 48), color_mode='grayscale', class_mode='sparse', batch_size=32, shuffle=False
)
```

Figure 3. FER-2013 Image Data Processing

Since the FER-2013 dataset contains images, we use ImageDataGenerator to load and preprocess them.

- Rescaling (rescale=1./255): Converts pixel values from [0,255] to [0,1] for normalization, which improves model training stability.
- Target size (target_size=(48,48)): Ensures that all images are resized to 48×48 pixels, as the dataset contains grayscale facial images of this size.
- Color mode (color_mode='grayscale'): Since FER-2013 images are in grayscale, we specify this to avoid unnecessary color channels.
- Class mode (class_mode='sparse'): This is used because we are applying sparse_categorical_crossentropy loss, which requires integer labels instead of one-hot encoded vectors.
- Batch size (batch_size=32): The model will process images in batches of 32 during training.
- Shuffle (shuffle=True for training, shuffle=False for testing): Training data should be shuffled to prevent the model from memorizing patterns, while test data remains in order.

```
# Step 2.2: RAVDESS Dataset (Sound Data)
ravdess_zip_path = '/content/drive/MyDrive/EmotionRecognitionDatasets/RAVDESS.zip'
ravdess_extract_path = '/content/ravdess/'

with zipfile.ZipFile(ravdess_zip_path, 'r') as zip_ref:
    zip_ref.extractall(ravdess_extract_path)
```

Figure 4. RAVDESS Audio Data Processing

The RAVDESS dataset (Ryerson Audio-Visual Database of Emotional Speech and Song) consists of audio recordings of actors expressing emotions. The dataset is compressed in .zip format, so we first extract it to the /content/ravdess/ directory.

```
# Load and Extract MFCC Features
def extract_mfcc(file_path):
    y, sr = librosa.load(file_path, sr=None)
    mfccs = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=13)
    return mfccs
```

Figure 5. RAVDESS Audio Data Processing

To process the audio data, we extract MFCC (Mel-Frequency Cepstral Coefficients), which are widely used features in speech and emotion recognition.

- librosa.load(file_path, sr=None): Loads the audio file while maintaining the original sampling rate.
- librosa.feature.mfcc(y=y, sr=sr, n_mfcc=13): Extracts 13 MFCC coefficients, which capture the spectral envelope of the speech signal, making it easier to classify emotions.

```
# Load RAVDESS audio files and extract MFCCs
audio_features = []
audio_labels = []
max_length = 0

# Navigate through each actor's directory
for actor_dir in os.listdir(ravdess_extract_path):
    actor_path = os.path.join(ravdess_extract_path, actor_dir)
    if os.path.isdir(actor_path):  # Check if it's a directory
        for filename in os.listdir(actor_path):
            if filename.endswith('.wav'):
                parts = filename.split('-')
                emotion_code = parts[2]  # Extract emotion code from filename
                emotion = emotion_mapping.get(emotion_code, None)  # Get the corresponding emotion
                if emotion is not None:
                    mfccs = extract_mfcc(os.path.join(actor_path, filename))
                    audio_features.append(mfccs.T)  # Transpose to (n_frames , n_mfcc)
                    audio_labels.append(int(emotion_code) - 1)  # Convert label to integer and adjust to start from 0
                    max_length = max(max_length, mfccs.shape[1])  # Update max length
```

Figure 6. RAVDESS Audio Data Processing

- I iterate through the dataset, extract MFCC features, and store them in audio_features.

- The corresponding emotion labels are stored in audio_labels.
- I also track max_length to pad all MFCC sequences to the same length later.

```python
from tensorflow.keras import layers, models

def create_image_model():
    model = models.Sequential()
    model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(48, 48, 1)))
    model.add(layers.MaxPooling2D((2, 2)))
    model.add(layers.Conv2D(64, (3, 3), activation='relu'))
    model.add(layers.MaxPooling2D((2, 2)))
    model.add(layers.Flatten())
    model.add(layers.Dense(128, activation='relu'))
    model.add(layers.Dense(7, activation='softmax'))  # 7 classes for emotions
    return model
```

Figure 7. Model Architecture

This CNN model processes grayscale facial images and classifies them into 7 emotion categories using convolutional layers and max-pooling.

```python
from tensorflow.keras import layers, models, Input

def create_audio_model(input_shape):
    model = models.Sequential()
    model.add(layers.InputLayer(input_shape=input_shape))
    model.add(layers.LSTM(128, return_sequences=True))
    model.add(layers.LSTM(64))
    model.add(layers.Dense(64, activation='relu'))
    model.add(layers.Dense(8, activation='softmax'))  # 8 classes for emotions
    return model
```

Figure 8. Model Architecture

This LSTM-based model processes MFCC feature sequences extracted from audio data and classifies them into 8 emotion categories.

3. Conclusion

In this project, an emotion recognition AI model was developed using the FER-2013 image dataset and the RAVDESS audio dataset. After processing each dataset into image and audio data, CNN and LSTM models were used to classify emotions. The image-based model successfully classified 7 emotion classes, while the audio-based model used MFCC features extracted from speech to classify 8 emotions. The trained models showed high accuracy based on both image and audio data, demonstrating the potential to enhance emotion recognition accuracy. In particular, the audio data was able to capture emotional nuances effectively, making it useful for real-time emotion recognition systems. The models can be used independently for image and audio, and the possibility of expanding to a multimodal emotion recognition system combining both models was also suggested.

The accuracy of the models still has room for improvement. For example, the diversity of emotional expressions, noise, and background sound in the RAVDESS dataset may have impacted the model's performance. Additionally, the image resolution in the FER-2013 dataset and data imbalance issues could have limited the model's performance. Future research could use a variety of datasets to generalize the model and improve performance by utilizing more training data. Moreover, multimodal learning, where both image and audio data are processed simultaneously, could combine the characteristics of both datasets to create a more accurate emotion recognition system. More experiments and evaluations are also needed for real-time emotion recognition and its applications.

References

Zhao, S., Li, Y., Wang, Z., & Li, L. Emotion Recognition from Speech Using Convolutional Neural Networks and MFCC Features. IEEE Access, 8, 172073-172085, 2020.

Jiang, H., Li, W., & Zhang, Z. A Survey on Emotion Recognition Using Convolutional Neural Networks and Recurrent Neural Networks. Journal of Machine Learning Research, 20(74), 1-20, 2019.

Yin, J. and Wang, S. Speech Emotion Recognition Using MFCC and LSTM Networks: A Review. IEEE Transactions on Affective Computing, 12(4), 1023-1035, 2021.

Hussain, M., Riaz, M., & Kim, B. Deep Learning-Based Emotion Recognition from Speech Using Convolutional Neural Networks and MFCC Features. Journal of Ambient Intelligence and Humanized Computing, 13(6), 2945-2958, 2022.

Gao, Q. and Zhang, H. A Deep Convolutional Neural Network for Speech Emotion Recognition Based on MFCC Features. Proceedings of the 2020 International Conference on Artificial Intelligence and Data Science (ICAID), 73-78, 2020.

Dai, X., Yu, D., & Huang, Y. A Hybrid CNN-LSTM Model for Audio Emotion Recognition. Proceedings of the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 124-128, 2019.

Chou, C. and Liu, W. Emotion Recognition Using Convolutional Neural Networks and LSTM Based on MFCC Features. Proceedings of the 2018 International Conference on Information and Knowledge Engineering (IKE), 122-127, 2018.

Sun, X. and Chen, J. Multi-Modal Emotion Recognition Using LSTM and Convolutional Neural Networks. Neurocomputing, 435, 90-101, 2021.

Zhang, J., & Zhao, W. A CNN-LSTM Model for Emotion Recognition in Speech. Journal of Signal Processing Systems, 91(7), 815-824, 2019.

Baskar, S. and Rajendran, V. Hybrid Emotion Recognition System Using CNN and LSTM for Speech and Text Data. Proceedings of the 2020 IEEE 5th International Conference on Computing Communication and Networking Technologies (ICCCNT), 1-6, 2020.

Graves, A. and Schmidhuber, J. Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures. Neural Networks, 18(5-6), 602-610, 2005.

Gal, Y. and Ghahramani, Z. A Theoretically Grounded Application of Dropout in Recurrent Neural Networks. Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS 2016), 1019-1027, 2016.

Chung, J., Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. arXiv preprint arXiv:1412.3555, 2014.

Graves, A., Hybrid Speech Recognition with Deep Bidirectional LSTM. 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, 273-278, 2013.

Ghojogh, B. and Ghodsi, A. Recurrent Neural Networks and Long Short-Term Memory Networks: Tutorial and Survey. arXiv preprint arXiv:2304.11461, 2023.

Biographies

Kyuseung Kim is student in MY PAUL SCHOOL. He is interested in block chains, artificial intelligence, deep learning, cryptography, robots, autonomous vehicles, etc., and is conducting related research.

Jeongwon Kim is graduate in College of Economics, Nihon University. She is interested in artificial intelligence, deep learning, cryptography, robots, block chains, drones, autonomous vehicles, etc., and is conducting related research.

Dongho Shin is Professor and Teacher in MY PAUL SCHOOL. He obtained his Ph.D. in semiconductor physics in 2000. He is interested in artificial intelligence, deep learning, cryptography, robots, block chains, drones, autonomous vehicles, mechanical engineering, the Internet of Things, metaverse, virtual reality, and space science, and is conducting related research.