

**SENTIMENTAL ANALYSIS FOR PRODUCT
FEEDBACK REVIEW**

*A
Project Report
Submitted in partial fulfilment of the
Requirements for the award of the Degree of*

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

By

1602-20-737-139, D. Kiran

1602-20-737-176, K. Sunil

Under the guidance of

Mrs. M. Sathya Devi

Assistant Professor



**Department of Information Technology
Vasavi College of Engineering (Autonomous)
ACCREDITED BY NAAC WITH 'A++' GRADE
(Affiliated to Osmania University)
Ibrahimbagh, Hyderabad-31 2024**

Vasavi College of Engineering (Autonomous)

ACCREDITED BY NAAC WITH 'A++' GRADE

(Affiliated to Osmania University)

Hyderabad-500 031

Department of Information Technology



DECLARATION BY THE CANDIDATE

We, **D. Kiran** and **K. Sunil** bearing hall ticket number, **1602-20-737-139** and **1602-20-737-176** hereby declare that the project report entitled **Sentimental Analysis for Product Feedback Review** under the guidance of **Mrs. M. Sathya Devi, Assistant Professor**, Department of Information Technology, Vasavi College of Engineering, Hyderabad, is submitted in partial fulfilment of the requirement for the award of the degree of **Bachelor of Engineering in Information Technology**

This is a record of bonafide work carried out by us and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

D. Kiran
1602-20-737-139

K. Sunil
1602-20-737-176

Vasavi College of Engineering (Autonomous)

ACCREDITED BY NAAC WITH 'A++' GRADE

(Affiliated to Osmania University)

Hyderabad-500 031

Department of Information Technology



BONAFIDE CERTIFICATE

This is to certify that the project entitled **Sentimental Analysis For Product Feedback Review** being submitted by **D. Kiran** and **K. Sunil** bearing , **1602-20-737-139** and **1602-20-737-176** in partial fulfilment of the requirements for the award of the degree of Bachelor of Engineering in Information Technology is a record of bonafide work carried out by them under my guidance.

Mrs. M. Sathya Devi
Assistant Professor
Internal Guide

Dr. K. Ram Mohan Rao
Professor & HOD, IT

External Examiner

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of the Main project would not have been possible without the kind support and help of many individuals. We would like to extend our sincere thanks to all of them.

It is with immense pleasure that we would like to take the opportunity to express our humble gratitude to **Mrs. M. Sathya Devi, Assistant Professor, Department of Information Technology** under whom we executed this project. We are also grateful to **Mrs. M. Sathya Devi, Assistant Professor, Department of Information Technology** for her guidance. Their constant guidance and willingness to share their vast knowledge made us understand this project and its manifestations in great depths and helped us to complete the assigned tasks.

We are very much thankful to **Dr. K. Ram Mohan Rao, Professor, Department of Information Technology**, for his kind support and for providing necessary facilities to carry out the work.

We wish to convey our special thanks to **Dr. S.V. Ramana, Principal of Vasavi College of Engineering and Management** for providing facilities. Not to forget, we thank all other faculty and non-teaching staff, and my friends who had directly or indirectly helped and supported me in completing my project in time.

ABSTRACT

Our feedback system introduces a systematic approach to sentiment analysis in product reviews, leveraging both video and audio modalities to extract nuanced emotional insights. Incorporating the Haar cascade algorithm for video analysis and the BERT model for audio processing, our method integrates these modalities to provide a holistic understanding of reviewer sentiment. By analyzing facial expressions, gestures, speech patterns, and vocal intonations, our approach captures the complex emotional context of product reviews, offering deeper insights into consumer perceptions and satisfaction. Our solution is adaptable to both offline and online product review systems, catering to diverse platform capabilities and user preferences. Through empirical evaluation and comparative analysis, we demonstrate the effectiveness and robustness of our approach across various product categories and review platforms. This research contributes to the advancement of sentiment analysis techniques and multimodal fusion, showcasing the potential for enhanced sentiment understanding in product review systems. Our findings offer practical implications for businesses seeking to harness consumer feedback effectively and inform decision-making processes.

TABLE OF CONTENTS

List of Figures	VII
List of Tables	VIII
List of Abbreviations	VIII
1 INTRODUCTION	1
1.1 Problem Statement	1
1.2 Proposed Method	2
1.3 Scope and Objectives of the Proposed work	4
2 LITERATURE SURVEY	5
3 PROPOSED WORK	10
3.1 Block Diagram	10
3.2 Algorithm	17
4 EXPERIMENTAL STUDY	19
4.1 Datasets	19
4.2 Software Requirements	19
4.3 Hardware Requirements	20
4.4 User Requirements	20
5 RESULTS AND ANALYSIS	21
5.1 Results	21
5.2 Analysis	24
6 CONCLUSION AND FUTURE SCOPE	25
REFERENCES	26
APPENDIX	27

LIST OF FIGURES

Fig. No	Description	Page No
1	Proposed method block diagram	2
2	Goggle Speech to Text API module	4
3	DCNN Layered Architecture	7
4	BERT Architecture	9
5	Facial Landmarks Points	10
6	Emotion Detection Probability	11
7	Video Sentiment Analysis block diagram	13
8	Audio Sentiment Analysis Flow diagram	14
9	Integrated Sentiment Analysis	15
10	Haar Features Calculation	17
11	Integral Image Creation	17
12	Adaboost Training	18
13	Cascading Classifiers Implementation	18
14	Video sentiment output of subject_1 emotion	21
15	Video sentiment output of subject_2 emotion	21
16	Video sentiment output of subject_3 emotion	21
17	Audio to Text conversion output into a .txt file	22
18	Integrated Final sentiment output	22
19	Integrated Final sentiment for Extremely positive and Neutral case	23

LIST OF TABLES

Fig.No	Description	Page No
1	Recall, Precision and Accuracy Comparisons	24

LIST OF ABBREVIATIONS

1. ML – Machine Learning
2. CNN – Convolutional Neural Network
3. DCNN – Deep Convolutional Neural Network
4. API - Application Programming Interface
5. FER - Facial Emotion Recognition
6. NLP - Natural Language Processing
7. BERT - Bidirectional Encoder Representations from Transformers
8. BiGRU - Bidirectional Gated Recurrent Unit network
9. BiLSTM - Bidirectional Long Short-Term Memory network
10. ROI – Region Of Interest

I INTRODUCTION

I.1 Problem Statement

In the age of e-commerce, product reviews play a crucial role in shaping consumer purchasing decisions. Online platforms are filled with user-generated reviews expressing opinions, sentiments, and experiences with various products and services. Sentiment analysis, the automated process of determining the emotional tone conveyed in text, has been extensively applied to analyze and categorize product reviews, enabling businesses to gain valuable insights into customer satisfaction, identify areas for improvement, and make data-driven decisions. Videos allow reviewers to convey their opinions with richer contextual information, including facial expressions, gestures, and product demonstrations, while audio recordings capture vocal intonations, speech patterns, and background sounds, all of which contribute to the overall sentiment conveyed. By incorporating both video and audio modalities into the sentiment analysis process, we can harness a more comprehensive understanding of the reviewer's emotional state and provide more nuanced insights into product perception and satisfaction. We propose an innovative approach to sentiment analysis in product reviews by integrating both video and audio modalities. Our method leverages the Haar cascade algorithm for facial detection and sentiment analysis in video content, enabling us to extract facial expressions and gestures indicative of different emotional states. Concurrently, we employ a deep learning model, specifically the BERT model, for audio sentiment analysis, capturing emotional cues from speech patterns and vocal intonations present in audio recordings. The integration of these two modalities allows us to capitalize on their complementary nature, combining visual and auditory cues to enhance the accuracy and robustness of sentiment analysis in product reviews. By fusing the outputs of the video-based Haar cascade algorithm and the audio-based BERT model, we aim to identify the most prominent emotional cues expressed by reviewers and provide a accurate assessment of product sentiment.

I.2 Proposed Method

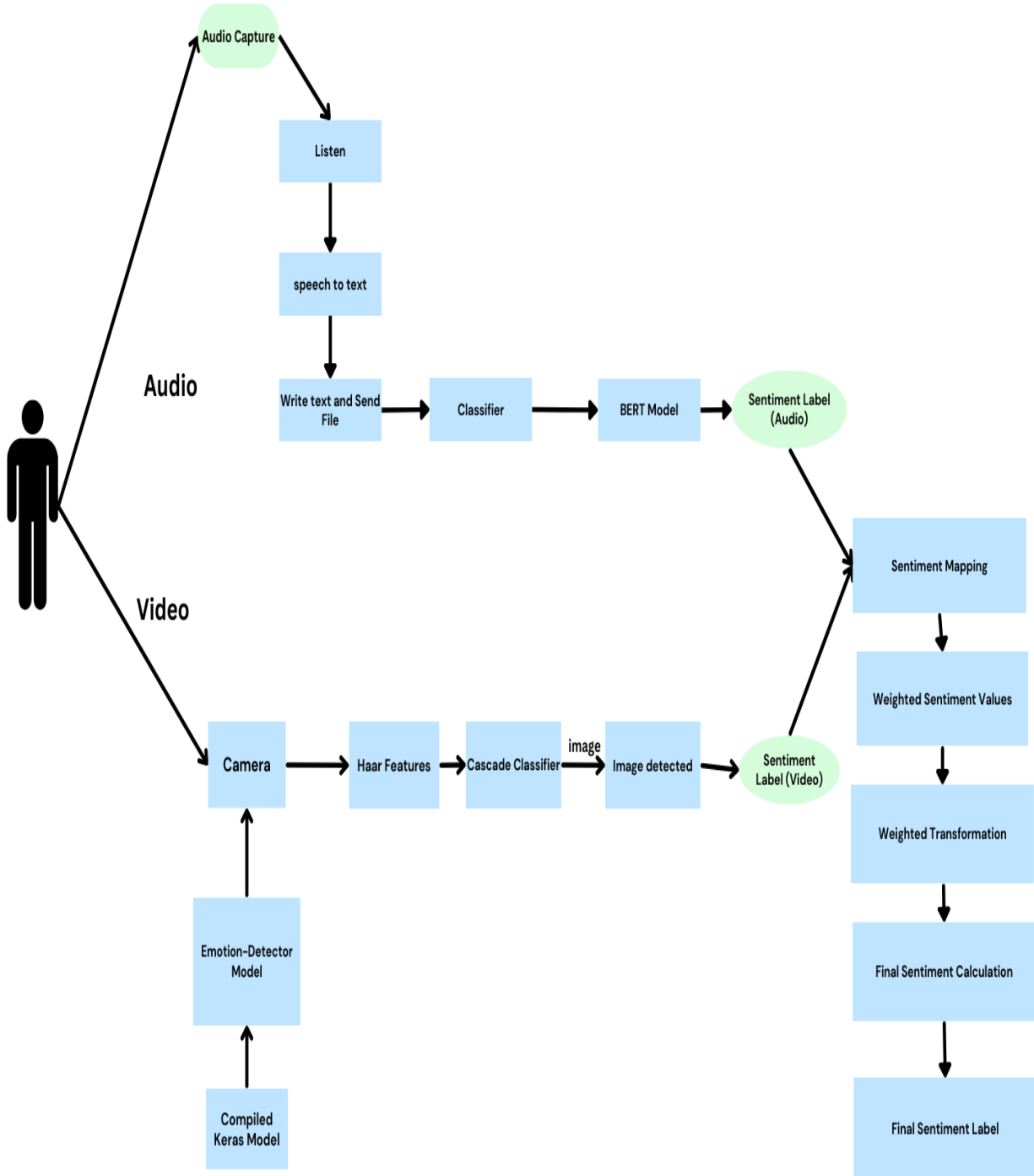


Fig 1 : Proposed method block diagram

The Data Acquisition serves as the cornerstone of the emotion detection system, facilitating the crucial task of gathering essential input data for subsequent emotion analysis, specifically in the domain of facial emotion detection. The module initiates

the process by retrieving video feeds or images containing human faces from diverse sources, including live video streams, recorded videos, or static images, tailored to the specific application requirements. Notably, in this instance, the data is sourced from Kaggle, a renowned platform recognized for hosting datasets suitable for machine learning and computer vision tasks. Once the data is acquired, the system's focus shifts to detecting facial features within each frame or image, which is efficiently carried out using the Haar Cascade algorithm. This algorithm is specifically trained to recognize patterns corresponding to facial features such as eyes, nose, and mouth, thereby laying the groundwork for subsequent analysis.

The Haar Cascade algorithm plays a pivotal role in the initial stages of the emotion detection pipeline by swiftly and accurately identifying crucial facial features within captured frames or images. Leveraging a cascade of simple Haar-like features derived from positive and negative samples during training, the algorithm distinguishes between regions of interest and non-interest in a hierarchical manner. This approach enables efficient localization of essential facial components within each frame, ensuring the generation of a robust foundational dataset for subsequent analysis in the emotion detection pipeline. The accuracy and efficiency of this initial stage significantly influence the reliability of emotion detection results in later stages of the system. Moreover, the integration of diverse datasets from Kaggle enriches the dataset's diversity and richness, contributing to a more comprehensive and robust emotion detection system capable of handling various scenarios and emotions effectively.

In our approach to analyzing audio data for sentiment using BERT, we implement a real-time audio-to-text conversion system powered by the Google API. This conversion process is essential as it transforms spoken content into textual format, enabling subsequent sentiment analysis using the BERT model. The Google API module depicted in the fig 4, efficiently converts incoming audio streams into text, capturing the spoken content accurately and promptly. This step is crucial as it bridges the gap between audio data, which is inherently unstructured, and the text-based analysis required by the BERT model.

Once the audio data is converted into text, we integrate a pre-trained BERT model into our system for sentiment analysis. BERT is a state-of-the-art language

representation model that excels in understanding contextual variables and relationships within text data. By incorporating BERT, we can capture the sentiment expressed in the transcribed audio text accurately. The BERT model's bidirectional nature allows it to consider the entire context of the input text, resulting in more advanced and contextually informed sentiment analysis outcomes. This integration of advanced NLP techniques with real-time audio transcription enhances the capability of our system to provide accurate and timely sentiment insights from spoken content, enabling applications across various domains such as customer service, market research, and voice-driven user interfaces.

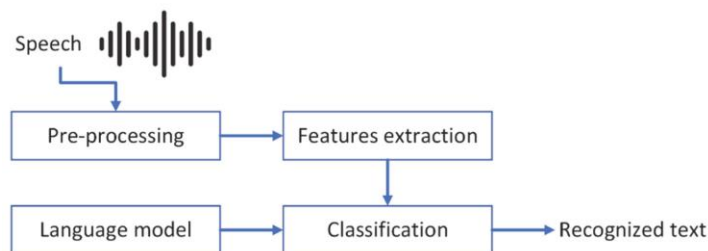


Fig 2 : Goggle Speech to Text API module

I.3 Scope and Objectives of the Proposed work

The scope of this study encompasses a systematic approach to achieve a sentiment analysis system that integrates both video (facial emotion detection n) and audio (speech-to-text sentiment analysis) modalities to derive sentiment labels from user input, specifically for feedback and review systems.

Input from a camera is required for video sentiment analysis. Furthermore, the sentimental analysis works on facial expressions of the subject not more than one. When multiple subjects are taken as input from the camera then for each frame obtained only one clearer face identified will be given the sentimental analysis.

Input from microphone for audio sentiment analysis is also required. The proposed work audio module can work perfectly when the microphone is preferably in-built but not any external microphone is attached. For the proper working of the google Api for audio to text conversion we need a stable internet connection.

The scope of the study is to integrate two separate sentimental analysis modules into one to gain higher accuracies than that of individually obtained values. The scope of

the study also adheres to the versions compatibility of the python libraries use and mentioned.

The Objectives of the study is to represent five sentiments namely “Extremely Negative”, “Negative”, “Neutral”, “Positive” and “Extremely Positive” relating to the product feedback from the user received. Further more to integrate the sentimental analysis obtained from the individual audio and video sentimental analysis. Comparative study of the individual accuracies with the integrated accuracy.

II LITERATURE SURVEY

[1] Sentiment Analysis for Video on Demand Application User Satisfaction with Long Short Term Memory Model [1]

Authors: Gina Khayatun Nufus, Mustafid Mustafid , dan Rahmat Gernowo

Proposed methodology :

The methodology involves utilizing Word2vec, a word embedding method employing skip-gram or continuous bag-of-words (CBOW) models, to generate vector representations for words from large datasets. Long Short-Term Memory (LSTM), a type of recurrent neural network (RNN), is employed to model sequential data with its memory structure and three gates. The LSTM is applied to aspect-level sentiment classification using equations, involving forget gates, input gates, and output gates. The LSTM model is integrated with a Word2vec embedding layer, and the processed vectors serve as inputs to the LSTM layer. The LSTM output is then fed through a fully connected layer with ReLU activation, followed by a softmax activation at the output layer for sentiment classification.

[II]Sentiment analysis on images using convolutional neural networks based Inception-V3 transfer learning approach [2]

Authors : Gaurav Meenaa , Krishna Kumar Mohbeya , Sunil Kumar

Proposed Methodology :

A CNN-based Inception-v3 architecture is employed for emotion detection and classification. The datasets CK+, FER2013, and JAFFE are used in this process. The findings are also compared with various well-known machine learning approaches, and the results obtained by the suggested model are superior. The foundation of this research is a set of pre-trained deep CNN models and a method called transfer learning. This research aims to determine which of the Inception-V3 pre-trained deep CNN models is most suitable for image-based sentiment analysis.

[III]Facial Emotion Recognition Using Transfer Learning in the Deep CNN [3]

Authors : M. A. H. Akhand Shuvendu Roy , Nazmul Siddique , Md Abdus Samad Kamal and Tetsuya Shimamura

Proposed Methodology :

This study addresses the limitations of conventional facial emotion recognition (FER) methods by proposing a very deep Convolutional Neural Network (DCNN) model using Transfer Learning. The proposed model leverages pre-trained DCNNs (VGG-16, VGG-19, ResNet-18, ResNet-34, ResNet-50, ResNet-152, Inception-v3, and DenseNet-161), replacing their upper layers and fine-tuning for FER on KDEF and JAFFE datasets. Unlike previous methods focusing only on frontal views, the proposed approach handles diverse facial angles. Achieving remarkable accuracies, the DenseNet-161 model attains 96.51% and 99.52% FER accuracy on KDEF and JAFFE datasets, respectively, demonstrating superior emotion detection proficiency and potential for real-life applications.

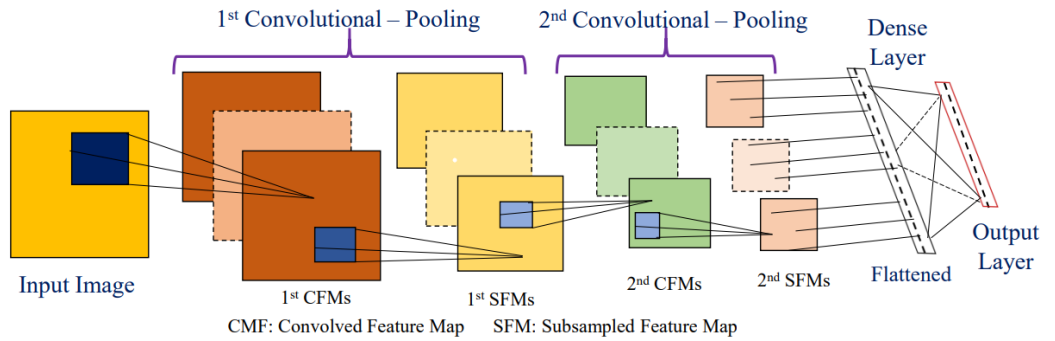


Fig 3 : DCNN Layered Architecture

[IV] Amazon Product Sentiment Analysis using RapidMiner [4]

Authors : Nur Hasifah A Razak, Muhammad Firdaus Mustapha, Nur Ami rah Marzuki, Nur Saidatul Sa'adiah Tajul Othamany.

Proposed Methodology : The proposed methodology for the research paper "Amazon Product Sentiment Analysis using RapidMiner" involves a systematic approach to analyze sentiment in Amazon product reviews across various categories like Health and Beauty, Electronics, and Toys and Games. The initial phase encompasses defining clear research objectives, which include evaluating sentiment across different product categories and assessing its impact on product attitudes. Subsequently, the raw data is acquired from the DataWorld website, and thorough exploration is conducted to understand its structure and potential challenges. Data preparation follows, involving cleansing the data, selecting relevant attributes, and normalizing and preprocessing it for consistency.

Validation of the results is conducted using cross-validation techniques to ensure the robustness of the findings. Finally, the conclusions are drawn based on the summarized findings, discussing the effectiveness of the proposed methodology in sentiment analysis of Amazon product reviews. The implications, limitations, and potential areas for future research are also deliberated upon to provide a comprehensive understanding of the research outcomes. This methodology provides a structured framework for conducting sentiment analysis research, enabling detailed analysis and interpretation of sentiment trends in Amazon product reviews.

[V] Sentiment Analysis Using Bert Model [5]

Authors : Dorca Manuel-Ilie , Pitic Antoniu Gabriel , Crețulescu Radu George.

Proposed Methodology :

The methodology employed in the "Sentiment Analysis Using BERT Model" paper involves a comprehensive approach to sentiment analysis that integrates concepts from deep learning and advanced technology. It starts by establishing a theoretical framework that goes beyond traditional sentiment analysis methods, aiming to develop a versatile understanding of emotions in textual content. Key components of the methodology include topic annotation to identify main subjects, emotion annotation with dynamic lexicon expansion to assign emotional scores to words, and subsequent emotion categorization to classify sentiment into positive, negative, or neutral groups. The methodology also

focuses on customizing a neural network architecture using the BERT model designed for sentiment analysis, involving hyperparameter tuning, data preprocessing, and systematic model evaluation. Overall, the methodology emphasizes nuanced emotional understanding, robust model configuration, and systematic evaluation to optimize sentiment analysis performance across diverse datasets and domains.

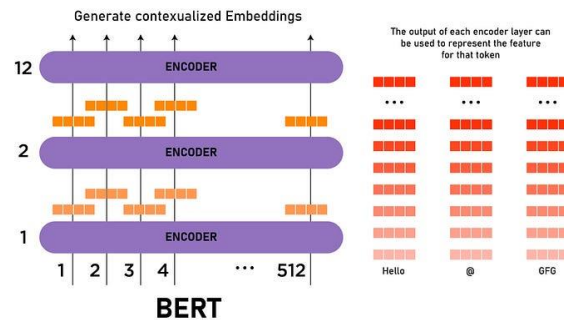


Fig 4 : BERT Architecture

[VI] Sentiment analysis classification system using hybrid BERT models [6]

Authors : Amira Samy Talaat

Proposed Methodology :

The paper introduces a sentiment analysis framework leveraging hybrid BERT models, blending BERT with BiLSTM and BiGRU algorithms. It outlines the significance of sentiment analysis in understanding public opinions, especially on social media platforms. After reviewing related literature, the methodology involves preprocessing datasets and comparing classical ML algorithms with the proposed hybrid models. Text cleaning includes normalization and emoji removal. Eight hybrid models are proposed, combining BERT variants with different BiLSTM and BiGRU configurations. Testing on three datasets demonstrates superior performance of hybrid models over classical ML methods. Overall, the paper provides a comprehensive framework for sentiment analysis using hybrid BERT models, showcasing promising results and avenues for further exploration.

III PROPOSED WORK

3.1 Block Diagram

1. Multimodal Data Processing

Video-based Emotion Detection:

Utilizing computer vision techniques to analyze facial expressions from video inputs. Detecting and recognizing key facial features (eyes, nose, mouth) to infer emotional states. Classifying emotions into predefined categories (e.g., Negative, Positive, Neutral).

Facial Feature analysis:

After the initial data acquisition, the Facial Feature Analysis module takes center stage in the emotion detection system, playing a crucial role in identifying and analyzing distinctive facial features. In Fig. the Haar Cascade algorithm, a machine learning object detection method, proves to be highly effective due to its proficiency in recognizing patterns. The Haar Cascade algorithm operates by employing a series of trained classifiers for specific facial features, such as eyes, nose, and mouth.

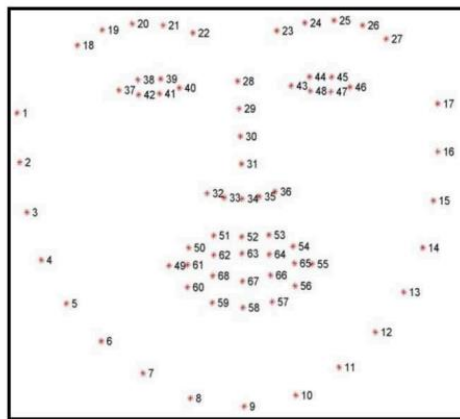


Fig 5: Facial Landmarks Points

As the Haar Cascade algorithm processes each frame or image from the acquired data, it efficiently scans through the image using a sliding window approach. Once the Haar Cascade algorithm identifies and localizes specific facial features, the Facial Feature Analysis module engages in the extraction and interpretation of these features. This involves capturing the spatial relationships, sizes, and configurations of the detected facial components. By understanding the unique arrangement of features on an individual's face, this module establishes a foundation for subsequent emotion detection. At each step, it applies the cascade of classifiers to identify regions of interest corresponding to the predefined facial features as in Fig 5. During the training phase, the algorithm learns to distinguish positive samples (regions containing the target facial feature) from negative samples (regions without the target feature). The resulting set of classifiers forms a cascade, and each stage in the cascade further refines the detection process.

Emotion Detection:

The Emotion Detection module serves as the central component of the system, leveraging the identified facial features to infer the emotional state of the individual. Building upon the localized facial features obtained from the Facial Feature Analysis module, this stage focuses on associating these features with specific emotional expressions. The Haar Cascade algorithm, with its proficiency in pattern recognition, plays a key role in localizing facial regions indicative of particular emotions. Fig. 3 depicts Emotion Detection Probability.

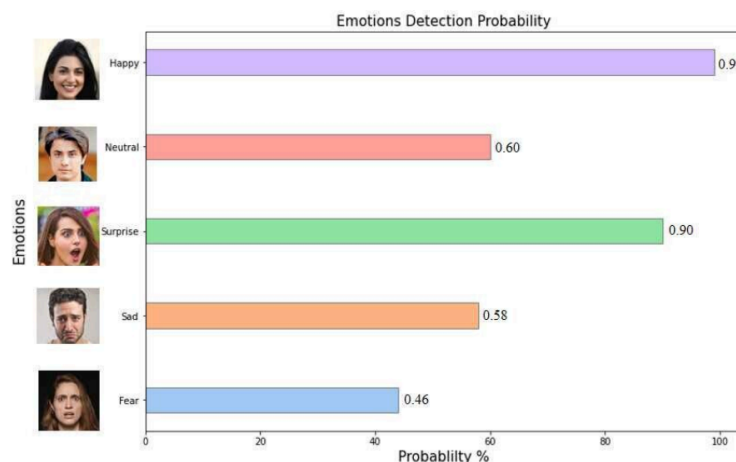


Fig 6 : Emotion Detection Probability

Audio-based Sentiment Analysis:

Implementing speech-to-text conversion to transcribe spoken feedback into textual format. Applying natural language processing (NLP) models (e.g., BERT) to analyze sentiment from the transcribed text. Categorizing sentiment into labels such as Extreme Positive, Positive, Negative, Extreme Negative, Neutral.

2. Integration of Modalities

Combined Analysis:

Integrating video and audio analysis to derive comprehensive sentiment insights. Synthesizing results from both modalities to provide a holistic understanding of user feedback.

Video Sentiment analysis (Train model):

Pre-trained MobileNet:

Utilizes a pre-trained MobileNet model with weights from ImageNet. MobileNet serves as the base convolutional neural network for feature extraction.

Custom Top Layers:

Adds custom layers (GlobalAveragePooling, Dense) on top of the MobileNet base. These layers constitute the "head" of the model and are responsible for classification.

Model Compilation:

Compiles the Keras model by specifying loss function, optimizer, and evaluation metrics. The compiled model is ready for training and evaluation.

Training and Validation Data:

Specifies the directories containing training and validation images. ImageDataGenerator is used for real-time data augmentation and preprocessing.

Callbacks:

Includes callbacks like ModelCheckpoint, EarlyStopping, and ReduceLROnPlateau. These callbacks monitor the training process and adjust model behavior accordingly (e.g., saving best model weights, early stopping, learning rate reduction).

Training Process:

The `model.fit_generator()` function trains the compiled model using the specified data generators. Training proceeds over a defined number of epochs with batch-wise processing.

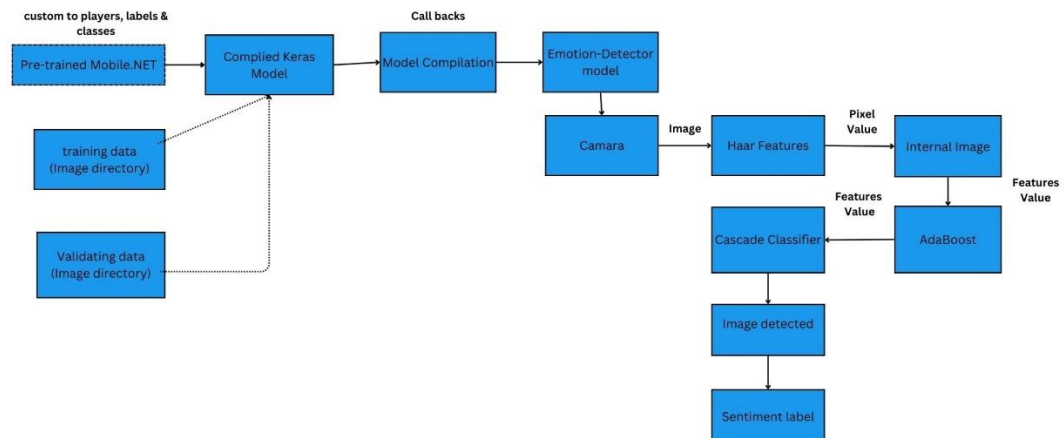


Fig 7: Video Sentiment Analysis block diagram

Video Sentiment analysis (Test model):

This involves several interconnected components that work together to perform real-time emotion detection using a pre-trained deep learning model. Here's an explanation of the block diagram mentioned in Fig 7.

Video Capture (OpenCV):

Starts by initializing a video capture using OpenCV, which captures frames from the default camera .

Face Detection:

For each frame captured from the video feed:

The frame is converted to grayscale , to simplify processing. The Haar cascade classifier is used to detect faces in the grayscale frame

Detected faces are enclosed within rectangles (`cv2.rectangle`) on the original color frame for visualization.

ROI Extraction:

For each detected face:

The region of interest (ROI) is extracted from the grayscale frame based on the detected face coordinates . The ROI is resized to a fixed size (48x48 pixels) using bilinear interpolation .

Preprocessing for Prediction:

If a valid ROI is extracted (i.e., the sum of its pixels is not zero):

The ROI is normalized by scaling its pixel values to the range $[0, 1]$. The ROI is converted to an array and reshaped to match the input shape expected by the deep learning model .

Emotion Prediction:

The pre-trained emotion detection model (classifier) predicts the emotion label for the processed ROI .

The predicted probabilities for each emotion class are obtained (preds). The emotion class with the highest probability is determined , and the corresponding label is retrieved from class_labels. The predicted emotion label is overlaid on the frame using OpenCV's cv2.putText function.

Audio Sentiment analysis (BERT model):

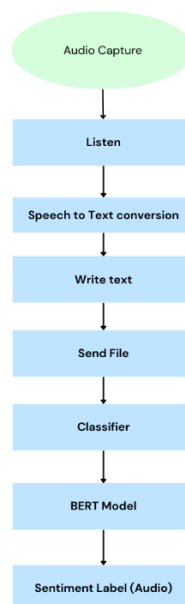


Fig 8: Audio Sentiment Analysis Flow Datagram

This module is designed to perform two main tasks: speech-to-text conversion and sentiment analysis. It uses various libraries such as speech_recognition, pyttsx3, and Hugging Face's transformers pipeline. The SpeakText function enables text-to-speech conversion using the pyttsx3 library, allowing the system to speak out the recognized text.

The program continuously listens to the user through the microphone, captures the spoken words, and converts them to text using Google's speech recognition (recognize_google function). The recognized text is then processed further: it is written into a file for logging purposes (audiototext.txt), passed to a sentiment analysis model (classifier), and then spoken back to the user using the SpeakText function. The sentiment analysis model predicts the sentiment (e.g., positive, negative) of the spoken text.

Integrated Sentimental Analysis :

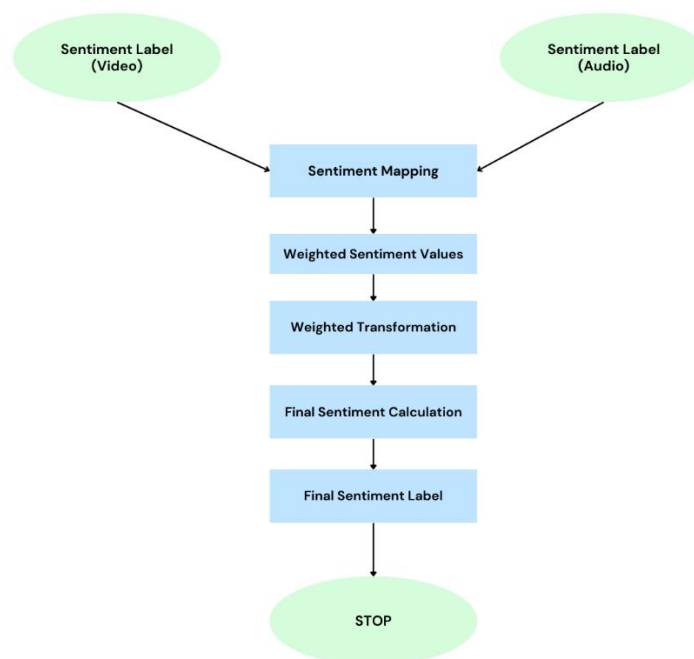


Fig 9: Integrated Sentimental Analysis

The block diagram for this module involves a sentiment fusion approach where sentiment values from audio and video sources are combined to produce a final sentiment classification.

Sentiment Mapping:

Sentiment labels ('Extremely Negative', 'Negative', 'Neutral', 'Positive', 'Extremely Positive') are mapped to numerical values (0, 1, 2, 3, 4) using a dictionary (sentiment_mapping). This mapping enables conversion between sentiment labels and numerical values for calculation.

Weighted Sentiment Values:

Sentiment values (audio_value and video_value) are calculated based on the sentiment labels (label_audio_senti and label_video) obtained from audio and video sentiment analysis processes, respectively. Each sentiment value is multiplied by a predefined weight (audio_weight and video_weight). These weights determine the relative importance of audio versus video sentiment in the final sentiment fusion.

Final Sentiment Calculation:

The weighted sentiment values (audio_value and video_value) are combined to produce a composite sentiment value (final_temp). The max() function is used to ensure that the final sentiment value (final_temp) does not exceed the defined sentiment range (0 to 4). The composite sentiment value (final_temp) is rounded up (math.ceil()) to the nearest integer to get the final sentiment score (final).

Sentiment Label Conversion:

The final sentiment score (final) is converted back into a sentiment label using the senti_final() function. This function maps the numerical sentiment score to its corresponding sentiment label ('Extremely Negative', 'Negative', 'Neutral', 'Positive', 'Extremely Positive').

The sentiment fusion process where sentiment values from different sources (audio and video) are combined using weighted averages to generate a comprehensive final sentiment classification. This approach allows for a more robust sentiment analysis by leveraging inputs from multiple modalities (audio and video) and accounting for their relative importance in the final sentiment determination.

3.2 Algorithm

Haar Cascade Algorithm :

This involves Four Stages that include:

1. Haar Features Calculation: Gathering the Haar features is the first stage. Haar features are nothing but a calculation that happens on adjacent regions at a certain location in a separate detecting window. The calculation mainly includes adding the pixel intensities in every region and between the sum differences calculation.

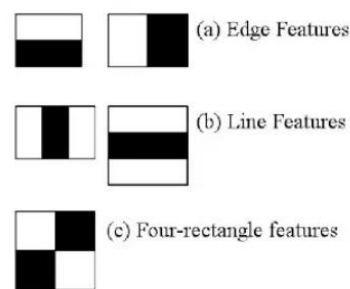


Fig 10: Haar Features Calculation

2. Integral Image Creation: Creating Integral Images reduces the calculation. Instead of calculating at every pixel, it creates the sub-rectangles, and the array references. The only important features are those of an object, and mostly all the remaining Haar features are irrelevant in the case of object detection. Here Adaboost enters the picture.

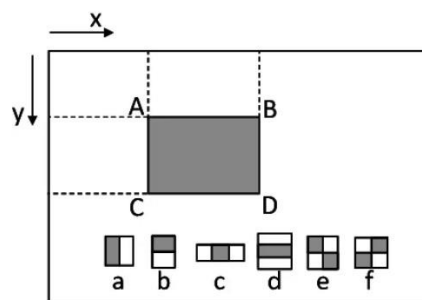


Fig 11: Integral Image Creation

3. Adaboost Training:

The "weak classifiers" are combined by Adaboost Training to produce a "strong classifier" that the object detection method can use. This essentially consists of selecting useful features and teaching classifiers how to use them. By moving a window across the input image and computing the Haar characteristics for each part of the image, weak learners are created. This distinction stands in contrast to a threshold that can be trained to tell objects apart from non-objects. These are "weak classifiers," but an accurate strong classifier needs many Haar properties.

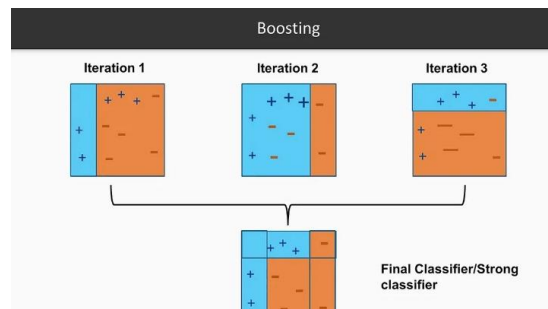


Fig 12: Adaboost Training

In the final step, weak learners might be combined with strong learners.

4. Cascading Classifiers Implementation:

Every stage at this point is a group of inexperienced students. Boosting trains weak learners, resulting in a highly accurate classifier from the average prediction of all weak learners. It depends based upon the prediction. The classifier decides for indication of an object that was found positive or moved to the next region, i.e., negative. Because most windows do not contain anything of interest, stages are created to reject negative samples as quickly as feasible. Because classifying an object as a non-object would significantly hurt your object detection system, having a low false negative rate is crucial.

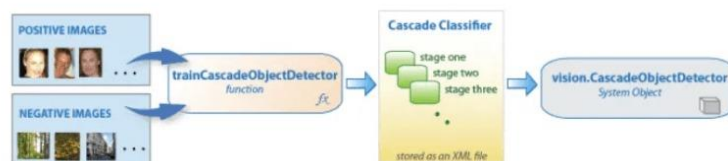


Fig 13: Cascading Classifiers Implementation

IV EXPERIMENTAL SETUP

4.1 DATASETS

Challenges in Representation Learning : Kaggle Dataset

The data consists of 48x48 pixel grayscale images of faces. The faces have been automatically registered so that the face is more or less centered and occupies about the same amount of space in each image. The task is to categorize each face based on the emotion shown in the facial expression in to one of seven categories (0=Extremely Negative, 1=Negative, 2=Neutral, 3=Positive, 4=Extremely Positive, 5=Neutral, 6=Extremely Negative). The train.csv contains two columns, "emotion" and "pixels". The "emotion" column contains a numeric code ranging from 0 to 6, inclusive, for the emotion that is present in the image. The "pixels" column contains a string surrounded in quotes for each image. The contents of this string a space-separated pixel values in row major order. test.csv contains only the "pixels" column and your task is to predict the emotion column.

The training set consists of 28,709 examples. The public test set used for the leaderboard consists of 3,589 examples. The final test set, which was used to determine the winner of the competition, consists of another 3,589 examples.

This dataset was prepared by Pierre-Luc Carrier and Aaron Courville, as part of an ongoing research project. They have graciously provided the workshop organizers with a preliminary version of their dataset to use for this contest.

<https://www.kaggle.com/competitions/challenges-in-representation-learning-facial-expression-recognition-challenge/data>

4.2 SOFTWARE REQUIREMENTS

- **Operating System:** Windows 10 or above
- **Programming Language:** Python 3.6 or above
- **Model :** Video Sentiment Analysis – MobileNet Architecture and Haar Cascade
Audio Sentiment Analysis – BERT Model
- **Libraries:** Keras versions- 2.12.0, OpenCV, NumPy version1.23.5, Tensorflow version-2.12.0, Transformers verion-4.39.0,Pyttx3, SpeechRecognition(Google API)

4.3 HARDWARE REQUIREMENTS

4.3.1 Processor

A CPU with 4 or more cores which speeds up model training and inference.

4.3.2 Random Access Memory

16GB or more RAM necessary to handle the data and model during training.

4.3.3 Storage

Fast Storage of at least 5 GB

4.4 USER REQUIREMENTS

Camera Module : A working camera model is mandatory. Built-in laptop webcams or external USB webcams are common choices. On some systems, you may need appropriate permissions to access the camera device. Ensure that your module has permission to use the camera and display the camera.

Microphone : A working microphone is mandatory . External microphones reduce the quality of audio ,Hence it should be avoided. Ensure that your module has permission to use the microphone.

V RESULTS AND ANALYSIS

In this section, we unveil the outcomes of our study, focusing on the criteria set for the Product Sentiment Feedback System. These results shed light on the performance and efficacy of the system, revealing insights into its usability, scalability, and accuracy. Through a comprehensive analysis, we aim to evaluate the system's effectiveness in delivering sentimental information correlating to the product and ensuring a seamless user experience. The findings presented here are pivotal in understanding the feedback system's suitability for addressing diverse user needs and guiding future enhancements to optimize its performance.



Fig 14: Video sentiment output of subject_1 emotion

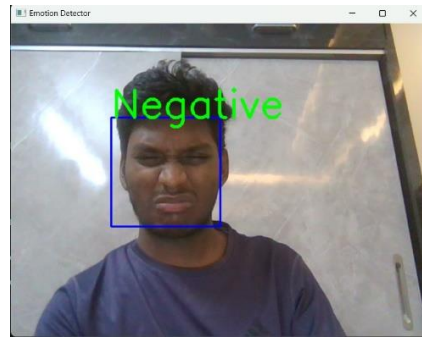


Fig 15: Video sentiment output of subject_2 emotion

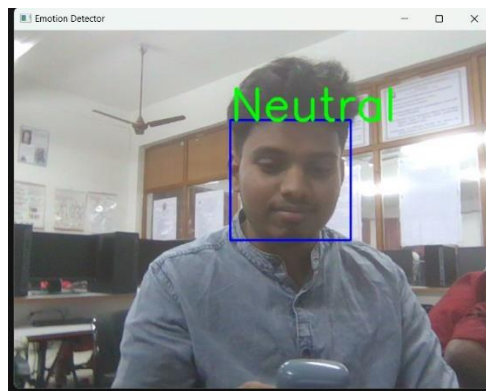
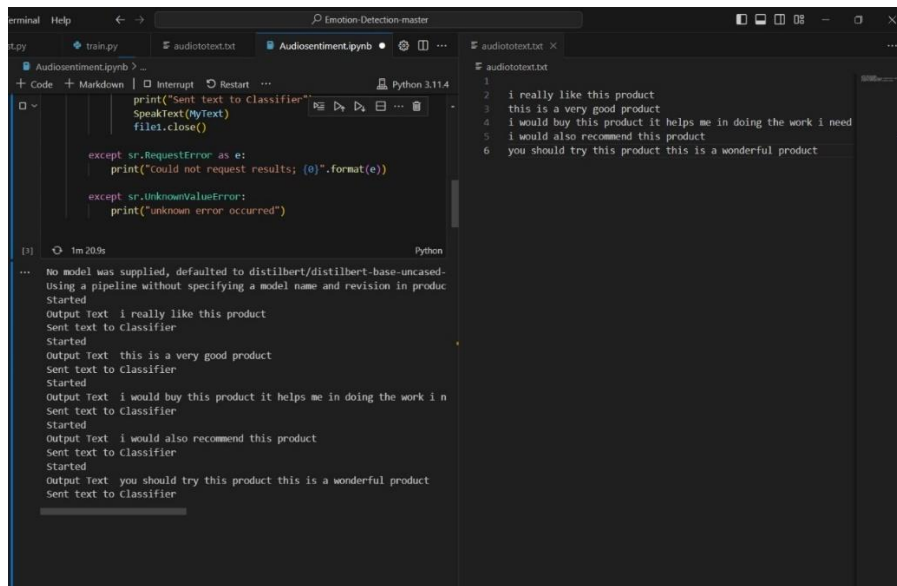


Fig 16: Video sentiment output of subject_3 emotion



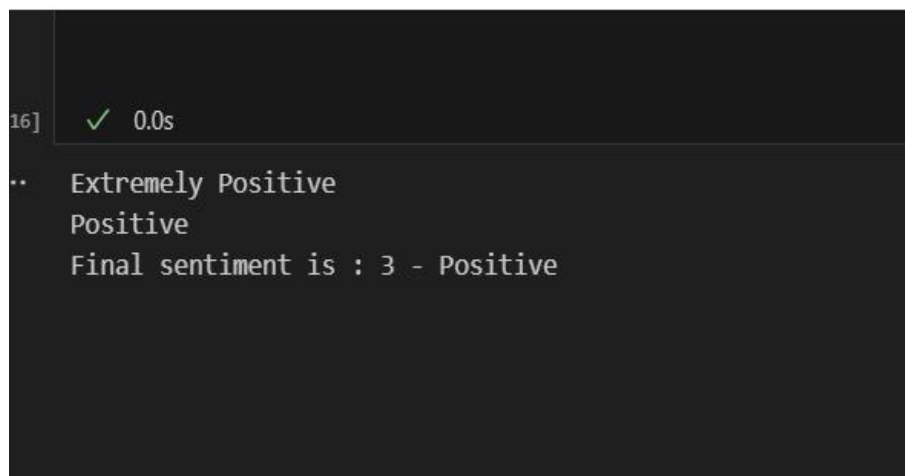
```
terminal Help < -> Emotion-Detection-master
Audiosentiment.ipynb > -
+ Code + Markdown | Interrupt Restart ... Python 3.11.4
print("Sent text to Classifier")
SpeakText(MyText)
file.close()

except sr.RequestError as e:
    print("could not request results; {0}".format(e))

except sr.UnknownValueError:
    print("unknown error occurred")

[5] 1m 20.9s Python
...
No model was supplied, defaulted to distilbert/distilbert-base-uncased-
Using a pipeline without specifying a model name and revision in produc
Started
Output Text i really like this product
Sent text to Classifier
Started
Output Text this is a very good product
Sent text to Classifier
Started
Output Text i would buy this product it helps me in doing the work i n
Sent text to Classifier
Started
Output Text i would also recommend this product
Sent text to Classifier
Started
Output Text you should try this product this is a wonderful product
Sent text to Classifier
```

Fig 17: Audio to Text conversion into a .txt file



```
16] ✓ 0.0s
.. Extremely Positive
Positive
Final sentiment is : 3 - Positive
```

Fig 18: Integrated Final sentiment output for positive case

```
[13] Python
... Extremely Positive
      Neutral
      2.4
      Final sentiment is : 3 - Neutral
```

Fig 19: Integrated Final sentiment for Extremely positive - Neutral case

ANALYSIS

Table 1: Analysis on individual and combined accuracies

Emotions	Precision		Recall		Accuracy (Individual)		Accuracy (Combined)
	Video (Haar)	Audio (Bert)	Video (Haar)	Audio (Bert)	Video (Haar)	Audio (Bert)	Video + Audio
Extremely negative	0.58	0.91	0.56	0.97	0.96	0.97	0.96
Negative	0.50	6.95	0.62	0.90	0.96	0.99	0.97
Neutral	0.60	0.88	0.66	0.85	0.97	0.99	0.98
Positive	0.87	0.98	0.81	0.96	0.97	0.99	0.98
Extremely positive	0.53	0.91	0.80	0.97	0.96	0.97	0.96

On combining video and audio modalities can lead to improved sentiment analysis results by leveraging complementary information. Facial expressions and spoken content often convey correlated emotional cues. Fusion of modalities may enhance accuracy and robustness, especially in scenarios where one modality alone may be insufficient or ambiguous.

Comparison of Accuracies:

Individual Modality Accuracies:

Haar Cascade (Video): Moderate to Good Accuracy.

BERT Model (Audio): High Accuracy.

Combined Modality Accuracy:

The accuracy of combined sentimental analysis can potentially exceed that of individual modalities:

Video provides visual cues (facial expressions) while audio provides verbal cues (spoken content).

Combining these modalities can capture a more holistic representation of the user's emotional state.

VI CONCLUSION AND FUTURE SCOPE

The integration of traditional video-based sentiment analysis using Haar cascade for facial emotion detection with audio-based sentiment analysis using the BERT model represents a promising approach to comprehensive emotion recognition for product feedback review systems .

Advantages of Integration:

Combines visual (facial expressions) and auditory (spoken content) cues for a more holistic understanding of emotions.

Leverages the strengths of each modality to compensate for their individual limitations.

Enhances accuracy and robustness by integrating multiple sources of emotional information.

Improved accuracy and reliability in emotion detection tasks.

Applicability in various domains such as customer feedback analysis, mental health monitoring, and human-computer interaction systems.

Future Scope:

Enhanced Integration Techniques:

Explore advanced fusion techniques (e.g., deep learning-based multimodal fusion) to integrate video and audio modalities more effectively.

Model Optimization:

Optimizing the Haar cascade model for better detection of subtle facial expressions under low lighting conditions. Also able to identify sentiments of multiple faces and getting a group sentiment regarding a product or any other domain

Multimodal Dataset Development:

Develop and curate multimodal datasets that include synchronized video and audio data for training and evaluation.

Cross-Domain Applications:

Extend the system's capabilities to analyze emotions across different domains, such as analyzing emotions in music or gestures.

Language Considerations:

Expanding the BERT model to multi-lingual datasets allows sentimental analysis on different languages.

REFERENCES

- [1] Gina Khayatun Nufus, Mustafid Mustafid, and Rahmat "Sentiment Analysis for Video on Demand Application User Satisfaction with Long Short Term Memory Model"
- [2] Gaurav Meena, Krishna Kumar Mohbey, Sunil Kumar "Sentiment analysis on images using convolutional neural networks based Inception-V3 transfer learning approach"
- [3] M.A.H. Akhand, Shuvendu Roy, Nazmul Siddique and Tetsuya Shimamura "Facial Emotion Recognition Using Transfer Learning in the Deep CNN"
<https://www.mdpi.com/2079-9292/10/9/1036>
- [4] Nur Hasifah A Razak, Muhammad Firdaus Mustapha. "Amazon Product Sentiment Analysis using RapidMiner"
- [5] Dorca Manuel-Ilie, Pitic Antoniu Gabriel, Crețulescu Radu George "Sentiment Analysis Using Bert Model"
https://www.researchgate.net/publication/376670839_Sentiment_Analysis_Using_Bert_Model
- [6] Amira Samy Talaat. "Sentiment analysis classification system using hybrid BERT models".
<https://journalofbigdata.springeropen.com/articles/10.1186/s40537-023-00781-w>
- [7] S. S. A. B. U. Rahul Ramachandran, "Exploring the relationship between emotionality and product star ratings in online reviews,"
- [8] Mika V. Mantyla, Daniel Graziotin and Miikka Kuutila, "The Evolution of Sentiment Analysis-A Review of Research Topics".
- [9] Sentiment Analysis, Available at: <https://insightsatlas.com/sentiment-analysis/>
- [10] Sentiment Analysis Explained, Available at: <https://www.lexalytics.com/technology/sentiment-analysis>
- [11] <https://github.com/ShivamGaurUQ/Sentiment-Analysis-of-Amazon-product-reviews>

APPENDIX

Test.py

```
#USAGE : python test.py

from keras.models import load_model

from time import sleep

from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.preprocessing import image

import cv2

import numpy as np

face_classifier = cv2.CascadeClassifier('.\haarcascade_frontalface_default.xml')
classifier =load_model('.\Emotion_Detection.h5')

class_labels = ['Angry','Positive','Neutral','Negative','Extremely positive']

cap = cv2.VideoCapture(0)

while True:

    # Grab a single frame of video

    ret, frame = cap.read()

    labels = []

    gray = cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)

    faces = face_classifier.detectMultiScale(gray,1.3,5)

    for (x,y,w,h) in faces:

        cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),2)

        roi_gray = gray[y:y+h,x:x+w]

        roi_gray = cv2.resize(roi_gray,(48,48),interpolation=cv2.INTER_AREA)

        if np.sum([roi_gray])!=0:

            roi = roi_gray.astype('float')/255.0

            roi = img_to_array(roi)

            roi = np.expand_dims(roi,axis=0)

            preds = classifier.predict(roi)[0]

            print("\nprediction = ",preds)

            label=class_labels[preds.argmax()]

            print("\nprediction max = ",preds.argmax())

            print("\nlabel = ",label)

            label_position = (x,y)

            cv2.putText(frame,label,label_position,cv2.FONT_HERSHEY_SIMPLEX,2,(0,255,0),3)
```

```

else:
    cv2.putText(frame,'No Face Found',(20,60),cv2.FONT_HERSHEY_SIMPLEX,2,(0,255,0),3)
    print("\n\n")
cv2.imshow('Emotion Detector',frame)
if cv2.waitKey(1) & 0xFF == ord('q'):
    break
cap.release()
cv2.destroyAllWindows()

```

Train.py

```

from keras.applications import MobileNet
from keras.models import Sequential,Model
from keras.layers import Dense,Dropout,Activation,Flatten,GlobalAveragePooling2D
from keras.layers import Conv2D,MaxPooling2D,ZeroPadding2D
from keras.layers import BatchNormalization
from tensorflow.keras.preprocessing.image import ImageDataGenerator
# MobileNet is designed to work with images of dim 224,224
img_rows,img_cols = 224,224
MobileNet = MobileNet(weights='imagenet',include_top=False,input_shape=(img_rows,img_cols,3))
# Here we freeze the last 4 layers
# Layers are set to trainable as True by default
for layer in MobileNet.layers:
    layer.trainable = True
# Let's print our layers
for (i,layer) in enumerate(MobileNet.layers):
    print(str(i),layer.__class__.__name__,layer.trainable)
def addTopModelMobileNet(bottom_model, num_classes):
    """creates the top or head of the model that will be
    placed ontop of the bottom layers"""
    top_model = bottom_model.output
    top_model = GlobalAveragePooling2D()(top_model)
    top_model = Dense(1024,activation='relu')(top_model)
    top_model = Dense(1024,activation='relu')(top_model)
    top_model = Dense(512,activation='relu')(top_model)

```

```

top_model = Dense(num_classes,activation='softmax')(top_model)

return top_model

num_classes = 5

FC_Head = addTopModelMobileNet(MobileNet, num_classes)

model = Model(inputs = MobileNet.input, outputs = FC_Head)

print(model.summary())

train_data_dir = '/Users/kiran/AppData/Local/Programs/Python/Python311/Emotion-Detection-master/challenges-in-representation-learning-facial-expression-recognition-challenge/fer2013'

validation_data_dir = '/Users/kiran/AppData/Local/Programs/Python/Python311/Emotion-Detection-master/challenges-in-representation-learning-facial-expression-recognition-challenge/fer2013'

train_datagen = ImageDataGenerator(

    rescale=1./255,

    rotation_range=30,

    width_shift_range=0.3,

    height_shift_range=0.3,

    horizontal_flip=True,

    fill_mode='nearest'

)

validation_datagen = ImageDataGenerator(rescale=1./255)

batch_size = 32

train_generator = train_datagen.flow_from_directory(

    train_data_dir,

    target_size = (img_rows,img_cols),

    batch_size = batch_size,

    class_mode = 'categorical'

)

validation_generator = validation_datagen.flow_from_directory(

    validation_data_dir,

    target_size=(img_rows,img_cols),

    batch_size=batch_size,

    class_mode='categorical')

from keras.optimizers import RMSprop,Adam

from keras.callbacks import ModelCheckpoint,EarlyStopping,ReduceLROnPlateau

checkpoint = ModelCheckpoint(

    'emotion_face_mobilNet.h5',

    monitor='val_loss',

```

```

        mode='min',
        save_best_only=True,
        verbose=1)
earlystop = EarlyStopping(
    monitor='val_loss',
    min_delta=0,
    patience=10,
    verbose=1,restore_best_weights=True)

learning_rate_reduction = ReduceLROnPlateau(monitor='val_acc',
                                             patience=5,
                                             verbose=1,
                                             factor=0.2,
                                             min_lr=0.0001)
callbacks = [earlystop,checkpoint,learning_rate_reduction]
model.compile(loss='categorical_crossentropy',
              optimizer=Adam(lr=0.001),
              metrics=['accuracy']
              )
nb_train_samples = 24176
nb_validation_samples = 3006
epochs = 25
history = model.fit_generator(
    train_generator,
    steps_per_epoch=nb_train_samples//batch_size,
    epochs=epochs,
    callbacks=callbacks,
    validation_data=validation_generator,
    validation_steps=nb_validation_samples//batch_size)

```

Audiototext.py

```
# Python program to translate
# speech to text and text to speech
from transformers import pipeline
classifier = pipeline('sentiment-analysis')
import speech_recognition as sr
import pyttsx3

# Initialize the recognizer
r = sr.Recognizer()

# Function to convert text to
# speech
def SpeakText(command):
    # Initialize the engine
    engine = pyttsx3.init()
    engine.say(command)
    engine.runAndWait()

file_clear = open("audiototext.txt", "w")
file_clear.close()

# Loop infinitely for user to speak till any keyboard interruption
while(1):
    # Exception handling to handle
    # exceptions at the runtime
    try:
        # use the microphone as source for input.
        with sr.Microphone() as source2:
            # wait for a second to let the recognizer
            # adjust the energy threshold based on
            # the surrounding noise level
            r.adjust_for_ambient_noise(source2, duration=0.2)
            #listens for the user's input
            audio2 = r.listen(source2)

            print("Started")

            # Using google to recognize audio
            MyText = r.recognize_google(audio2)
            MyText = MyText.lower()
            file1 = open("audiototext.txt", "a")
```

```

        file1.write("\n")
        file1.write(MyText)
        print("Output Text ",MyText)
        classifier(MyText)
        print("Sent text to Classifier")
        SpeakText(MyText)
        file1.close()
except sr.RequestError as e:
    print("Could not request results; {0}".format(e))
except sr.UnknownValueError:
    print("unknown error occurred")

```

audiosentiment.py

```

from transformers import AutoTokenizer, TFAutoModelForSequenceClassification
model_name = "nlpTown/bert-base-multilingual-uncased-sentiment"
model = TFAutoModelForSequenceClassification.from_pretrained(model_name, from_pt=True)
tokenizer = AutoTokenizer.from_pretrained(model_name)
classifier = pipeline('sentiment-analysis', model=model, tokenizer=tokenizer)
file1 = open("audiototext.txt", "r")
MyText = file1.read()
print(MyText)
result=classifier(MyText)
print(result)
label_audio = result[0]['label']
label_audio_score = result[0]['score']
print(label_audio)
print(label_audio_score)
print("Audio Sentiment: \t",end="")
if(label_audio == '1 stars'):
    label_audio_senti = "Negative"
    print("Negative")
elif(label_audio == '2 stars'):
    print("Negative")
    label_audio_senti = "Negative"
elif(label_audio == '3 stars'):

```



```

    print("Neutral")
    label_audio_senti = "Neutral"
elif(label_audio == '4 stars'):
    print("Positive")
    label_audio_senti = "Positive"
elif(label_audio == '5 stars'):
    print("Extremely Positive")
    label_audio_senti = "Extremely Positive"

```

finalsentiment.py

```

def senti_final(val):
    if(val>=0 and val<=1):
        return "Extremely Negative"
    elif(val>1 and val<=2):
        return "Negative"
    elif(val>2 and val<=3):
        return "Neutral"
    elif(val>3 and val<=4):
        return "Positive"
    else :
        return "Extremely Positive"

import math
print(label_audio_senti)
print(label_video)
sentiment_mapping = {
    'Extremely Negative': 0,
    'Negative': 1,
    'Neutral': 2,
    'Positive': 3,
    'Extremely Positive': 4 }
audio_value = sentiment_mapping[label_audio_senti]
video_value = sentiment_mapping[label_video]
audio_weight = 0.6
video_weight = 0.4

```

```
audio_value *= audio_weight
video_value *= video_weight
final_temp=max(video_value,audio_value)
print(final_temp)
final = math.ceil(final_temp)
print("Final sentiment is :",final,"-",senti_final(final))
```