

Integrating Facial Expressions, Audio and Textual Data for Trimodal Emotion Recognition

Aryaman Chauhan 21BCE1486, Ayan Siddiqui 21BCE1727, Kshitij Sharma 21BCE5933, *SCOPE*

Abstract—This study addresses the challenge of emotion recognition across diverse modalities, motivated by its significance in affective computing, psychology, and artificial intelligence. The problem entails accurately identifying emotions from speech, facial expressions, and text. The methodology involves leveraging three distinct datasets: RAVDESS for audio, FER2013 for images, and tweet_emotions.csv for text. Each dataset undergoes specific preprocessing steps tailored to its modality, followed by model construction and evaluation. Performance metrics such as accuracy, precision, recall, and F1-score are computed for assessment. Results demonstrate the effectiveness of the proposed machine learning and deep learning approaches in emotion recognition tasks, with implications for human-computer interaction and mental health analysis.

Index Terms— Emotion recognition, affective computing, multi-modality, machine learning, deep learning, preprocessing, performance evaluation.

I. INTRODUCTION

Emotion recognition, a fascinating facet of human communication, reverberates across diverse domains—from affective computing to psychology and artificial intelligence. The ability to discern and interpret emotions accurately not only enhances human-computer interaction but also fuels personalized services and advances mental health analysis. In this pursuit, robust emotion recognition systems have become a focal point, driving research efforts to decode emotional cues across various modalities.

Emotion recognition encompasses three distinct avenues: speech, facial expressions, and textual content. Each modality offers unique insights into the intricate fabric of emotional expression.

The RAVDESS dataset forms the cornerstone of our exploration into audio-based emotion recognition. Comprising recordings of emotional speech by professional actors, this dataset encapsulates a wide spectrum of emotions, intensity levels, and vocal nuances. Through meticulous preprocessing, including feature extraction using Mel-frequency cepstral coefficients (MFCC) and other techniques, we aim to unravel the intricate patterns underlying emotional speech. The subsequent utilization of a Multi-Layer Perceptron (MLP) classifier facilitates the classification of emotional states with

high accuracy and granularity.

In parallel, our research delves into facial expression recognition using the FER2013 dataset, which offers a rich repository of images annotated with seven distinct emotions. By harnessing advanced deep learning architectures and sophisticated preprocessing techniques such as Haar Cascade face detection and spatial normalization, we endeavor to decode the subtle nuances of facial expressions. Through the deployment of convolutional neural networks (CNNs) trained on a diverse array of facial expressions, we aim to achieve robust emotion recognition capabilities from visual cues.

Furthermore, we explore text-based emotion classification using the tweet_emotions.csv dataset, which encapsulates real-world text data reflective of diverse emotional states. Through meticulous preprocessing, including text cleaning, lemmatization, and vectorization, we strive to distill meaningful insights from textual content. Leveraging sequential neural network models equipped with embedding layers and sophisticated regularization techniques, we seek to discern emotional nuances embedded within textual narratives.

By evaluating findings across these modalities, our research seeks to illuminate the challenges and opportunities inherent in emotion recognition. Through rigorous evaluation and analysis of performance metrics, we aim to delineate the efficacy of machine learning and deep learning approaches in deciphering the intricate tapestry of human emotions. Ultimately, our endeavors aspire to pave the way for the development of robust emotion recognition systems with far-reaching implications for human-computer interaction, affective computing, and beyond. [15]

II. LITERATURE SURVEY

[1] presents challenges due to the absence of clear gesture onset and cessation indicators, the necessity for single recognition of gestures, and the constraints of memory and power budgets. This study proposes a hierarchical architecture facilitating the transition of offline convolutional neural network (CNN) models to online operation via a sliding window approach. Performance evaluation is conducted on EgoGesture and NVIDIA Dynamic Hand Gesture Datasets, focusing on temporal detection and classification. The ResNeXt-101 classifier achieves state-of-the-art offline classification accuracies of 94.04% and 83.82% for depth modality.

[2] addresses the automatic detection of individual intake gestures during eating occasions, aiming to enhance dietary monitoring and support dietary recommendations. While previous research predominantly utilizes on-body solutions like inertial and audio sensors, video-based detection of intake gestures remains largely unexplored. These findings underscore the potential of deep learning methods for accurate video-based detection of intake gestures, highlighting the importance of incorporating temporal context and leveraging appearance features in such systems.

The increasing emphasis on emotional intelligence in human-computer interaction underscores the need for computers to perceive human emotions accurately. [3] the authors make two primary contributions: Firstly, they introduce the first authentic facial expression database, wherein test subjects display natural facial expressions corresponding to their emotional states. Secondly, they evaluate various machine learning algorithms for emotion detection, including Bayesian Networks, Support Vector Machines (SVMs), and Decision Trees.

[4] introduces Label-Less Learning for Emotion Cognition (LLEC) as a method to leverage a vast amount of unlabeled data. The approach examines unlabeled data from two perspectives: the feature layer and the decision layer. Through the utilization of a similarity model and an entropy model, a hybrid label-less learning framework is presented, capable of automatically labeling data without human intervention. Experimental findings demonstrate a significant improvement in the accuracy of emotion detection using the LLEC algorithm. Key terms: Deep learning, emotion detection, label-less learning, multimodal emotion cognition.

[5] Detecting emotions in text presents a challenging task on classifying polarity. This paper introduces a Multi-label Emotion Detection Architecture (MEDA) designed to identify all associated emotions expressed within a given text. MEDA consists of two core modules: the Multi-Channel Emotion-Specified Feature Extractor (MC-ESFE) and the Emotion Correlation Learner (ECorL). The effectiveness of the proposed MEDA architecture is assessed on the RenCECps and NLPCC2018 datasets. Experimental findings demonstrate that MEDA surpasses state-of-the-art methods in this task, underscoring its capability in capturing the intricate nuances of emotional expressions within textual data.

[6] Understanding an architectural framework designed as a proof-of-concept for emotion detection and regulation within smart health environments is the primary objective of this study. The proposed framework aims to discern the emotional state of patients by analyzing their physiological signals, facial expressions, and behavior. The paper delineates the three primary components of the architecture: "Emotion Detection," "Emotion Regulation," and "Emotion Feedback Control." Ongoing efforts involve testing the overall architecture and interventions within real-world environments to realize the ultimate goal.

[7] A burgeoning topic in human-computer interaction, with extensive research focusing on detecting emotions from facial and audio cues is emotion detection. This paper presents a comprehensive knowledge-based survey on emotion detection from textual data, along with the methods employed for this purpose. The proposed emotion detector system accepts a text document and the emotion ontology as inputs, subsequently generating one of six emotion classes—love, joy, anger, sadness, fear, or surprise—as the output.

[8] This study endeavors to overcome the inherent challenges of identifying human emotions solely through facial expressions and behavior by exploring the detection of emotions through the analysis of brain waves. The research methodology comprises four key components: (i) simultaneous collection of both visual and EEG data from participants, (ii) complete data recording through image capture using cameras and EEG devices, (iii) concurrent preprocessing, classification, and feature extraction of the collected data, and (iv) classification of the extracted features using artificial intelligence techniques to discern emotional states from facial expressions.

[9] The emotional states of individuals, also referred to as moods, play a pivotal role in shaping their expressions of thoughts, ideas, and opinions, consequently influencing attitudes and behaviors. This paper proposes a method for detecting and classifying the emotions or moods conveyed in tweets, assigning them to appropriate emotional categories. The approach employs a two-step process, combining a Rule-Based Approach (RBA) and a Machine Learning Approach (MLA) for classification. Notably, this paper contributes to the detection of emotion in non-hashtagged data and the creation of labeled data for the machine learning approach without manual intervention.

[10] This paper introduces a novel approach for emotion detection using a cascade of Mutation Bacteria Foraging Optimization (MBFO) and Adaptive Median Filter (AMF) in highly corrupted noisy environments. The method involves noise removal from images through the combination of MBFO and AMF, followed by the extraction of local, global, and statistical features from the images. The proposed method achieves a high recognition rate for all emotions, including neutral, both on the training dataset and a user-defined dataset. The approach utilizes the cascading of MBFO and AMF for noise removal, coupled with Neural Networks for emotion classification.

[11] The way our emotions work play a crucial role in education, affecting affective development, attitudes, and learning outcomes, which are typically assessed through affective assessment methods. The detected emotions are processed and displayed on a website-based information system for affective assessment. Validation tests show emotion detection accuracy values of 84.4% and 80.95% after testing in school settings. Furthermore, usability acceptance tests conducted with teachers and students yielded scores of 77.56% and 79.85%, respectively. Through various tests, the

developed system demonstrates its capability to facilitate the implementation of affective assessment processes effectively.

[12] Machine learning algorithms have revolutionized computer vision, enabling the development of novel approaches for emotion detection, which in turn facilitates the prediction of human behavior and subsequent actions. With the computational power provided by GPUs, these algorithms excel in real-world scenarios, extending beyond niche fields to encompass various domains, including behavioral sciences. Concurrently, technologies like augmented reality and virtual reality are gaining prominence, with robotic vision and interactive robotic communication emerging as intriguing applications.

[13] A pivotal role in human decision-making and communication, and with the recent surge in human-computer interaction, affective computing has emerged as a prominent research area is often played by our emotions. It aims to develop computational systems capable of understanding and responding to human emotions. The review analyzed methodological aspects including emotion models, devices, architectures, and classification techniques employed by the selected studies, identifying popular techniques and current trends in visual emotion recognition. It highlights current trends, such as the increasing use of deep learning algorithms and the importance of studying body gestures.

[14] What stands as a burgeoning and significant research field within the domain of pattern recognition. In everyday interactions, non-verbal communication plays a pivotal role, contributing approximately 55% to 93% to overall communication is Facial emotion recognition (FER). This paper presents a comprehensive review of FER literature collected from reputable research publications during the current decade. The review encompasses both conventional machine learning (ML) and various deep learning (DL) approaches employed in FER. Additionally, the paper discusses and compares different FER datasets utilized for evaluation metrics and compares them with benchmark results.

III. PROPOSED WORK

A) Audio Sources

1. Data Source:

The RAVDESS dataset comprises 1440 audio files featuring 24 professional actors (12 female, 12 male) expressing various emotions such as calm, happy, sad, angry, fearful, surprise, and disgust, at different intensity levels. Each file is labeled with a unique identifier indicating modality, vocal channel, emotion, emotional intensity, statement, repetition, and actor.

The dataset aids research in emotion recognition and speech analysis, offering a diverse range of emotional expressions in a neutral North American accent. It is

valuable for studies in affective computing, psychology, and artificial intelligence.

2. Data Preprocessing:

The code commences by loading audio files from the RAVDESS dataset, which comprises recordings of emotional speech. It then attempts to convert these audio files into text using the *SpeechRecognition* library, although this functionality is commented out due to potential limitations in accurately transcribing a large volume of files. Following this, the code performs various preprocessing steps on the audio data, including down-sampling to enhance computational efficiency, applying a masking technique to remove unnecessary empty segments, and saving the cleaned audio files to a designated directory.

Additionally, the code defines functions for extracting essential features from the audio files, such as Mel-Frequency Cepstral Coefficients (MFCC), chroma, and mel frequency cepstral coefficients (MFCC), which are commonly utilized in speech analysis tasks. These preprocessing operations are crucial for preparing the audio data for subsequent analysis and model training, ensuring data integrity, computational tractability, and extraction of informative features essential for emotion recognition tasks.

3. Machine Learning Models

The Multi-Layer Perceptron (MLP) is a type of artificial neural network (ANN) that belongs to the class of feedforward neural networks. It consists of multiple layers of nodes, each connected to nodes in the adjacent layers. In the project, the MLP classifier is employed as a machine learning model for the classification of emotional speech audio data from the RAVDESS dataset.

i. Structure:

The MLP classifier is composed of an input layer, one or more hidden layers, and an output layer. Each layer contains multiple nodes (or neurons), and connections between nodes are associated with weights. The layers are interconnected through weighted connections, and each node applies an activation function to the weighted sum of its inputs to produce an output.

ii. Training:

a. Forward Propagation: During training, input data (features extracted from audio files) are fed forward through the network. Each layer performs a linear transformation (weighted sum of inputs) followed by an activation function (such as sigmoid or ReLU).

b. Backpropagation: The output of the network is compared to the true labels (emotions) using a loss

function (e.g., cross-entropy loss). The error is propagated backward through the network using the gradient descent algorithm, adjusting the weights to minimize the loss.

c. Optimization: Various optimization algorithms (e.g., stochastic gradient descent, Adam) are utilized to update the weights iteratively, optimizing the network's performance.

iii. Features in the Project:

In the project, features extracted from audio files serve as input to the MLP classifier. These features include:

- MFCC (Mel-frequency cepstral coefficients): Representations of the short-term power spectrum of sound.
- Chroma: Represents the energy distribution of pitch classes in the audio.
- Mel Spectrogram: Represents the energy distribution of frequencies in the audio.

iv. Performance Metrics:

After training, the MLP classifier is evaluated using a testing dataset. Performance metrics such as accuracy, precision, recall, and F1-score are computed to assess the classifier's effectiveness in classifying emotional speech audio.

4. Pseudocode

```
# Define functions for feature extraction and data loading
def extract_feature(file_name, mfcc, chroma, mel):
    # Extract features from the audio file
    # Parameters: file_name - path to the audio file
    # mfcc, chroma, mel - boolean flags indicating which features to extract
    # Returns: feature vector containing extracted features

def load_data(test_size):
    # Load audio files, extract features, and split data into training and testing sets
    # Parameters: test_size - proportion of data to be used for testing
```

```
# Returns: x_train, x_test, y_train, y_test - feature vectors and labels for training and testing
```

```
# Define emotions in the RAVDESS dataset to be classified audio files based on
```

```
emotions = {
    '01': 'neutral',
    '02': 'calm',
    '03': 'happy',
    '04': 'sad',
    '05': 'angry',
    '06': 'fearful',
    '07': 'disgust',
    '08': 'surprised'
}
```

```
# Define observed emotions
```

```
observed_emotions = ['calm', 'happy', 'fearful', 'disgust']
```

```
# Define the Multi-Layer Perceptron (MLP) classifier
```

```
model = MLPClassifier(alpha=0.01,
    batch_size=256, epsilon=1e-08,
    hidden_layer_sizes=(300,),
    learning_rate='adaptive', max_iter=500)
```

```
# Load the data and split into training and testing sets
```

```
x_train, x_test, y_train, y_test =
    load_data(test_size=0.25)
```

```
# Train the model
```

```
model.fit(x_train, y_train)
```

```
# Predict using the trained model
```

```
y_pred = model.predict(x_test)
```

```
# Calculate accuracy
```

```
accuracy = accuracy_score(y_test, y_pred)
```

```
# Store the prediction probabilities into a CSV file
```

```
predictions_df = pd.DataFrame(y_pred,
    columns=['predictions'])
predictions_df['file_names'] = test_filename
predictions_df.to_csv('predictionfinal.csv')
```

```
# Define input audio file path
```

```
INPUT_AUDIO_FILE = "test.wav"
```

```
# Write audio data to WAV file
```

5. Equations

1. Mean Squared Error (MSE)

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

2. Root Mean Squared Error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

3. Signal Envelope

$$E(t) = \frac{1}{N} \sum_{i=t}^{t+N} |x_i|$$

4. Fourier Transform

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt$$

5. Mel-Frequency Cepstral Coefficients (MFCCs)

$$\text{MFCC}(t) = \text{DCT}(\log(\text{Mel}(\text{STFT}(x(t))))$$

6. Chroma Feature Extraction

$$C_q(t) = \sum_{n=0}^{N-1} |X(n, t)|^2$$

7. Mel-Spectrogram

$$S_M(f, t) = \sum_{n=0}^{N-1} |X(n, t)|^2 H(f, t)$$

8. Model Evaluation - F1 Score

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

9. Mean Absolute Error (MAE)

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

10. Spectrogram

$$S(f, t) = |X(f, t)|^2$$

11. Convolutional Neural Network (CNN) Layer

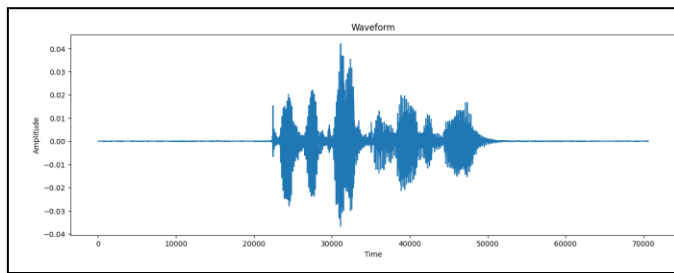
$$Z^{[l]} = W^{[l]} * A^{[l-1]} + b^{[l]}$$

12. ReLU Activation Function

$$\text{ReLU}(x) = \max(0, x)$$

6. Results

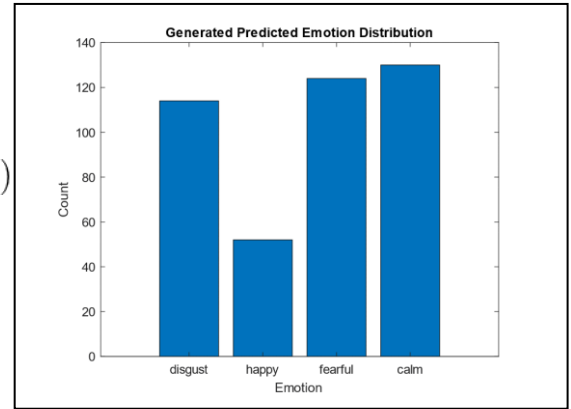
Analyzing waveform data:



Final model predictions

	predictions	file_names
0	disgust	03-01-07-01-02-01-19.wav
1	disgust	03-01-06-01-02-02-01.wav
2	disgust	03-01-03-01-01-01-22.wav
3	happy	03-01-03-01-01-01-02.wav
4	fearful	03-01-06-01-02-01-22.wav
..
187	fearful	03-01-06-02-01-01-05.wav
188	disgust	03-01-07-01-02-01-24.wav
189	calm	03-01-02-01-01-01-14.wav
190	happy	03-01-07-02-01-02-10.wav
191	calm	03-01-02-02-01-01-03.wav

[192 rows x 2 columns]



B) Text Source

1. Data Source

The data source is a CSV file (tweet_emotions.csv) that contains information about tweets and their corresponding emotions. This file serves as the dataset for training and evaluating the model. The dataset is organized with three columns:

tweet_id: This column contains a unique identifier for each tweet. It is a numerical value that can be used to reference specific tweets.

sentiment: This column contains the emotion or sentiment associated with each tweet. The emotions can include categories such as "empty," "sadness," "enthusiasm," "neutral," "worry," and others. These labels provide the target classes for the model to learn and classify.

content: This column contains the text content of the tweet. The text can be a short message, often in a conversational style, and may include a variety of elements such as links, punctuation, emoticons, and other text features.

The dataset includes a variety of emotions, such as sadness, enthusiasm, worry, and neutral. This diversity allows the model to learn from a range of emotional expressions.

2. Data Preprocessing

- a. **Cleaning Text Data:** The project includes cleaning text data by converting all text to lowercase and removing unnecessary characters and URLs. Specific regular expressions are used to filter out bad symbols and unwanted characters. Additionally, numbers and other irrelevant text are removed.
- b. **Lemmatization and Stopword Removal:** The project uses lemmatization to convert words into their base forms, which helps in generalizing the text data and reduces complexity. Stopwords (common words like "the," "is," etc.) are removed to focus on significant words that contribute more to determining emotions.
- c. **Vectorization:** A `TextVectorization` layer is employed to convert text data into integer sequences. This involves tokenizing the text data and mapping it to integer representations. The layer also limits the maximum number of unique tokens (words) to be considered, and it standardizes the text input for efficient processing.
- d. **Splitting Data:** The data is split into training and testing sets. Training data is used to train the model, while testing data is used to evaluate the model's performance.

3. Machine Learning and Deep Learning Model

- a. **1. Embedding Layer:** An embedding layer is used to transform integer sequences into dense vectors of fixed size (embedding dimension). This allows the model to capture semantic meaning and relationships between words.
- b. **Spatial Dropout:** Spatial dropout is employed to regularize the model and prevent overfitting. This technique randomly drops units in the input embedding layer, enhancing generalization.
- c. **Global Max Pooling:** A global max pooling layer follows the embedding layer, aggregating the most important features from the embeddings and reducing dimensionality.

- d. **Dense Layers:** The model includes dense (fully connected) layers with dropout layers in between. These layers process the extracted features and learn to classify the input data into one of the emotion categories.
- e. **Output Layer:** The output layer is a dense layer with softmax activation, which provides a probability distribution across the different emotion categories.
- f. **Model Compilation:** The model is compiled using an Adam optimizer, a sparse categorical cross-entropy loss function, and accuracy as the primary metric for evaluation.
- g. **Training and Validation:** The model is trained using the training data with specified batch size and epochs. Early stopping is implemented to prevent overfitting.
- h. **Evaluation and Predictions:** The model is evaluated using the testing data, and its performance is assessed based on accuracy and loss. Predictions can be made on new input text to classify it into one of the emotion categories.

4. Pseudocode

```
// Import necessary libraries
import libraries for data manipulation,
visualization, neural networks, and natural
language processing

// Set random seed for reproducibility
set SEED = 42
set_random_seed(SEED)

// Load and filter the dataset
load data from 'file_path'
filter data for 'happiness' and 'sadness' emotions
```

```
// Split data into training and testing sets
train_df, test_df = train_test_split(data,
test_size=0.2, random_state=SEED)

// Display length of train and test sets
print(len(train_df), len(test_df))

// Calculate class weights for handling
imbalanced classes
class_weights =
compute_class_weight('balanced',
classes=unique_labels, y=labels)

// Define text cleaning function
function clean_text(text):
    lowercase text
    remove URLs, symbols, punctuation
    return cleaned text

// Define vectorizer layer for text data
vectorizer_layer =
TextVectorization(max_tokens=5000,
standardize=clean_text,
output_sequence_length=256)
vectorizer_layer.adapt(train_df['sentence'])

// Define neural network model architecture
model = Sequential([
    Input(shape=(1,), dtype='string'),
    vectorizer_layer,
    Embedding(max_features, embedding_dim),
    SpatialDropout1D(0.2),
    GlobalMaxPooling1D(),
    Dense(256, activation='gelu'),
    Dropout(0.4),
    Dense(num_classes, activation='softmax')
])

// Compile the model with optimizer, loss, and
metrics

// Prepare training and testing data
X_train = train_df['sentence']
y_train = train_df['label'].codes
X_test = test_df['sentence']
y_test = test_df['label'].codes

// Train the model with early stopping callback
history = model.fit(
    X_train, y_train,
    validation_data=(X_test, y_test),
    batch_size=256,
    epochs=300,
    class_weight=class_weights,
    callbacks=[EarlyStopping(patience=3)]
)
```

```
// Evaluate the model on test data
test_loss, test_acc = model.evaluate(X_test,
y_test)

// Plot training and validation accuracy and loss
function plot_history(history):
    plot accuracy and loss for training and
validation sets
    show the plot

// Call the plot history function
plot_history(history.history)

// Make predictions on test data
y_pred =
model.predict(X_test).argmax(axis=1)

// Print classification report
print(classification_report(y_test, y_pred,
target_names=label_names))

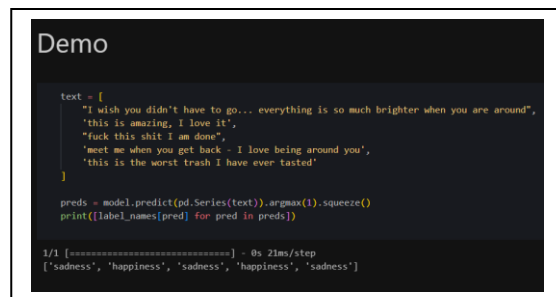
// Display confusion matrix
display confusion matrix

// Define new test sentences
test_sentences = [
    "I wish you didn't have to go... everything is
so much brighter when you are around",
    "this is amazing, I love it",
    "I am done with this",
    "meet me when you get back - I love being
around you",
    "this is the worst trash I have ever tasted"
]

// Make predictions on new test sentences
preds =
model.predict(test_sentences).argmax(axis=1)

// Print the predicted emotions
print([label_names[pred] for pred in preds])
```

5. Result



```
Demo

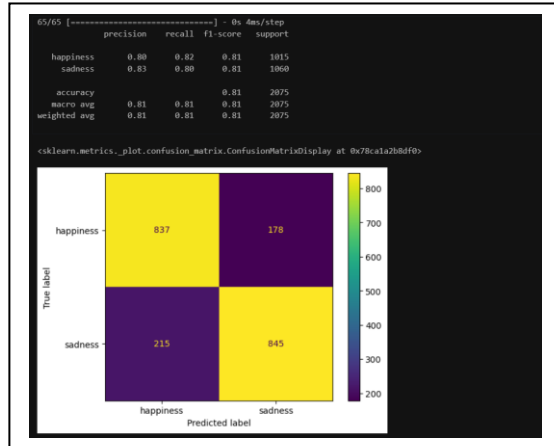
text = [
    "I wish you didn't have to go... everything is so much brighter when you are around",
    "this is amazing, I love it",
    "fuck this shit I am done",
    "meet me when you get back - I love being around you",
    "this is the worst trash I have ever tasted"
]

preds = model.predict(pd.Series(text)).argmax(1).squeeze()
print([label_names[pred] for pred in preds])

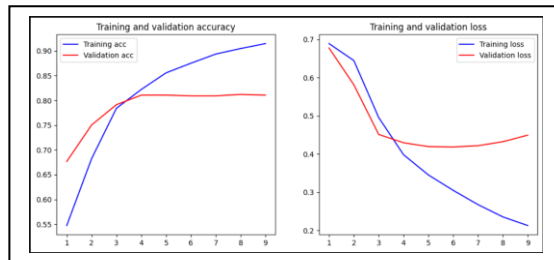
1/1 [=====] - 0s 21ms/step
['sadness', 'happiness', 'sadness', 'happiness', 'sadness']
```

Demo for text source

6. Evaluation



Confusion Matrix

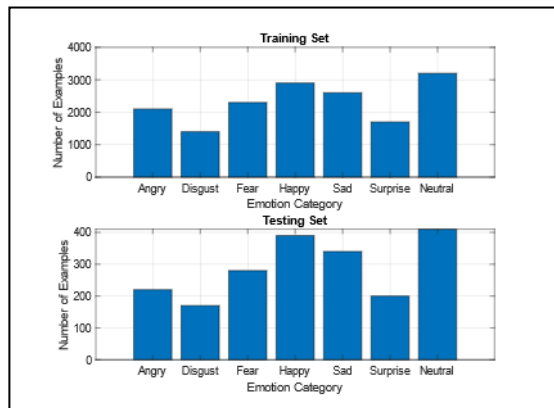


Training and validation accuracy and loss

C) Video Source

1. Data Source

FER2013 dataset, comprising images of human faces annotated with seven different emotions: anger, disgust, fear, happiness, sadness, surprise, and neutrality.



2. Data preprocessing

The process initiates with real-time capture of video frames, which undergoes face detection using the Haar Cascade classifier. Detected faces are extracted as regions of interest (ROIs) from the original images, with subsequent resizing to match the model's input size. Grayscale conversion

simplifies processing, followed by normalization and scaling of pixel values to ensure uniformity across images. Optional data augmentation techniques may be employed to enhance dataset diversity. Ultimately, preprocessed facial images are fed into the emotion recognition model, which outputs probabilities for each emotion class. This meticulous preprocessing ensures that input data conforms to model requirements, optimizing accuracy in real-time emotion detection from video streams.

3. Training

The provided OpenCV code utilizes the Haar Cascade classifier to detect faces in real-time from a webcam feed or video stream. Detected faces are then passed through the trained emotion recognition model (model.h5) to predict the associated emotion. The emotions' probabilities are plotted in real-time to visualize the model's predictions.

```
Epoch 12: ReduceLRonPlateau reducing learning rate to 0.000200000000949949026.  
225/225 [-----] - 16s 70ms/step - loss: 0.8039 - accuracy: 0.6965 - val_loss: 1.1843  
Epoch 12: early stopping
```

Accuracy of model.h5 is 69.65%.

Notebook available at:

<https://colab.research.google.com/drive/1OsWXklkOWmB2vNE0aLhIMWSXBG6WIyJQ?usp=sharing>

4. Machine Learning Models

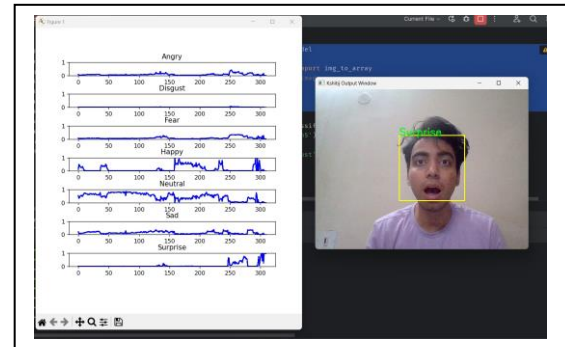
The model architecture includes convolutional layers followed by batch normalization, activation layers, max-pooling, and dropout for regularization.

- CNN: The CNN architecture consists of multiple layers, including convolutional layers, pooling layers, and fully connected layers.
- Convolutional layers extract features from input images by applying convolutional filters, which detect patterns and spatial relationships within the images.

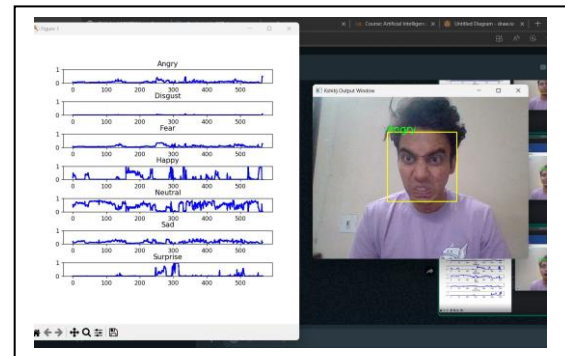
5. Pseudocode

- Load pre-trained face detection classifier (e.g., haarcascade_frontalface_default.xml) and emotion classification model (e.g., model.h5)
- Define emotion labels: ['Angry', 'Disgust', 'Fear', 'Happy', 'Neutral', 'Sad', 'Surprise']
- Initialize parameters:
 - scale = 25
 - xs = []
 - ys = [] for _ in range(7)
 - index = 0

4. Create a figure and subplots for each emotion label using matplotlib
5. Open a video capture device (e.g., webcam)
6. Start an infinite loop to continuously process frames:
 - Read frame from the video capture device
 - Convert the frame to grayscale
 - Detect faces in the grayscale frame using the face detection classifier
 - If faces are detected:
 - Preprocess the face region
 - Make predictions using the emotion classification model
 - Update the plot data for each emotion label
 - Display the emotion label on the frame
 - Display the frame
 - Wait for a key press (e.g., 'q' to quit)
 - Increment the index
7. Close the video capture device and destroy all windows

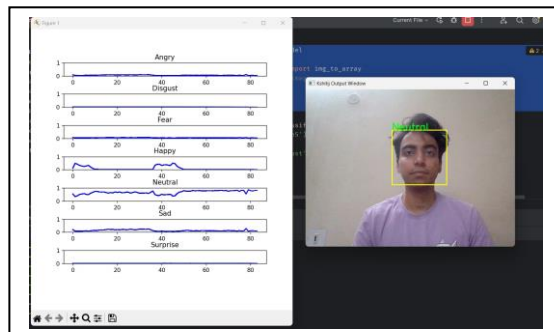


Surprise

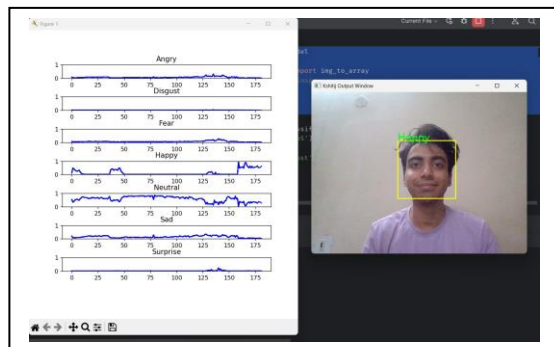


Anger

6. Result



Neutral



Happy

IV. CONCLUSION

This paper has addressed the challenge of emotion recognition across diverse modalities, spanning speech, facial expressions, and textual content. Leveraging datasets such as RAVDESS, FER2013, and tweet_emotions.csv, specific preprocessing techniques tailored to each modality were employed, followed by model construction and evaluation using machine learning and deep learning approaches. Through rigorous analysis and evaluation of performance metrics, including accuracy, precision, recall, and F1-score, the effectiveness of the proposed methodologies in emotion recognition tasks has been demonstrated. The findings have significant implications for human-computer interaction, affective computing, and mental health analysis, paving the way for the development of robust emotion recognition systems in various real-world applications.

REFERENCES

- [1] Köpüklü, Okan, et al. "Real-time hand gesture detection and classification using convolutional neural networks." 2019 14th IEEE international conference on automatic face & gesture recognition (FG 2019). IEEE, 2019
- [2] Rouast, Philipp V., and Marc TP Adam. "Learning deep representations for video-based intake gesture detection." IEEE journal of biomedical and health informatics 24.6 (2019): 1727-1737.
- [3] Sun, Yafei, et al. "Authentic emotion detection in real-time video." Computer Vision in Human-Computer Interaction: ECCV 2004 Workshop on HCI, Prague, Czech Republic, May 16, 2004. Proceedings. Springer Berlin Heidelberg, 2004.

- [4] Chen, Min, and Yixue Hao. "Label-less learning for emotion cognition." *IEEE transactions on neural networks and learning systems* 31.7 (2019): 2430-2440.
- [5] Deng, Jiawen, and Fuji Ren. "Multi-label emotion detection via emotion-specified feature extraction and emotion correlation learning." *IEEE Transactions on Affective Computing* 14.1 (2020): 475-486.
- [6] Fernández-Caballero, Antonio, et al. "Smart environment architecture for emotion detection and regulation." *Journal of biomedical informatics* 64 (2016): 55-73.
- [7] Shivhare, Shiv Naresh, and Sri Khetwat Saritha. "Emotion detection from text documents." *International Journal of Data Mining & Knowledge Management Process* 4.6 (2014): 51.
- [8] Ismail, WOAS Wan, et al. "Human emotion detection via brain waves study by using electroencephalogram (EEG)." *International Journal on Advanced Science, Engineering and Information Technology* 6.6 (2016): 1005-1011.
- [9] Suhasini, Matla, and Srinivasu Badugu. "Two step approach for emotion detection on twitter data." *International Journal of Computer Applications* 179.53 (2018): 12-19.
- [10] Nagpal, Renu, Pooja Nagpal, and Sumeet Kaur. "Hybrid technique for human face emotion detection." *International Journal of Advanced Computer Science and Applications* 1.6 (2010).
- [11] Artanto, Herjuna, and Fatchul Arifin. "Emotions and gesture recognition using affective computing assessment with deep learning." *Int J Artif Intell* 12.3 (2023): 1419-1427.
- [12] Raman, Swati, et al. "Emotion and Gesture detection." *International Journal for Research in Applied Science and Engineering Technology* 10 (2022): 3731-3734.
- [13] Leong, Sze Chit, et al. "Facial expression and body gesture emotion recognition: A systematic review on the use of visual data in affective computing." *Computer Science Review* 48 (2023): 100545.
- [14] Khan, Amjad Rehman. "Facial emotion recognition using conventional machine learning and deep learning methods: Current achievements, analysis and remaining challenges." *Information* 13.6 (2022): 268.
- [15] Keshari, Tanya, and Suja Palaniswamy. "Emotion recognition using feature-level fusion of facial expressions and body gestures." *2019 international conference on communication and electronics systems (ICCES)*. IEEE, 2019.