UE20CS390B – Capstone Project Phase – 2

SEMESTER - VII

END SEMESTER ASSESSMENT

Project Title : FeelSpeak: Generating Emotional Speech with Deep Learning
Project ID : PW23_VRB_07
Project Guide : Prof. V R Badri Prasad
Project Team : 235_320_345_362

Abstract

- Goal: Develop a system for generating emotional speech from input text.
- Approach: Identify emotions in text, synthesize speech with appropriate prosodic features.
- Tasks: Natural Language Processing (NLP), text emotion detection, speech synthesis, emotion recognition.
- Integration: Fusion of NLP, speech synthesis, and emotion recognition for holistic interaction.

Team Roles and Responsibilities

Name	Responsibility
Rahul Roshan G	Detecting emotions from text by training the preprocessed dataset on various machine learning models.
Rohit Roshan	Collecting datasets for both project components, performing preprocessing for the first part and validating using emoroberta , configuring hyperparameters for the second part
S M Sutharsan Raj	Building tacotron model to identify prosodic features of speech and add that to input text based on annotated emotion of the text and then generate the mel-spectrogram out of it to get an emotional speech. Also validate the emotional speech generated.
Sohan M H	Appending the detected emotions alongside their corresponding text from part 1 and developing a user-friendly interface for part 1 using Streamlit.

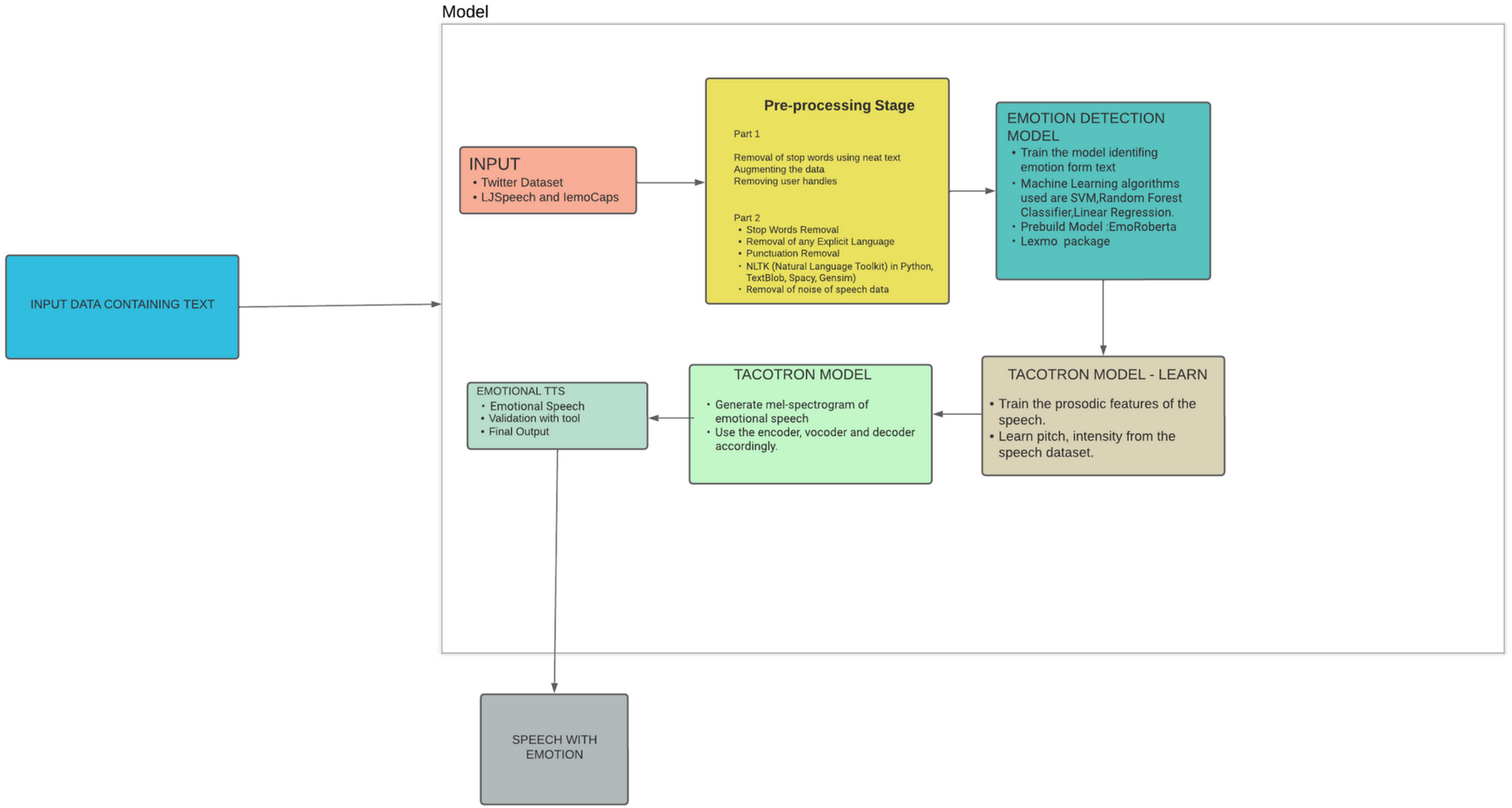
Summary of Requirements and Design

REQUIREMENTS

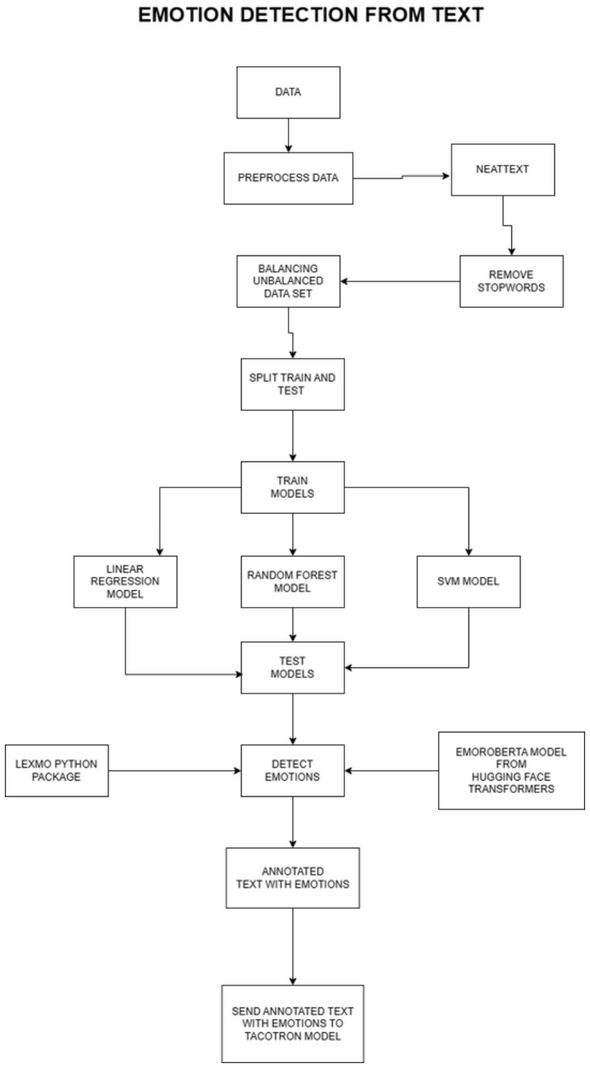
- Develop a system for converting plain text into emotionally expressive speech utilizing the Tacotron model and Text-to-Speech (TTS) methods, involving two main phases: Training and Testing.
- Utilize a labeled text dataset with emotions to train the Tacotron model for melspectrogram generation from input text.
- Implement attention mechanisms to capture emotional nuances during text to mel spectrogram conversion.
- Generate audio files containing speech with the desired emotion.
- Develop an intuitive user interface for inputting text and accessing synthesized emotional speech.
- Ensure compatibility with various devices and platforms.
- Achieve high accuracy in emotion detection.
- Ensure precise adjustment of speech features during synthesis.
- Define the minimum requirements for labeled text and speech datasets for effective training.
- Establish evaluation metrics for assessing the accuracy and generalization capabilities of the Tacotron, emotion detection, and regression models.
- Develop a comprehensive testing plan to validate the accuracy and effectiveness of the complete emotional text to speech synthesis system.
- Emotion detection from text using SVM, Linear Regression, Random Forest, EmoRoberta model from hugging face transformer and LeXmo python package.

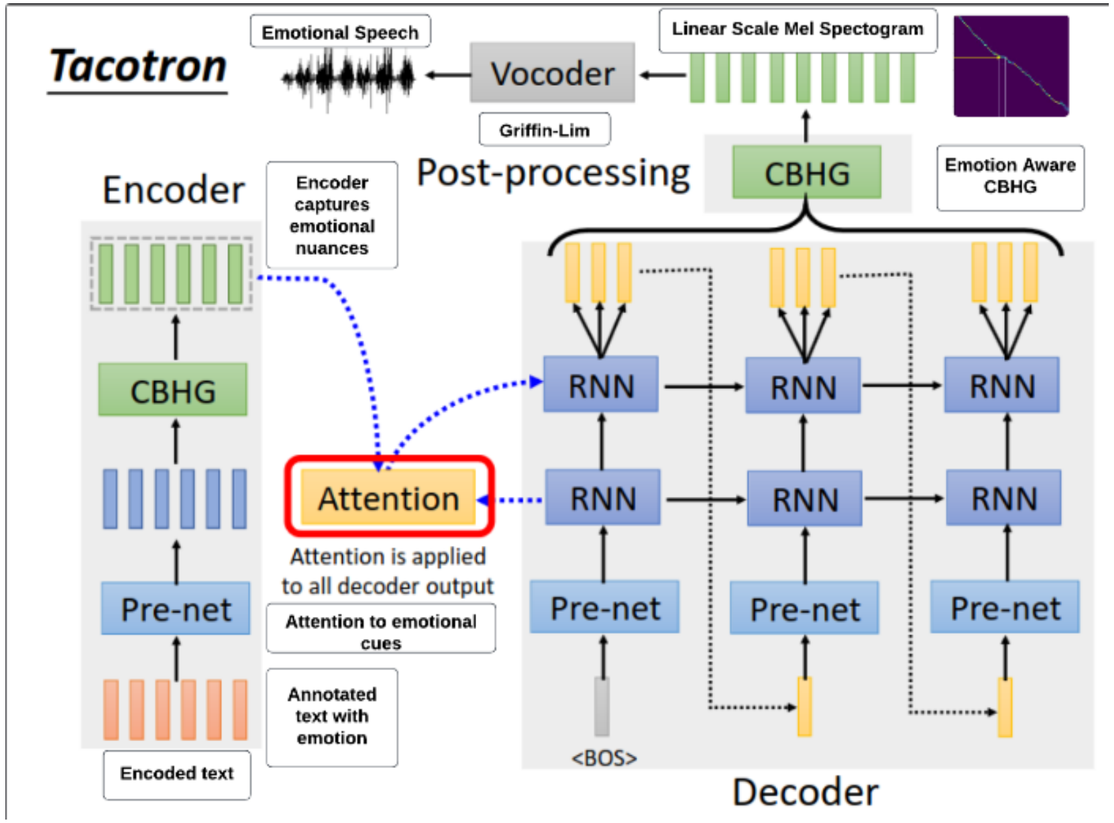
Summary of Requirements and Design

DESIGN DETAILS



Design Description





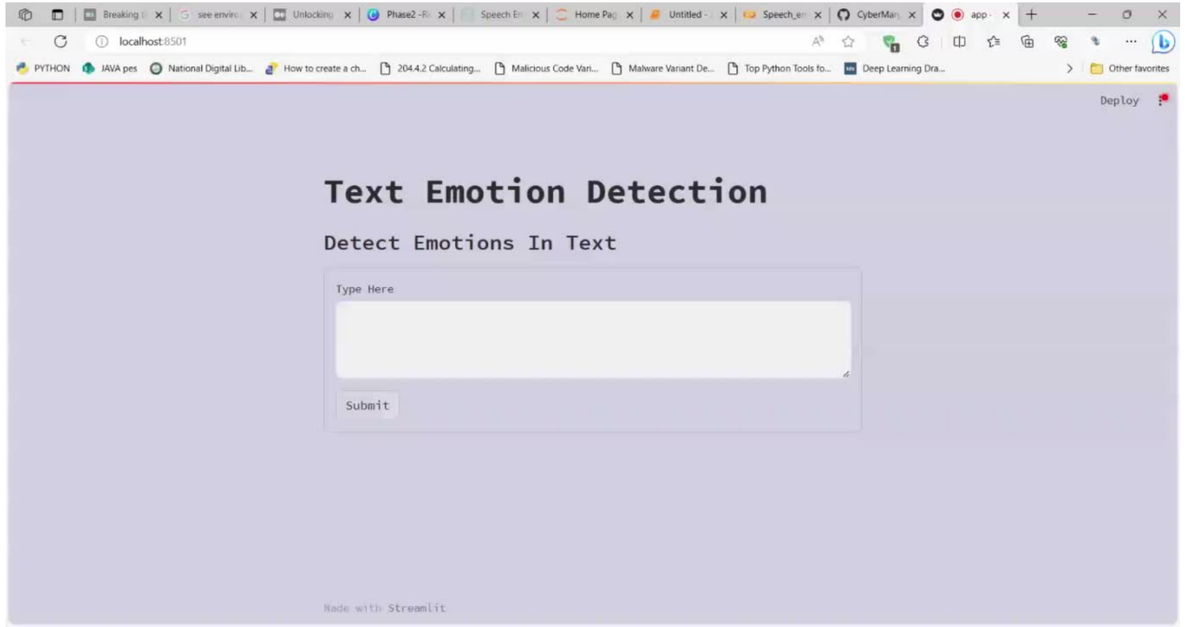
Modules and Implementation Details

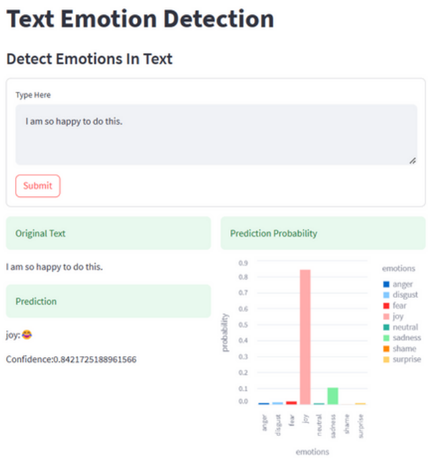
- Enlist all the modules/ features of the application.
- Module-wise implementation details that include
- Module name, Technology used, code explanation.
- Interpretation with Algorithms & Pseudocode used. (applicable for Research projects)

Modules and Implementation Details

Emotion detection from text.

- Detecting emotion from text using models like SVM, Random forest, linear regression and EmoRoBERTa Model from Huggingface. Transformers.
- Saving the model using pickle library
- Visualizing the model with graph and confidence score using streamlit.





Modules and Implementation Details

- LeXmo: The first Python package for classifying emotions in English texts
- LeXmo converts text into a pandas data frame, calculating emotion weights by dividing emotional association by word count.
- Find the demo here.

- It uses Emo-Roberta model to detect text from emotions from hugging face transformer see emotions below.
- It calls the model use this link and predicts the emotion.
- The models gives dictionary with key as label(emotion) and score.
- Best result f1-score: 49.03%

Dataset labelled 58000 Reddit comments with 28 emotions

- admiration, amusement, anger, annoyance, approval, caring, confusion, curiosity, desire, disappointment, disapproval, disgust, embarrassment, excitement, fear, gratitude, grief, joy, love, nervousness, optimism, pride, realization, relief, remorse, sadness, surprise + neutral

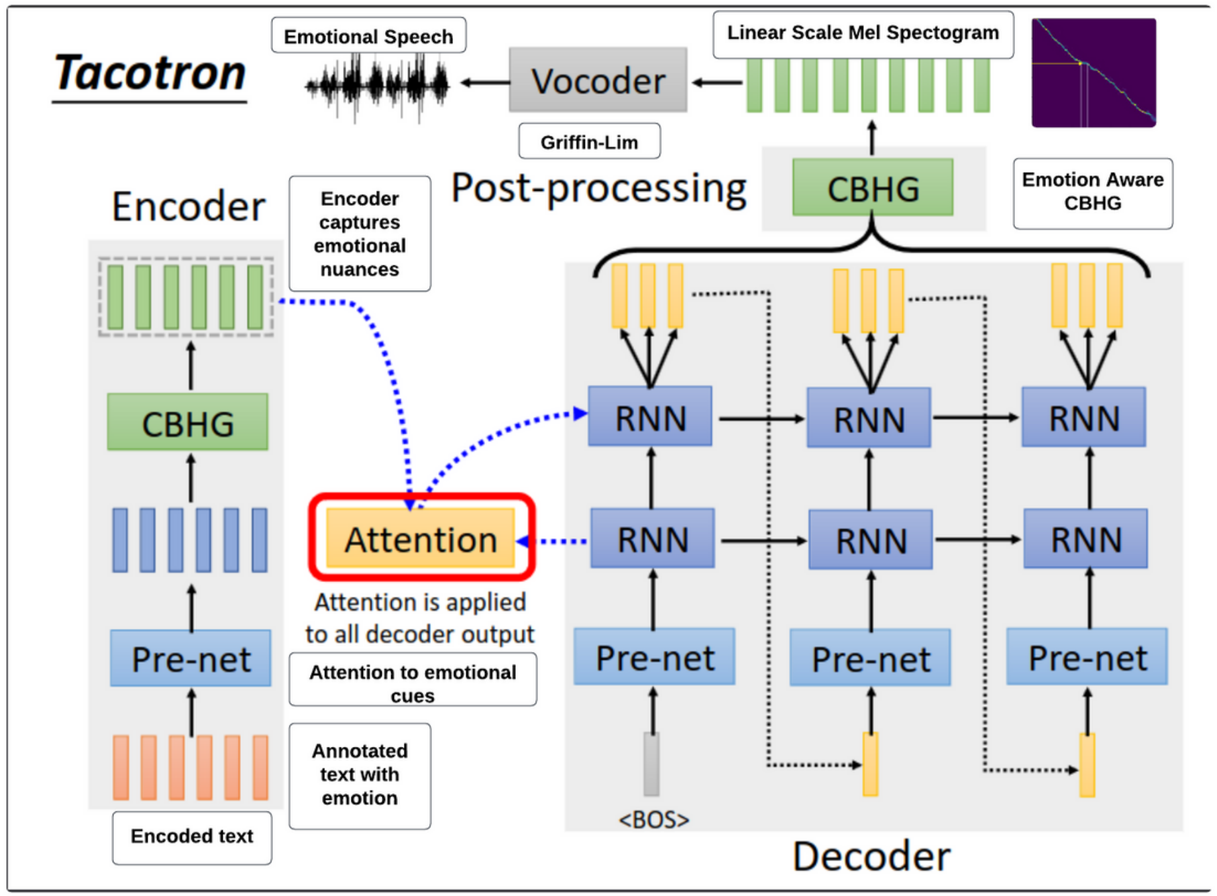
Modules and Implementation Details

1. Dataset:
Objective: Download the LJ Speech dataset for English speech samples.
Action Taken:
Downloaded the LJ Speech dataset.
Organized the dataset, including audio files and text transcripts, into a structured directory.

2. Preprocessing:
Objective: Prepare audio and text data for model training.
Actions Taken:
Extracted features, such as mel spectrograms, from the audio files.
Tokenized and preprocessed text data to ensure suitability for training.

Modules and Implementation Details

3. Model Architecture:
   - Objective: Design the Tacotron model for sequence-to-sequence mapping.
   - Actions Taken:
     - Developed the Tacotron model architecture, including an encoder, attention mechanism, and decoder.
     - Utilized recurrent neural networks (RNNs) or LSTM networks for effective sequence modeling.



Modules and Implementation Details

4. Training:
   - Objective: Train the Tacotron model using preprocessed data.
   - Actions Taken:
     - Defined loss functions, incorporating spectrogram loss and alignment loss.
     - Utilized the Adam optimizer, experimenting with learning rates and other hyperparameters.

5. Hyperparameter Tuning:
   - Objective: Optimize hyperparameters based on model performance.
   - Actions Taken:
     - Fine-tuned hyperparameters, including learning rates, batch sizes, and training epochs.

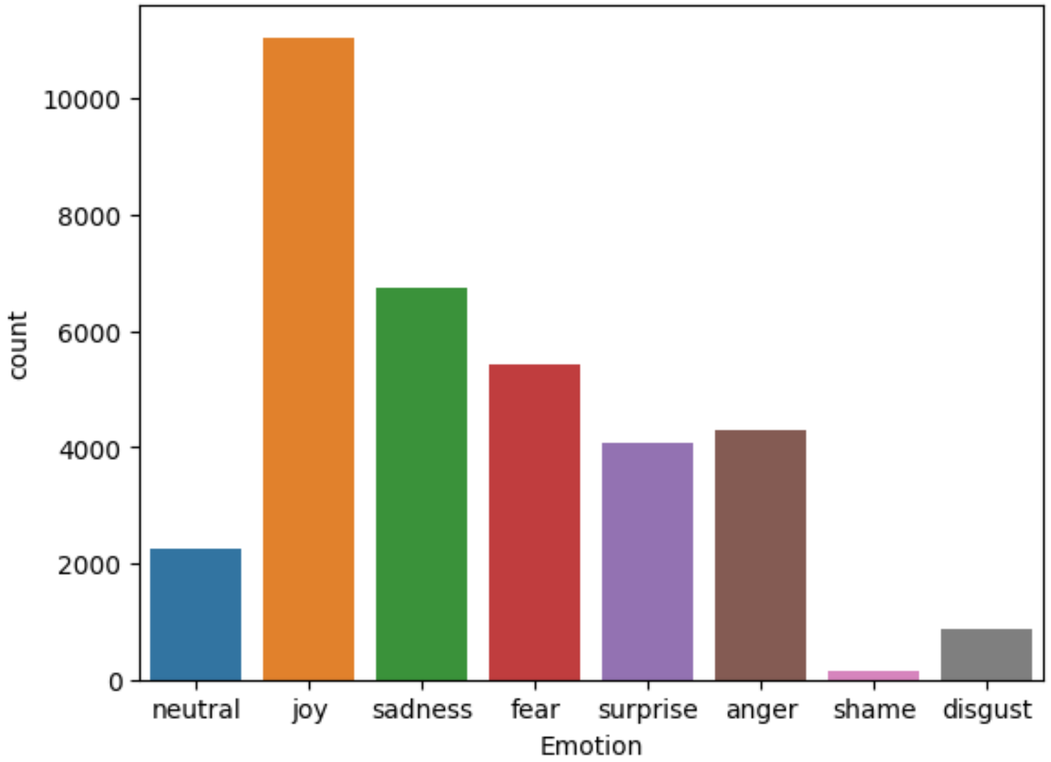
Modules and Implementation Details

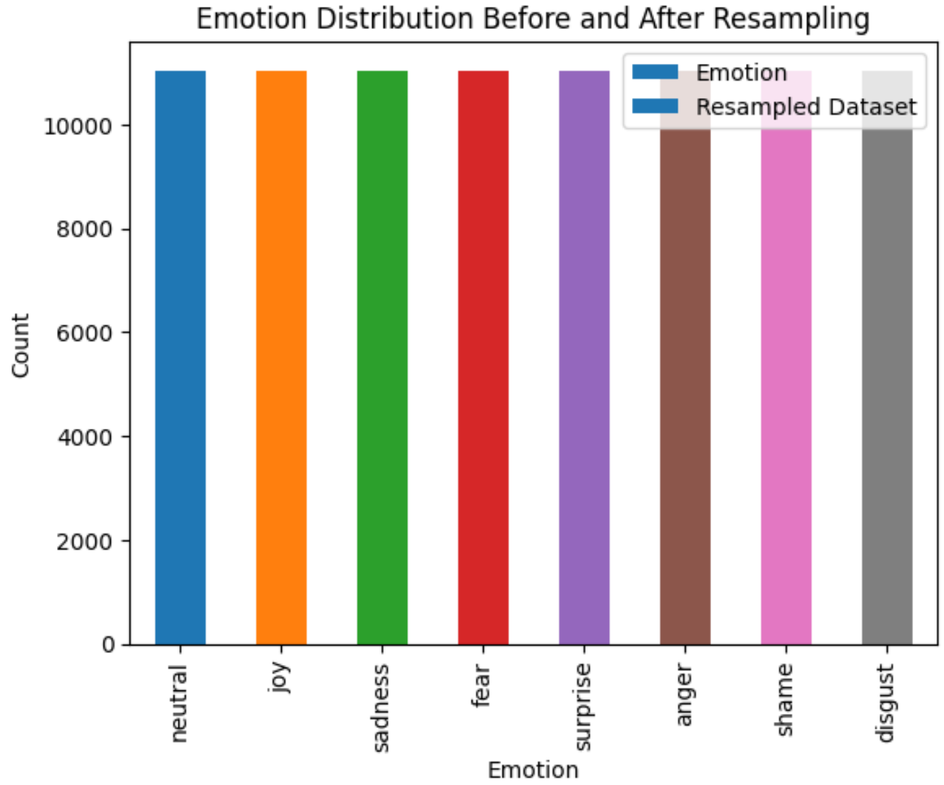
6. Evaluation:
   - Objective: Assess the performance of the trained model.
   - Actions Taken:
     - Evaluated the model on a validation set to ensure proper learning.
     - Leveraged metrics like Mean Opinion Score (MOS) in subjective listening tests for voice quality assessment.

7. Inference:
   - Objective: Implement an inference pipeline for synthesizing speech from text.
   - Actions Taken:
     - Developed an inference pipeline to synthesize speech using the trained Tacotron model.
     - Combined Tacotron output with a vocoder (e.g., Griffin-Lim) to generate the final waveform.

Project Demonstration

EMOTION DETECTION FROM TEXT - BALANCING DATASET

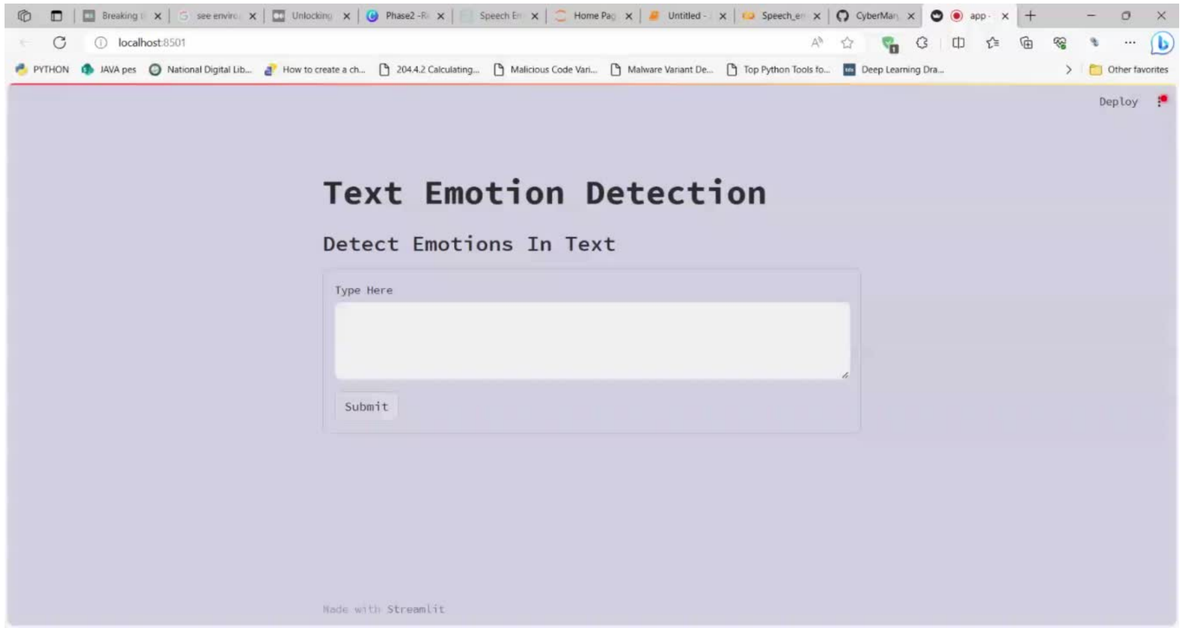




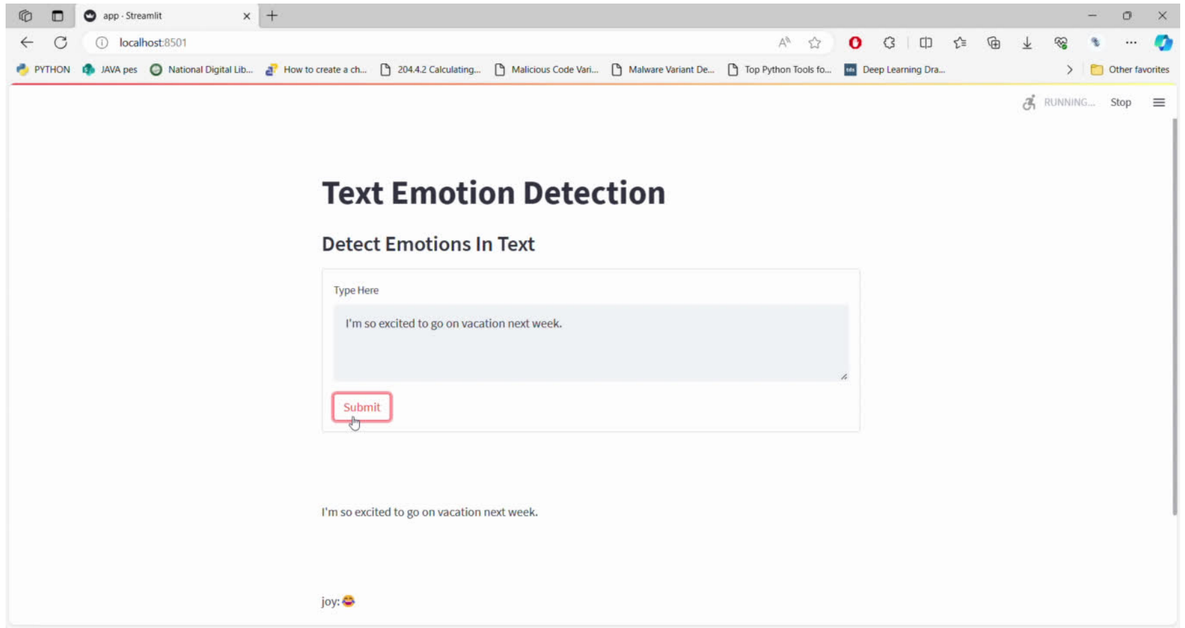
M H SOHAN_RAHUL ROSHAN G_ROHIT ROSHAN_S M SUTHARSAN RAJ

Project Demonstration

With unbalanced dataset

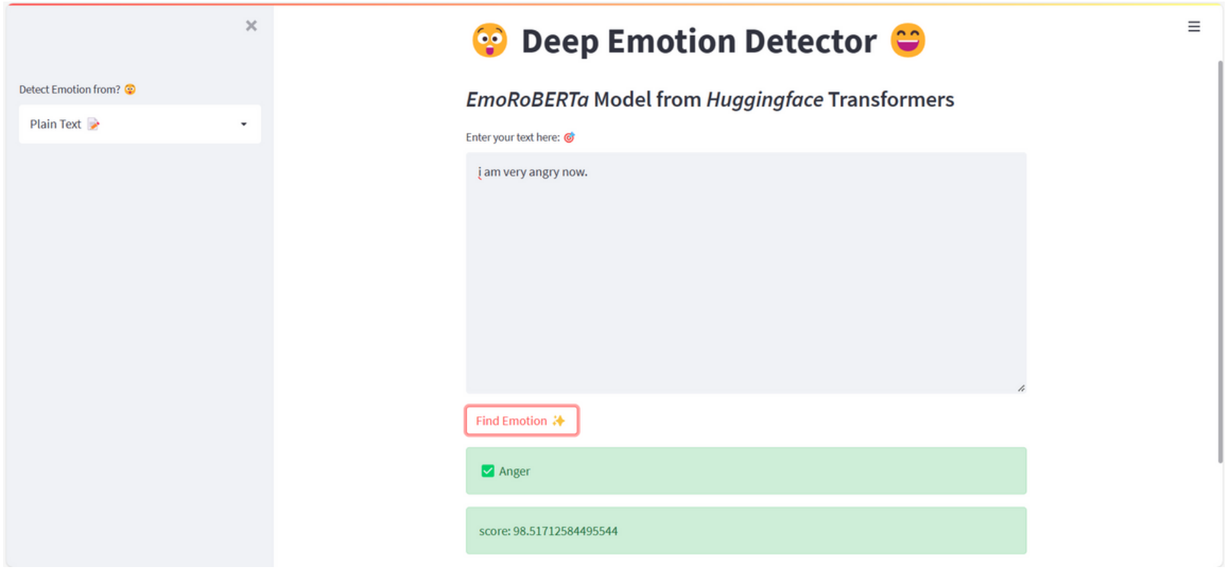


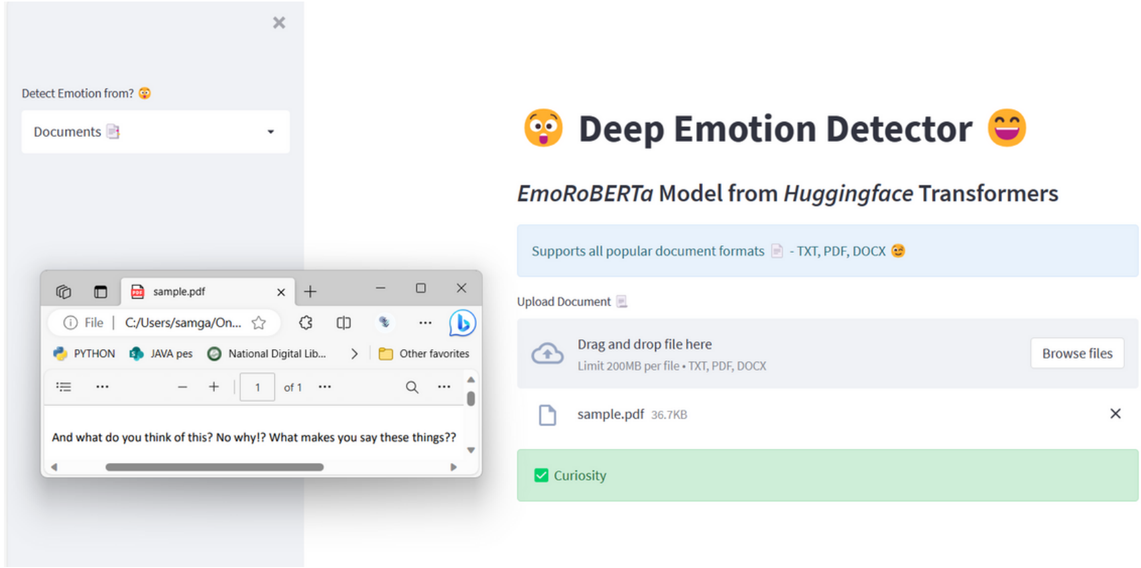
With balanced dataset



Project Demonstration

EmoRoberta model from Hugging face Transformers





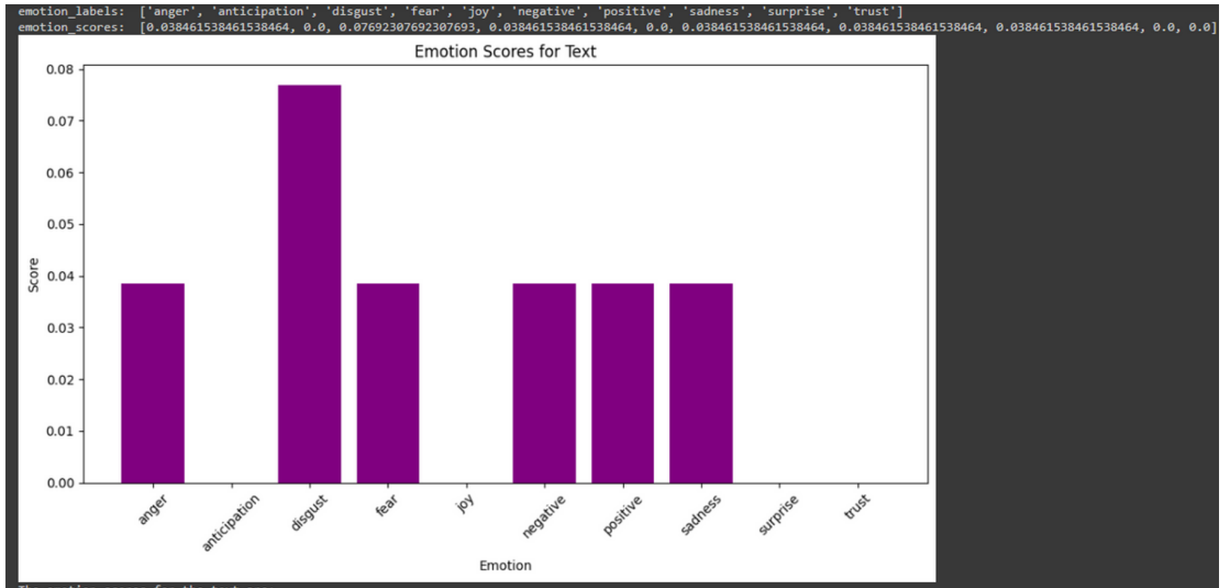
M H SOHAN_RAHUL ROSHAN G_ROHIT ROSHAN_S M SUTHARSAN RAJ

Project Demonstration

LexMo python package

```
[ ] t= """From the beginning, she had sat looking at him fixedly.
    As he now leaned back in his chair, and bent his deep-set eyes upon her in his turn,
    perhaps he might have seen one wavering moment in her,
    when she was impelled to throw herself upon his breast,
    and give him the pent-up confidences of her heart.
    But, to see it, he must have overleaped at a bound the artificial barriers he had for many years been erecting,
    between himself and all those subtle essences of humanity which will elude the utmost cunning of algebra
    until the last trumpet ever to be sounded shall blow even algebra to wreck.
    The barriers were too many and too high for such a leap. With his unbending,
    utilitarian, matter-of-fact face, he hardened her again;
    and the moment shot away into the plumbless depths of the past,
    to mingle with all the lost opportunities that are drowned there."""

[ ] emo=LeXmo.LeXmo(t)

[ ] print(emo)

    {'text': 'From the beginning, she had sat looking at him fixedly.\n  As he now leaned back in his chair, and bent h

[ ] emo.pop('text', None)

    'From the beginning, she had sat looking at him fixedly.\n  As he now leaned back in his chair, and bent his deep-s
    \n  when she was impelled to throw herself upon his breast,\n  and give him the pent-up confidences of her heart.\n
    any years been erecting, \n  between himself and all those subtle essences of humanity which will elude the utmost
    o wreck.\n  The barriers were too many and too high for such a leap. With his unbending,\n  utilitarian, matter-of-
    f the past,\n  to mingle with all the lost opportunities that are drowned there.'

[ ] print(emo)

    {'anger': 0.023255813953488372, 'anticipation': 0.0, 'disgust': 0.005813953488372093, 'fear': 0.023255813953488372,
```

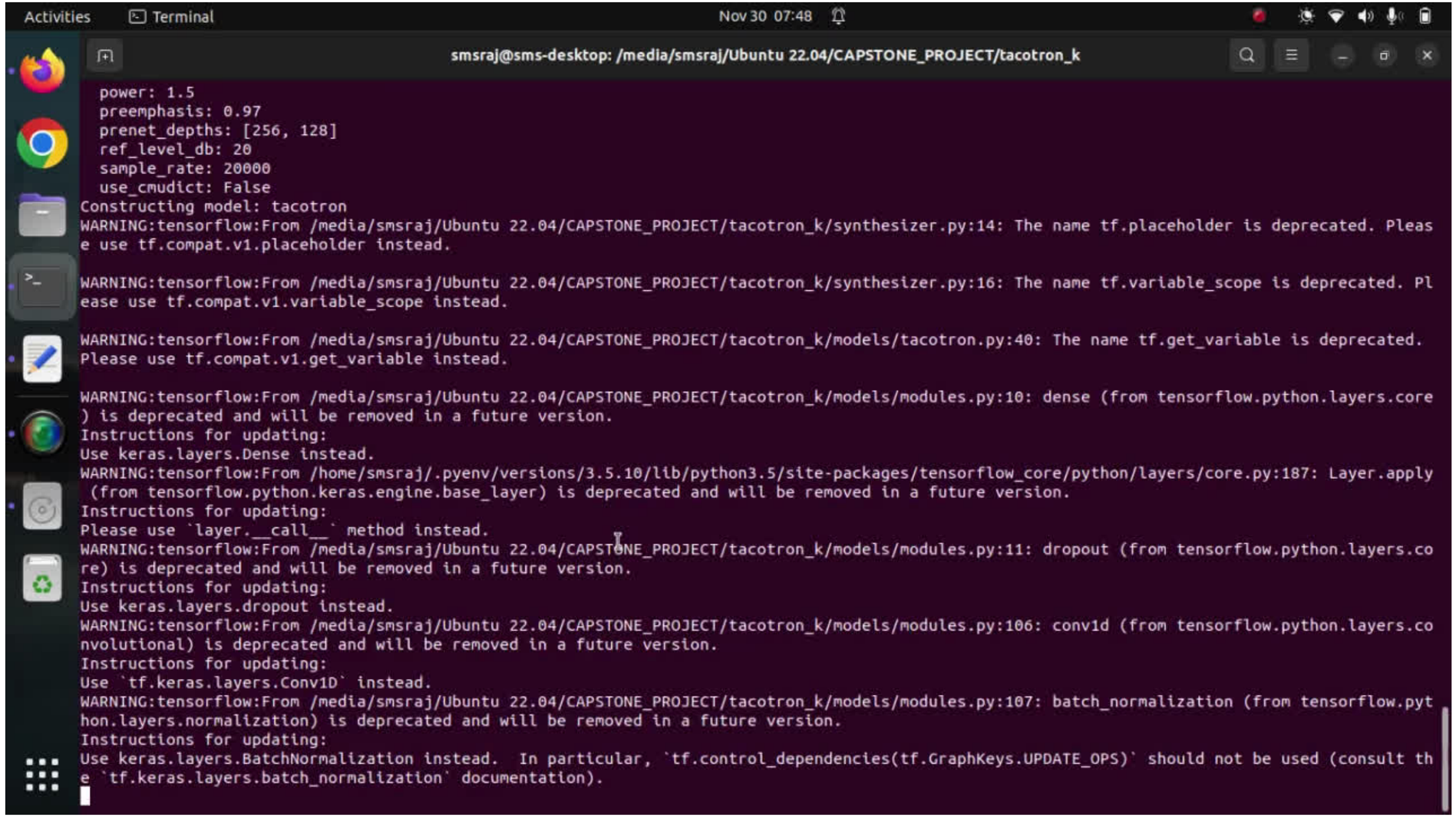
```
emotion_labels:  ['anger', 'anticipation', 'disgust', 'fear', 'joy', 'negative', 'positive', 'sadness', 'surprise', 'trust']
emotion_scores:  [0.038461538461538464, 0.0, 0.07692307692307693, 0.038461538461538464, 0.0, 0.038461538461538464, 0.038461538461538464, 0.038461538461538464, 0.0, 0.0]
```



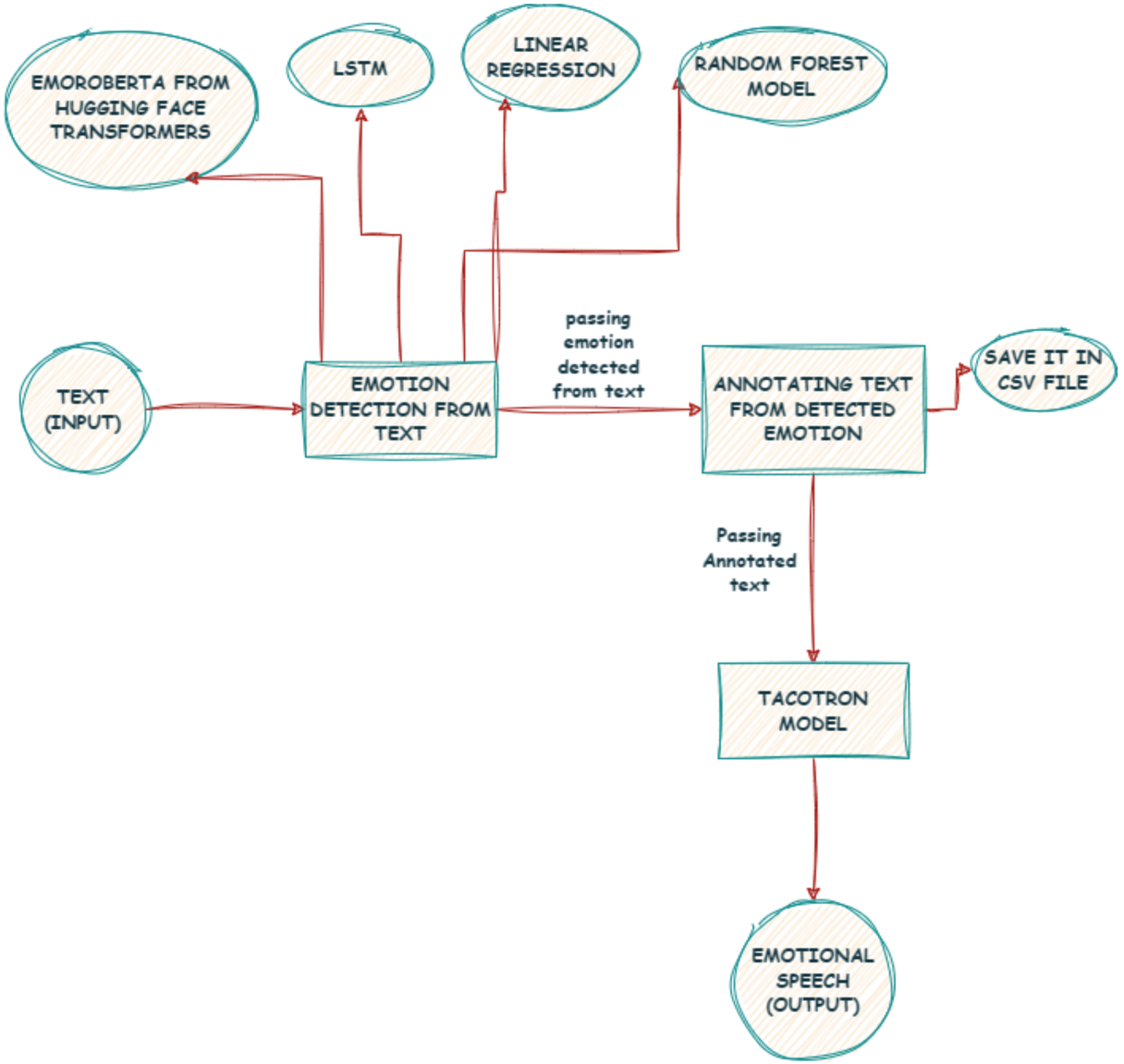

```
The emotion scores for the text are:
anger: 3.8461538461538463
anticipation: 0.0
disgust: 7.6923076923076925
fear: 3.8461538461538463
joy: 0.0
negative: 3.8461538461538463
positive: 3.8461538461538463
sadness: 3.8461538461538463
surprise: 0.0
trust: 0.0
```

Project Demonstration

Tacotron model video



Walkthrough



Test Plan and Strategy

- In the Speech Emotion Synthesis Project, testing is crucial at every step to guarantee quality and performance.
- Unit testing ensures individual components work correctly.
- Integration testing checks how these components collaborate.
- System testing examines the overall behavior, ensuring accurate emotion synthesis across different scenarios. Regular regression testing catches potential issues after each development cycle.
- Performance testing assesses system responsiveness under varying loads.
- User Acceptance Testing involves real users providing valuable feedback on synthesized speech, refining the system from an end-user perspective.
- Effective defect reporting and tracking mechanisms, like Jira, help resolve issues promptly. This testing strategy, coupled with clear exit criteria, ensures a high-quality project before deployment.

Test Plan and Strategy

Emotion detection from text using LR,RF, SVM

SENTENCE	EMOTION DETECTED	EXPECTED EMOTION
I am happy today.	JOY 76.5%	JOY
Alas, I lost all my project data due to a technical glitch with tacotron	SADNESS 77.9%	SADNESS
It's frustating how unreliable the results are . It's making me so angry!	ANGER 90.9%	ANGER

Test Plan and Strategy

Emotion detection from text using LR,RF, SVM

SENTENCE	EMOTION DETECTED	EXPECTED EMOTION
I'm scared of what the future holds.	FEAR 99.5%	FEAR
I didn't expect you to remember my birthday.	SURPRISE 73.64%	SURPRISE
I feel so embarrassed about what I did.	SHAME 99.5%	SHAME

Test Plan and Strategy

Emotion detection from text using EmoRoBERTa Model from Huggingface Transformers

SENTENCE	EMOTION DETECTED	EXPECTED EMOTION
Life’s good, you should get one.	NEUTRAL	NEUTRAL
The bear was ravenous, he was fierce and furious	ANGER 97.51%	ANGER
In sooth I know not why I am so melancholic.	SADNESS 73.61%	SADNESS

Test Plan and Strategy

Emotion detection from text using EmoRoBERTa Model from Huggingface Transformers

SENTENCE	EMOTION DETECTED	EXPECTED EMOTION
Waaaaw!, this car is amazing!	EXCITMENT 77.86%	HAPPY
I'm so ashamed of my behavior.	EMBARRASSMENT 98.62%	SHAME
I'm afraid of public speaking.	FEAR 99.03%	FEAR

Results and Discussion

Emotion detection from text

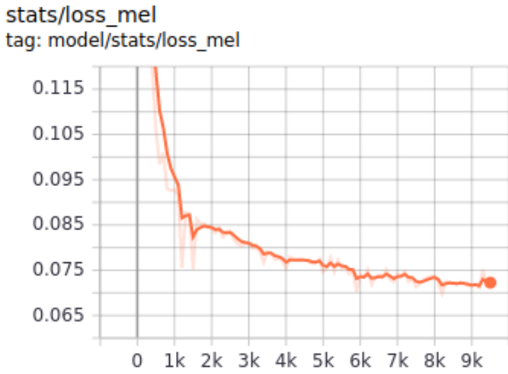
- After training and testing the model this are the accuracy score:
  - Linear Regression: 86.13%
  - Random Forest model: 89.38%
  - Support Vector Machine: 87.93%

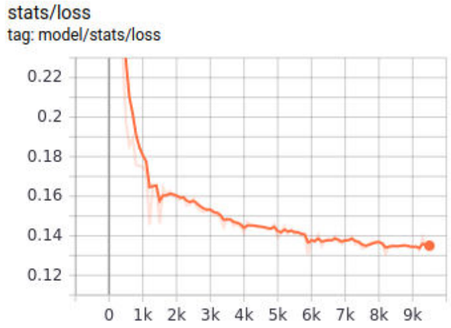
- EmoRoberta model of hugging face transformers is a pretrained model with the f1-score of 49.30%

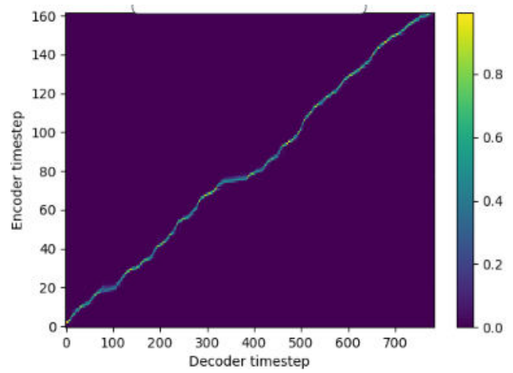
Results and Discussion

Emotional Speech Synthesis

Speech	Expected	Happy %	Sad %	Neutral %
“Ha Ha, this joke is so funny to laugh”	Happiness	73.2	0.5	2.31
“I am so excited for the vacation”	Happiness	81.4	10.4	1
“Alas, I lost all my project data due to a technical glitch with tacotron”	Sadness	18.1	81.0	0.4
“Oh, I feel grieved depressed about the mournful incident”	Sadness	0.26	63.1	2
“He is chasing the butterflies”	Neutral	20.2	2.3	75.4
“I am working on this project”	Neutral	10.4	15.6	78.7



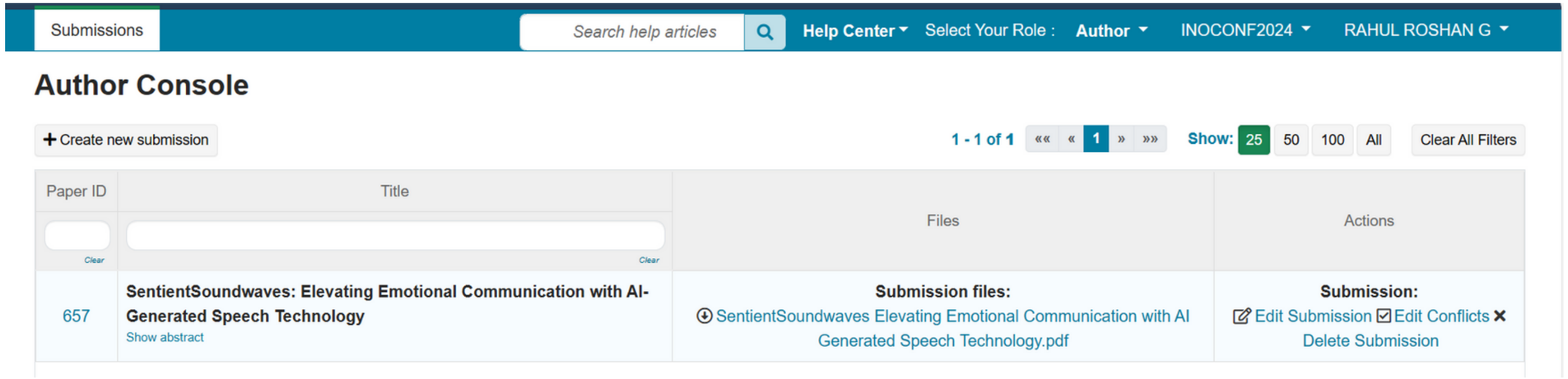


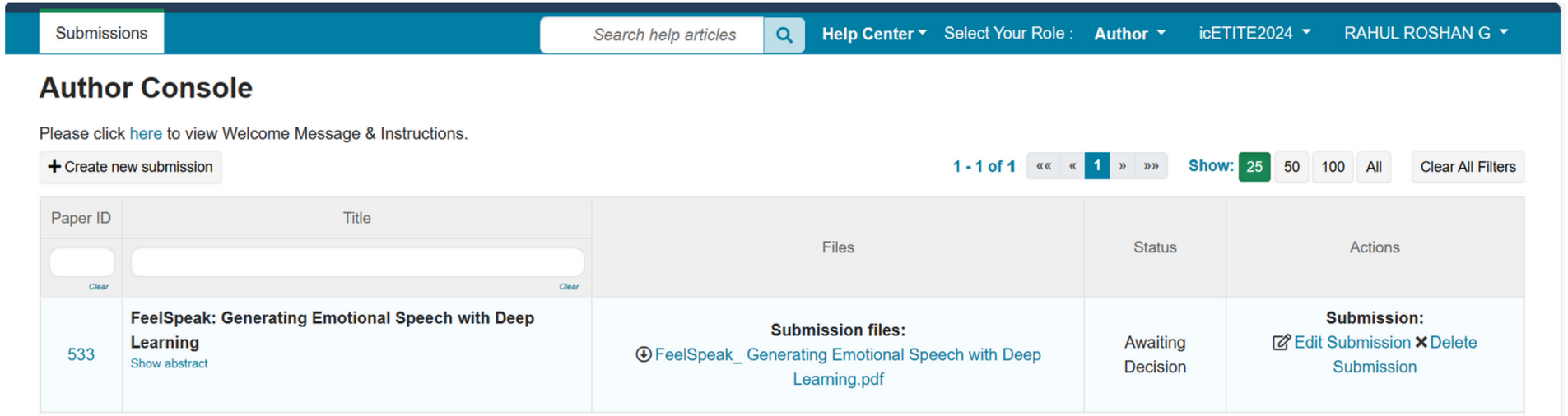



```
smsraj@sms-desktop:~/Desktop/OpenVokaturi-4-0/OpenVokaturi-4-0$ python3 examples/OpenVokaWavMean.py Sad/2.wav
Loading library...
Analyzed by: OpenVokaturi version 4.0 for open-source projects, 2022-08-22
Distributed under the GNU General Public License, version 3 or later
Reading sound file...
   sample rate 20000.000 Hz
Allocating Vokaturi sample array...
   72000 samples, 1 channels
Creating VokaturiVoice...
Filling VokaturiVoice with samples...
Extracting emotions from VokaturiVoice...
Neutral: 0.004
Happy: 0.181
Sad: 0.810
Angry: 0.001
smsraj@sms-desktop:~/Desktop/OpenVokaturi-4-0/OpenVokaturi-4-0$
```

Documentation

Conference





Documentation

Banner

FeelSpeak: Generating Emotional Speech with Deep Learning

Department of Computer Science and Engineering
PES University, RR Campus, Bengaluru - 560085.

PROBLEM STATEMENT

FeelSpeak, This project presents an innovative approach to imbue synthesized speech with emotion, comprising a two-phase framework: training and testing. In the training phase, models are developed to detect emotions from labeled text data and to learn emotion-specific pitch, intensity, and modulation from labeled speech data. The testing phase involves converting input text to neutral speech using text-to-speech (TTS) methods, employing an emotion detection model to discern the emotion in the text, annotating the text with the detected emotion and utilizing a tacotron model to synthesize emotionally expressive speech. This method exemplifies the integration of machine learning techniques to seamlessly infuse emotion into speech, showcasing the potential for creating emotionally resonant audio from ordinary text inputs.

BACKGROUND

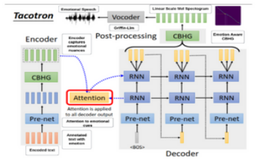
Through Literature survey we explored the advancements in text-to-speech (TTS) and emotion detection. "FastSpeech" introduces a fast and robust TTS system. "MTLTacotron" focuses on prosody modeling for improved voice quality. Another paper proposes a training strategy for Tacotron-based TTS to enhance speech styling. Lastly, a paper presents an emotion detection model using big data and LSTM, emphasizing careful preprocessing for accuracy.

DATASET AND FEATURES / PROJECT REQUIREMENTS/ PRODUCT FEATURES

In this project, emotion detection from text was achieved through the application of linear regression, Random Forest classifier, and SVM on a comprehensive dataset of 34,973 labeled entries. The second aspect involved infusing emotions into neutral speech using the Tacotron model, leveraging the LJ dataset consisting of 13,100 audio clips from LibriVox. These clips, recorded in 2016-17, featured a single speaker reading passages from non-fiction books published between 1884 and 1964. The diverse model selection for emotion detection ensured robust performance. Importantly, all datasets used, including LJ and the emotion-labeled dataset, are in the public domain, underscoring the project's commitment to ethical data usage.

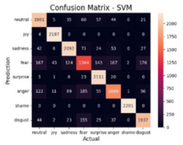
DESIGN APPROACH / METHODS

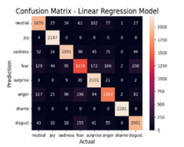
The Design approach for Emotion detection from text model and Speech synthesis model (Tacotron).

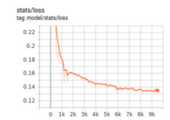


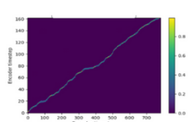
RESULTS AND DISCUSSION

The proposed framework offers a valuable contribution to the field of natural language processing and emotional speech synthesis.









Authors:

S M Sutharsan Raj

Rahul Roshan G

Rohit Roshan

M H Sohan

SUMMARY OF PROJECT OUTCOME

To summarize, our project introduces a groundbreaking framework for infusing synthesized speech with emotion through a two-phase process. The training phase develops models for emotion detection from labeled text and captures emotion-specific pitch, intensity, and modulation from labeled speech. Text-to-speech methods convert input text to neutral speech. In the testing phase, an emotion detection model, TTS, and a tacotron model seamlessly apply emotion-specific features to mel-spectrograms, resulting in emotionally expressive synthesized speech. The model can process up to 300 characters of text, highlighting the potential of machine learning for creating emotionally resonant audio. This framework represents a significant contribution to natural language processing and text-to-speech technology.

CONCLUSION AND FUTURE WORK

In conclusion, this paper presents a novel framework for adding emotions to synthesized speech. It involves training models for emotion detection and acquiring emotion-specific speech features. The framework then applies these features to mel-spectrograms, resulting in emotionally expressive speech. This work showcases the potential of machine learning in creating emotional audio and represents a significant contribution to natural language processing and text-to-speech technologies.Future works: Integrate the Tacotron-based TTS system into text editor read-aloud button. Use in any story books helps kids in classroom education. Help people with disabilty to listen to their favourite story book in a human way.Can be the voice to the text AI Assistant Can be extended to other accents.

REFERENCES

[1] Wang, Yuxuan, R. J. SkerryRyan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, 2018Navdeep Jaitly, Zongheng Yang, Ying Xiao, Z. Chen, Samy Bengio, Quoc V. Le, Yannis Agiomyrgiannakis, Robert A. J. Clark and Rif A. Saurous. “Tacotron: Towards End To End Speech Synthesis.” Interspeech (2017).”
[2] Rui Liu, Member, IEEE, Berrak Sisman, Member, IEEE, Guanglai Gao, Haizhou Li, Fellow, IEEE, 2021, “Expressive TTS Training with Frame and Style Reconstruction-Loss”,DOI 10.1109/TASLP.2021.3076369,IEEE/ACM-Transactions On Audio, Speech, and Language Processing

Prof. VR Badri Prasad

Documentation

Links

Git-hub link:
FEELSPEAK_GENERATING_EMOTIONAL_SPEECH_WITH_DEEP_LEARNING

Dataset:
LJ dataset: here
Emotion Detection from Text dataset: here

Pretrained Models:
EmoRoberta Model from Hugging Face transformers: here.
LeXmo python package used for emotion detection from text for comparison: here.

Conclusion and Future work

Integration:

- Integrate the Tacotron-based TTS system into text editor read-aloud button.
- Use in any story books helps kids in classroom education.
- Help people with disabilty to listen to their favourite story book in a human way.
- Can be the voice to the text AI Assitant
- Can be extended to other accents.

References

[1]Ren, Yi, Ruan, Yangjun, Tan, Xu, Qin, Tao, Zhao, Sheng, Zhao, Zhou, and Tie Liu. "FastSpeech: Fast, Robust and Controllable Text to Speech." ArXiv, (2019).

[2]Liu, Rui, et al. "Modeling prosodic phrasing with multitask learning in tacotron based TTS.

[3]Berrak Sisman, Member, IEEE, Guanglai Gao, Haizhou Li, Fellow, IEEE, 2021, “Expressive TTS Training with Frame and Style Reconstruction-Loss

[4]P. Chandra et al., "Contextual Emotion Detection in Text using Deep Learning and Big Data,2022

Thank You