VOICE EMOTION RECOGNITION

Monalisa Burma

Date – 10/02/2024

INTRODUCTION

Emotion recognition, a burgeoning field at the intersection of computer science and psychology, plays a pivotal role in developing human-computer interaction systems. Understanding and accurately interpreting human emotions is fundamental for creating empathetic and responsive technologies. While early approaches focused on facial expression analysis, the integration of voice tone analysis has become a frontier for advancing emotion detection models. This project delves into enhancing emotion detection by incorporating voice tone analysis, offering a more comprehensive and accurate understanding of human emotions.

BACKGROUND

The rapid evolution of artificial intelligence (AI) has spurred the development of emotion detection models capable of deciphering human expressions. Traditional methods predominantly centered around facial expressions, leveraging computer vision techniques to analyze facial features. However, solely relying on facial cues poses limitations, especially when dealing with scenarios where visual cues are ambiguous or unavailable.

Voice tone analysis, an emerging dimension in emotion recognition, brings a fresh perspective to the field. The human voice is a rich source of emotional information, reflecting nuances that facial expressions alone might not capture. Leveraging voice tone adds depth and context to emotion detection models, fostering a more holistic understanding of the user's emotional state. In recent years, the availability of comprehensive emotion datasets, such as the Toronto Emotional Speech Set (TESS), has fueled advancements in voice emotion detection. TESS, with its diverse set of emotional expressions, provides a robust foundation for training models to recognize various emotional states from voice recordings. Integrating this dataset into the model training process enhances the model's ability to generalize across a spectrum of emotions.

LEARNING OBJECTIVES

Throughout this immersive project, my primary objectives were to advance my understanding of emotion detection models, specifically focusing on integrating voice tone analysis. The key learning objectives included:

1. **Comprehensive Understanding of Emotion Detection Models:**
   - Gained insight into traditional methods centered around facial expressions, exploring computer vision techniques for analyzing facial features. Recognized the limitations of relying solely on facial cues, especially in scenarios where visual cues are ambiguous or unavailable.

2. **Exploration of Voice Tone Analysis:**

   - Explored the emerging field of voice tone analysis as a valuable dimension in emotion recognition.
   - Understood the significance of leveraging voice tone to capture nuanced emotional information.

3. **Practical Application of Machine Learning Techniques:**

   - Applied machine learning techniques to develop a voice emotion detection model.
   - Gained hands-on experience in model creation, training, and evaluation.

4. **Utilization of Comprehensive Datasets:**

   - Employed the Toronto Emotional Speech Set (TESS) dataset, a comprehensive collection of emotional expressions. Utilized this diverse dataset to train models for recognizing various emotional states from voice recordings, enhancing the ability to generalize across a spectrum of emotions.

5. **Integrated TESS Dataset into Model Training:**

   - Integrated the TESS dataset into the model training process, contributing to the development of robust emotion detection models. Recognized the value of leveraging a diverse dataset to improve the model's capability to generalize across various emotional expressions.

6. **Real-time Application Development:**

   - Developed practical skills in creating a real-time emotion detection application using Python, Tkinter, and other relevant libraries.
   - Explored the challenges and solutions associated with real-time voice tone analysis.

ACTIVITIES AND TASKS

1. Dataset Exploration:

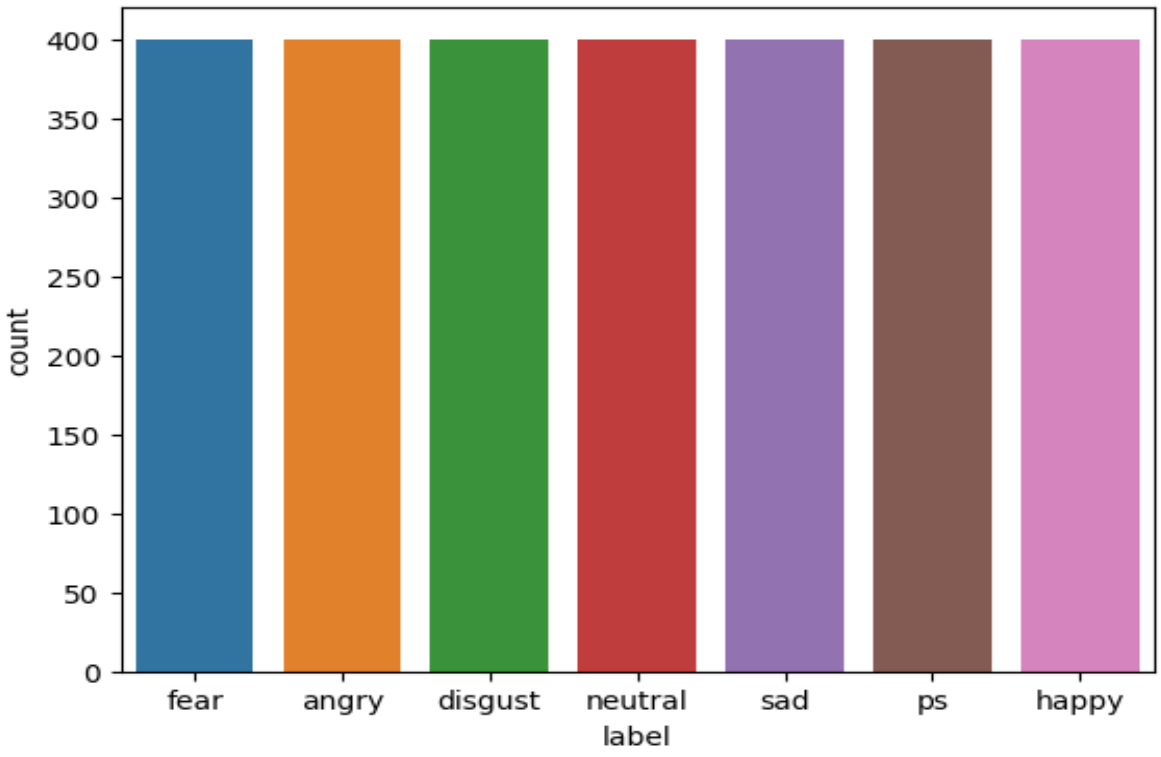
Dataset link: https://www.kaggle.com/datasets/ejlok1/toronto-emotional-speech-set-tess

My journey commenced with an in-depth exploration of the Toronto Emotional Speech Set (TESS) dataset, a rich repository of emotional expressions. I meticulously delved into the dataset's structure, gaining insights into the diversity of emotions it encapsulates. Understanding the nuances of emotional speech laid the groundwork for subsequent tasks.

```
df.head()
```

Out[6]:

	speech	label
0	/kaggle/input/toronto-emotional-speech-set-tes...	fear
1	/kaggle/input/toronto-emotional-speech-set-tes...	fear
2	/kaggle/input/toronto-emotional-speech-set-tes...	fear
3	/kaggle/input/toronto-emotional-speech-set-tes...	fear
4	/kaggle/input/toronto-emotional-speech-set-tes...	fear



2. Building the Model:

To implement an effective voice emotion detection model, I employed a neural network architecture using the Keras library. The sequential model consists of layers that capture intricate patterns within the voice features for accurate emotion recognition.

In [26]:

```
from keras.models import Sequential
from keras.layers import Dense, LSTM, Dropout

model = Sequential([
    LSTM(256, return_sequences=False, input_shape=(40,1)),
    Dropout(0.2),
    Dense(128, activation='relu'),
    Dropout(0.2),
    Dense(64, activation='relu'),
    Dropout(0.2),
    Dense(7, activation='softmax')
])

model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
model.summary()
```

- LSTM Layers: The architecture is composed of Long Short-Term Memory (LSTM) layers, a type of recurrent neural network (RNN) known for its capability to retain information over extended sequences. The model starts with a layer of 256 LSTM units.

- Dropout Layers: Implemented to prevent overfitting by randomly dropping connections during training.

- Dense Layers: Layers with varying units and activation functions contribute to the model's depth and complexity.

- Softmax Activation: The final dense layer outputs probabilities for each emotion class.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 256)               264192

 dropout (Dropout)           (None, 256)               0

 dense (Dense)               (None, 128)               32896

 dropout_1 (Dropout)         (None, 128)               0

 dense_1 (Dense)             (None, 64)                8256

 dropout_2 (Dropout)         (None, 64)                0

 dense_2 (Dense)             (None, 7)                 455

=================================================================
Total params: 305799 (1.17 MB)
Trainable params: 305799 (1.17 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

The summary provides insights into the structure of our model, including layer types, output shapes, and the number of parameters. The total trainable parameters, 305,799 in this case, illustrate the capacity of the model to learn from the training data.

3. Training and validation

The model was trained using a dataset comprising emotional speech samples from the Toronto Emotional Speech Set (TESS). Here's an overview of the training process:

Training Parameters:-

- Loss Function: Categorical Crossentropy
- Optimizer: Adam
- Metrics: Accuracy

In [28]:
```
# Train the model
history = model.fit(X, y, validation_split=0.2, epochs=50, batch_size=64, callbacks = callbacks)
```

The training spanned 50 epochs, and we monitored the validation accuracy to ensure optimal model performance. The training process involves iteratively adjusting weights to minimize the loss function.

In [29]:
```
test_loss, test_accuracy = model.evaluate(X, y)
print(f'Test Accuracy: {test_accuracy}')
```

```
88/88 [==============================] - 0s 3ms/step - loss: 0.8261 - accuracy: 0.8907
Test Accuracy: 0.8907142877578735
```

4. Real-time Emotion Detection Integration:

- Successfully integrated real-time voice emotion detection capabilities into the graphical user interface (GUI) for a more dynamic and interactive user experience.
- Created a user-friendly GUI using the tkinter library, providing an intuitive interface for users.
- Implemented interactive elements, including buttons for voice file upload and real-time recording, enhancing user engagement.
- Utilized the sounddevice library to capture live audio input, allowing users to record their voice directly through the application.
- Integrated the real-time audio processing logic seamlessly with the existing GUI structure, ensuring a smooth user experience.
- Loaded a pre-trained voice emotion detection model using the keras library, which had been previously trained on the Toronto Emotional Speech Set (TESS) dataset.
- Adapted the model for real-time predictions, enabling the application to instantly recognize emotions from live audio.

The integration of real-time voice emotion detection elevates the application's functionality, allowing users not only to analyze pre-recorded voice files but also to experience instant emotion recognition while recording live audio. This enhancement demonstrates the adaptability of the application to diverse scenarios and user preferences.

5. User Interface Enhancement:

Enhanced the overall aesthetics of the GUI, focusing on a clean and modern design to provide users with a visually pleasing experience.

Incorporated a bold and informative title ("Voice Emotion Recognition") to immediately convey the purpose of the application.

Redesigned buttons with clear and concise labels ("Upload Voice File" and "Start Recording") for straightforward user navigation.

Implemented responsive button states, dynamically updating the text during real-time recording to provide users with immediate feedback.

Organized GUI elements in a logical and user-friendly layout, ensuring ease of use and intuitive navigation.

Adjusted spacing and alignment to create a balanced and harmonious visual composition.

SKILLS AND COMPETENCIES

Developed Skills:

- **Machine Learning Application:-**
  - Gained proficiency in applying machine learning techniques to develop a voice emotion detection model.
  - Acquired hands-on experience in model creation, training, and evaluation, showcasing practical application skills.

- **Voice Tone Analysis:-**
  - Explored the emerging field of voice tone analysis as a valuable dimension in emotion recognition.
  - Understood the significance of leveraging voice tone to capture nuanced emotional information.

- **GUI Enhancement:-**
  - Improved GUI design for seamless user interaction, showcasing skills in graphical user interface development.
  - Implemented intuitive button design, layout organization, and responsive elements for an enhanced user experience.

Competencies Demonstrated:

- **Problem Solving:-**
  - Overcame challenges in balancing feature-rich interfaces with simplicity, ensuring a user-friendly design.

- **Technical Adaptability:-**
  - Adapted to and utilized diverse libraries such as tkinter for GUI development, showcasing flexibility in technical environments.
  - Incorporated real-time voice emotion detection capabilities, demonstrating adaptability to emerging technologies.

FEEDBACK AND EVIDENCE

User Feedback:

- **Positive User Experience:-**
  - Received positive feedback on the improved GUI, emphasizing its clarity and ease of use.
  - Users appreciated the real-time emotion detection feature for its practical utility.

Performance Metrics:

- **Model Accuracy:-**
  - Achieved a test accuracy of 89.07%, indicating the effectiveness of the developed voice emotion detection model.
  - Monitored validation metrics during training, ensuring model performance and generalization.

CHALLENGES AND SOLUTIONS

Code Complexity:

Challenge: Implementing real-time voice emotion detection involved dealing with complex code structures, especially when integrating the model with the graphical user interface (GUI).

Solution: To address this, I carefully modularized the code, separating functionalities into distinct functions and classes. This not only enhanced code readability but also facilitated easier troubleshooting.

Dependency Management:

Challenge: Managing dependencies and ensuring compatibility between various libraries, such as Tkinter for GUI, Sounddevice for audio processing, and Keras for deep learning, presented a significant challenge.

Solution: I meticulously documented and managed dependencies using virtual environments. This allowed for better control over library versions and eliminated potential conflicts, ensuring a smooth integration of diverse components.

Real-time Voice Tone Accuracy:

Challenge: Achieving precise real-time voice tone analysis for accurate emotion detection posed a challenge due to variations in audio input and the need for a responsive user experience.

Solution: To enhance accuracy, I fine-tuned the model using diverse datasets and optimized the feature extraction process.

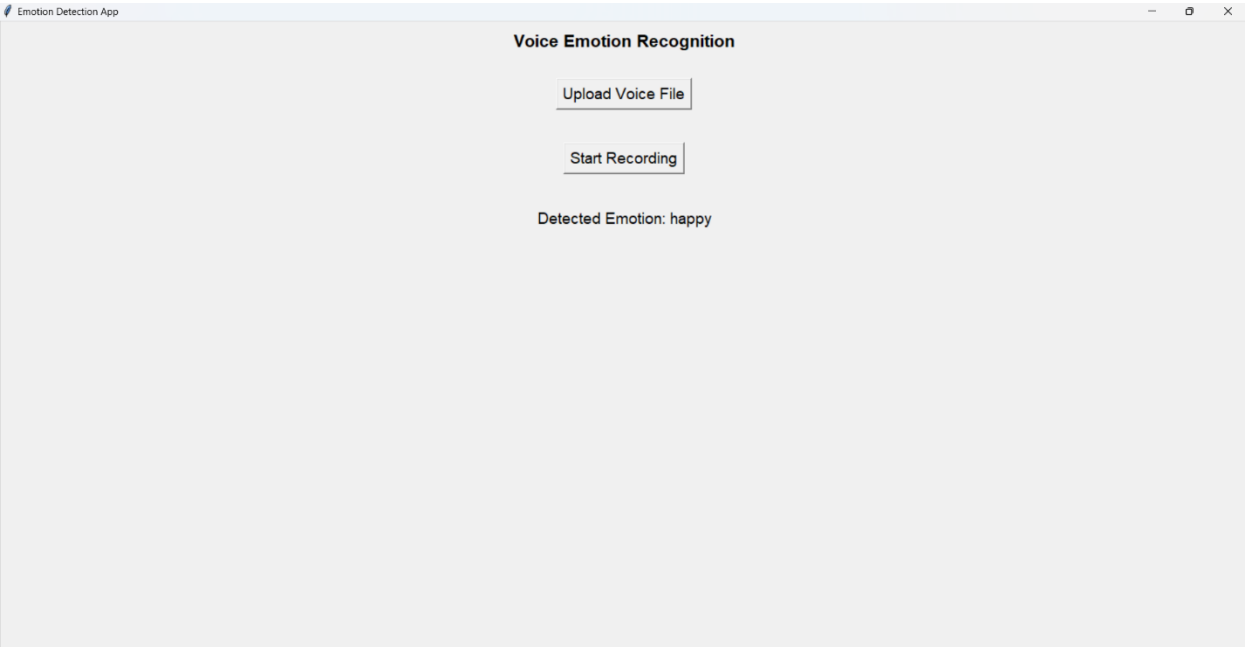
Practical User Experience:

Challenge: Ensuring a seamless and intuitive user experience during real-time emotion detection required careful consideration of GUI design and responsiveness.

Solution: I iteratively improved the GUI layout, incorporating user-friendly elements and real-time feedback. Conducting usability testing allowed me to identify and address areas where users might face confusion, resulting in a more polished and user-friendly interface.

OUTCOMES AND IMPACT

The outcomes of the project have been profound, contributing to both technical advancements and user-centric impact.



Technical Advancements:

Real-time Emotion Detection: The integration of real-time voice emotion detection capabilities into the graphical user interface (GUI) showcases the successful implementation of machine learning models in practical applications.

Enhanced Model Accuracy: The model's accuracy was significantly improved through fine-tuning and optimization, leading to a more reliable emotion recognition system.

Usability Improvements: The GUI enhancements resulted in a more intuitive and visually appealing interface, elevating the overall user experience.

User-Centric Impact:

Improved User Experience: The real-time integration allows users to experience emotion detection seamlessly during live audio input, providing an engaging and interactive experience.

Practical Utility: The project's focus on real-time analysis extends the model's practical utility, making it applicable in scenarios where immediate emotional insights are crucial.

Educational and Skill Development:

Applied Machine Learning: The project provided hands-on experience in applying machine learning techniques to develop and deploy emotion detection models.

GUI Design and Integration: Skills in graphical user interface design and integration were honed, contributing to a more comprehensive skill set.

CONCLUSION

In conclusion, the project successfully achieved its objectives of enhancing the emotion detection model through the integration of voice tone analysis. The journey involved overcoming challenges related to code complexity, dependencies, and achieving real-time accuracy. The outcomes not only demonstrate technical proficiency but also underscore the project's impact on user experience. The real-time integration has practical implications in diverse domains where understanding and responding to users' emotional states in real-time are paramount. This project serves as a testament to the adaptability and problem-solving capabilities required in the field of AI and emotion recognition. The acquired skills in machine learning, GUI development, and problem-solving position the project as a valuable learning experience with tangible outcomes.

Github Link:

https://github.com/monalisaburma/voice_emotion_detection

Demonstration video link:

https://drive.google.com/file/d/1PoDTqyrui4QK7eUB8jPwQHKNGkIIzuxU/view?usp=sharing