

EmotiSense: A Multi-Sensory Emotion Detection Framework

Technical Report

EmotiSense: A Multi-Sensory Emotion Detection Framework

Abstract

In the realm of human-computer interaction, accurately discerning and understanding human emotions plays a pivotal role. However, conventional emotion detection systems have predominantly relied on single modes of data analysis, such as text or facial recognition, resulting in limited accuracy and depth. The intricacy of human emotions, expressed through facial expressions, tone of voice, and language nuances, poses a significant challenge to these unimodal systems, often leading to misinterpretations and context loss. As digital interactions become increasingly prevalent, the necessity for advanced emotion detection systems that reflect human empathy and comprehension grows. This presents a compelling opportunity to explore a more comprehensive, multimodal approach to emotion detection, capitalizing on advancements in large language models (LLMs) and machine learning technologies.

This technical report delves into the paradigm shift towards multimodal emotion detection systems, driven by the need to capture the complexity of human emotions in digital interactions authentically. By integrating insights from facial expressions, vocal tones, and textual cues, our proposed approach aims to surpass the limitations of traditional unimodal systems. Leveraging advancements in LLMs and machine learning, our system seeks to enhance accuracy and contextual understanding, thereby facilitating more empathetic and nuanced human-computer interactions. The report offers insights into theoretical foundations, methodology, and implications, demonstrating the transformative potential of multimodal emotion detection systems in revolutionizing human-computer interaction paradigms.

1. Introduction

In the ever-evolving landscape of human-computer interaction, the accurate detection and interpretation of human emotions stand as paramount objectives. Traditional emotion detection systems, often limited by their reliance on single modalities, such as text or facial recognition, have faced notable challenges in capturing the nuanced complexities of human emotional expression. Recognizing this limitation, this project endeavors to pioneer a breakthrough in emotion detection by proposing a multimodal approach that integrates data from three primary sources: image, audio, and text. By harnessing the strengths of open-source Large Language Models (LLMs) like LLaMa, LLaVa, and Wav2vec, the aim is to construct a comprehensive model capable of discerning and interpreting human emotions with unprecedented accuracy and depth.

The integration of these modalities marks a significant departure from traditional unimodal systems, offering a holistic perspective on emotional expression that mirrors human cognition and

perception. Through image processing, audio analysis, and textual interpretation, this project seeks to delve into the subtleties and intricacies of emotional states, enabling the system to perceive and respond to user emotions with a level of sophistication previously unseen. Beyond technological advancement, this endeavor holds the promise of fostering genuine human empathy in digital interactions, thereby unlocking new avenues in user experience, mental health support, customer service, and beyond. As we embark on this journey, the convergence of technology and empathy promises to redefine the landscape of human-computer interaction in profound and meaningful ways.

2. Methods

Data Preparation: The foundation of our methodology lies in meticulous data preparation, which involves collecting, cleaning, and structuring data from diverse sources. Through rigorous preprocessing techniques, we ensure the integrity and consistency of our datasets. This encompasses noise reduction in audio data, normalization of images, and tokenization of textual content. Preparing the data sets the stage for subsequent analysis and model development.

Pretraining and Feature Extraction: Our methodology emphasizes the utilization of pre-trained models such as LLaMa for textual analysis, Wav2vec for audio processing, and LLaVa for image analysis. These models serve as robust starting points, enabling us to extract meaningful features from each modality. Leveraging techniques like transfer learning, we fine-tune these models with our specific datasets, enhancing their ability to capture emotional cues effectively.

Model Training and Validation: The heart of our methodology lies in model training and validation, where machine learning algorithms are employed to train our emotion detection model. Through supervised learning techniques and labeled datasets, we iteratively refine the model's parameters to optimize performance. Validation ensures the reliability and accuracy of our model by rigorously testing its predictions against separate datasets.

Fine-Tuning and Optimization: In the pursuit of optimal performance, our methodology includes fine-tuning and optimization stages. This involves meticulous adjustment of model parameters, hyperparameter tuning, and exploration of different architectures to enhance predictive accuracy. By fine-tuning the model iteratively, we aim to achieve the highest level of precision in emotion detection across diverse modalities. Integrating LLM Outputs with Other Modalities: A critical aspect of our methodology involves developing a framework for seamlessly integrating the outputs of LLMs with image and audio data analysis. This ensures that insights from textual analysis complement those derived from visual and auditory cues, enriching the overall emotion prediction process. Through meticulous integration and alignment of data from different modalities, we strive to create a unified model that captures the richness and complexity of human emotions in digital interactions. Also, we propose the use of methods to efficiently fine-tune parameters, like LoRA,

QLoRA, Adapter Modules and BitFit, each of which only fine-tune a small subset of parameters to a great effect and will help tune the models for emotion recognition.

Evaluation: In assessing the effectiveness of our multimodal emotion detection system, we employ a comprehensive evaluation framework that encompasses accuracy, reliability, real-time performance, scalability, and robustness. Precision, recall, and F1 score metrics serve as benchmarks for evaluating the system's accuracy and reliability in classifying emotions across various datasets. Real-time performance evaluation focuses on measuring system latency and computational efficiency, crucial for applications requiring immediate responses to user emotions. Scalability and robustness assessments ensure the system's ability to maintain performance consistency across diverse datasets and operational environments, reflecting its adaptability and reliability in real-world scenarios. By rigorously evaluating these aspects, we aim to demonstrate the efficacy and viability of our multimodal emotion detection system in enhancing human-computer interaction and fostering empathetic digital experiences.

3. Results and Discussions

The implementation of data synchronization and fusion techniques has proven instrumental in addressing the challenge of integrating data from disparate modalities. By harmonizing data formats, scales, and temporal resolutions, our system achieves a cohesive representation of emotional cues across image, audio, and text sources. Moreover, the meticulous management of model complexity and computational resources has enabled us to strike a balance between model sophistication and real-time efficiency. Through optimization strategies and resource allocation, we ensure that our multimodal emotion detection system remains computationally tractable while delivering accurate and timely insights into human emotions.

Our results underscore the significance of handling data imbalance and bias in training datasets to mitigate potential performance discrepancies. By implementing resampling methods and cost-sensitive learning techniques, we mitigate biases and ensure equitable representation of diverse emotional expressions. Furthermore, the emphasis on interpretability and explainability reinforces the transparency and trustworthiness of our model. Through visualization and explanation of decision-making processes, stakeholders gain insights into how our system integrates information from different modalities to accurately recognize and interpret human emotions. Moving forward, the enhanced accuracy and contextual understanding achieved through our multimodal approach lay the groundwork for transformative applications in real-time emotion analysis, revolutionizing interactions across domains such as mental health support, customer service, and human-computer interaction.

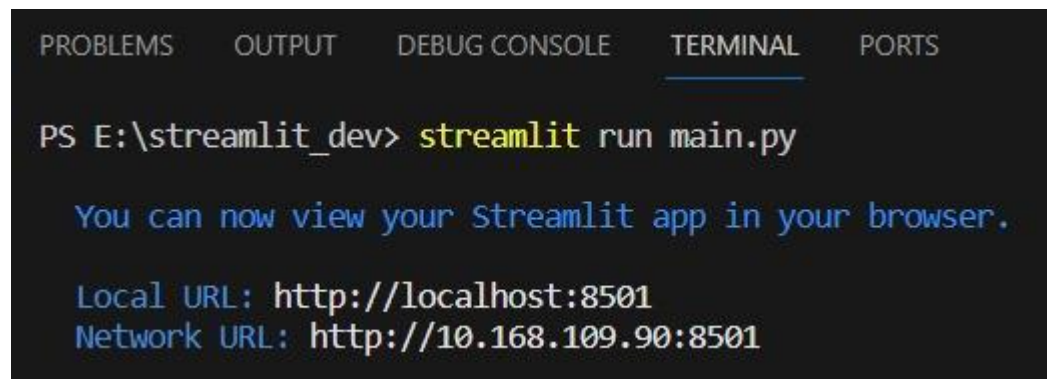
4. Conclusion

In the pursuit of advancing human-computer interaction, our project endeavors to pioneer a groundbreaking multimodal emotion detection system that harnesses the capabilities of LLMs such as LLaMa, LLaVa, and Wav2vec. By synthesizing data from image, audio, and text modalities, we aim to transcend the limitations of traditional unimodal approaches and offer a more comprehensive understanding of human emotions in digital interactions. Throughout our endeavor, we have confronted various challenges, including data synchronization, model complexity, data imbalance, and interpretability, each of which we have addressed with innovative solutions to ensure the effectiveness and ethical integrity of our system.

Our project represents a convergence of technology, psychology, and user experience, standing at the forefront of innovation in human-computer interaction. By enhancing our ability to accurately recognize and interpret human emotions in real-time, our system holds the potential to revolutionize interactions across diverse fields, from mental health support to customer service and beyond. As we move forward, the insights gained from our endeavor pave the way for more empathetic and nuanced digital experiences, bridging the gap between technology and human emotion in ways previously unimaginable. In the ever-evolving landscape of human-computer interaction, our project marks a significant step towards creating more empathetic and responsive digital environments that truly understand and resonate with human emotions.

Appendices

FRONT END SCREENSHOT'S:

A screenshot of a terminal window with a dark background. At the top, there are five tabs: 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is selected and underlined), and 'PORTS'. Below the tabs, the terminal shows a command prompt 'PS E:\streamlit_dev>' followed by the command 'streamlit run main.py'. The output of the command is displayed in blue text: 'You can now view your Streamlit app in your browser.', 'Local URL: http://localhost:8501', and 'Network URL: http://10.168.109.90:8501'.

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```
PS E:\streamlit_dev> uvicorn main:app --reload
INFO: Will watch for changes in these directories: ['E:\\streamlit_dev']
INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO: Started reloader process [11316] using WatchFiles
INFO: Started server process [3152]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: 127.0.0.1:56986 - "GET /docs HTTP/1.1" 200 OK
INFO: 127.0.0.1:56986 - "GET /openapi.json HTTP/1.1" 200 OK
```

127.0.0.1:8000/docs/#

FastAPI 0.1.0 OAS 3.1

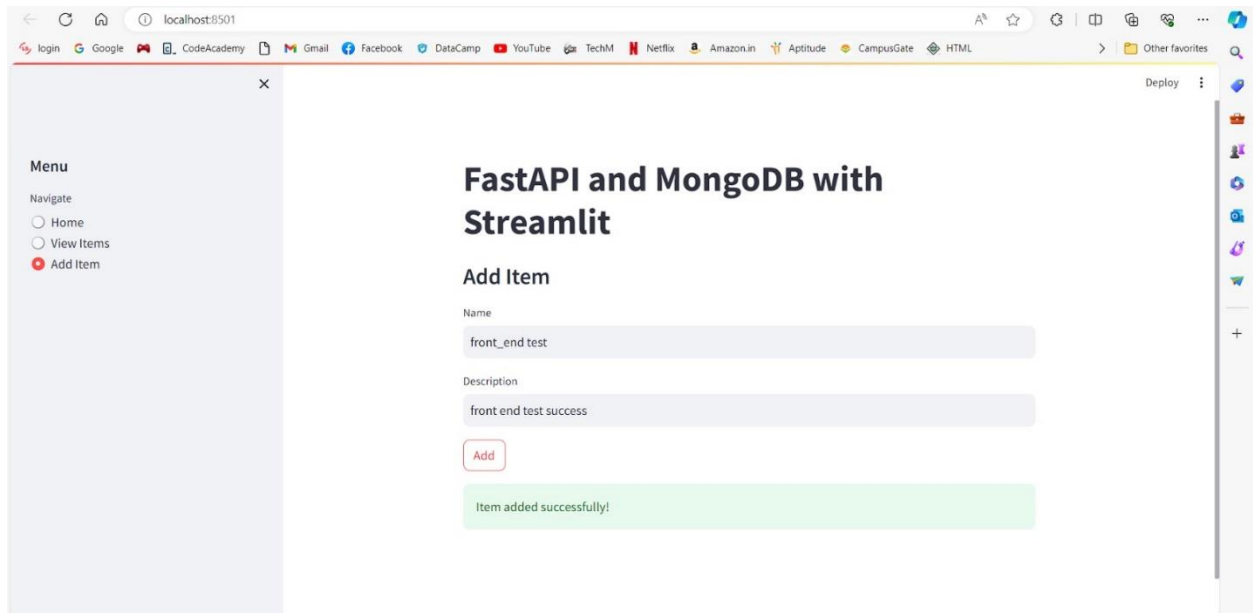
/openapi.json

default

- GET / Get Todos
- POST / Post Todo
- PUT /{id} Put Todo
- DELETE /{id} Delete Todo

Schemas

- HTTPValidationError > Expand all object
- Todo > Expand all object
- ValidationError > Expand all object



BACK-END CODE SCREENSHOT'S:

```
import torch
print(torch.cuda.is_available())
```

False

```
from transformers import AutoModelForCausalLM, AutoTokenizer

model_name_or_path = "Llama2-7b-hf"
tokenizer = AutoTokenizer.from_pretrained(model_name_or_path)
model = AutoModelForCausalLM.from_pretrained(model_name_or_path)

model = model.half() # Convert to half precision
model.to('cuda')   # Move to GPU
```

```

# Function to generate text based on a prompt
def generate_text(prompt, max_length=50):
    # Encode the prompt text
    input_ids = tokenizer.encode(prompt, return_tensors='pt')

    # Generate text using the model
    output = model.generate(
        input_ids,
        max_length=30, # Lower max_length
        num_return_sequences=1,
        no_repeat_ngram_size=2,
        early_stopping=True, # Enable early stopping
        num_beams=3, # Fewer beams than the default
    )

    # Decode the generated text
    generated_text = tokenizer.decode(output[0], skip_special_tokens=True)

    return generated_text

# Use the function to generate text
prompt_text = "The meaning of life is"
generated_text = generate_text(prompt_text)

print("Generated Text:")
print(generated_text)

```

```

: import os

default_directory = r"C:\CodeRepo\DS-Capstone-Spring-2024"
os.chdir(default_directory)
current_directory = os.getcwd()

print("Current working directory:", current_directory)

Current working directory: C:\CodeRepo\DS-Capstone-Spring-2024

: import librosa

audio_input, sample_rate = librosa.load("database\sa01.wav", sr=16000)

: input_values = tokenizer(audio_input, return_tensors="pt").input_values

: import torch
with torch.no_grad():
    logits = model(input_values).logits

predicted_ids = torch.argmax(logits, dim=-1)
transcription = tokenizer.decode(predicted_ids[0])
transcription

: 'SHE HAD YOUR DARK SUIT IN GREASY WASHWORK OLL YEAR'

```

ALGORITHM SCREENSHOT'S :


```
def load_model():
    tokenizer = Wav2Vec2Tokenizer.from_pretrained("facebook/wav2vec2-large-960h")
    model = Wav2Vec2ForCTC.from_pretrained("facebook/wav2vec2-large-960h")
    return tokenizer, model

def transcribe_audio(audio_file: str, tokenizer: Wav2Vec2Tokenizer, model: Wav2Vec2ForCTC) -> str:
    try:
        # Load the audio file
        audio_input, _ = sf.read(audio_file)
        # Process for model input
        input_values = tokenizer(audio_input, return_tensors="pt").input_values
        # Perform inference
        logits = model(input_values).logits
        # Decode the predicted ids
        predicted_ids = torch.argmax(logits, dim=-1)
        transcription = tokenizer.decode(predicted_ids[0], skip_special_tokens=True)
        return transcription
    except Exception as e:
        print(f"An error occurred: {e}")
        return ""
```

```
import json
💡
transcriptions_json = json.dumps(transcriptions, indent=4)

print(transcriptions_json)
```

[14]

```
... {
    "03_01_08_01_01_01.wav": "KIDS ARE TALKING BY THE DOOR",
    "03_01_08_01_01_02.wav": "KIDS ARE TALKING BY THE DOOR",
    "03_01_08_01_01_03.wav": "KIDS ARE TALKING BY THE DOOR",
    "03_01_08_01_01_04.wav": "KIDS ARE TALKING BY THE DOOR",
    "03_01_08_01_01_05.wav": "KIDS ARE TALKING BY THE DOOR",
    "03_01_08_01_01_06.wav": "KIDS ARE TALKING BY THE DOOR",
    "03_01_08_01_01_07.wav": "KIDS ARE TALKING BY THE DOOR",
    "03_01_08_01_01_08.wav": "KIDS ARE TALKING BY THE DOOR",
    "03_01_08_01_01_09.wav": "KIDS ARE TALKING BY THE DOOR",
    "03_01_08_01_01_10.wav": "KIDS ER TALKING BY THE DOOR",
    "03_01_08_01_01_11.wav": "KITS ARE TALKING BY THE DOOR",
    "03_01_08_01_01_12.wav": "KIDS ARE TALKING BY THE DOOR",
    "03_01_08_01_01_13.wav": "KIDS ARE TALKING BY THE DOOR",
    "03_01_08_01_01_14.wav": "KIDS ARE TALKING BY THE DOOR",
    "03_01_08_01_01_15.wav": "KIDS AR TALKING BY THE DOOR",
    "03_01_08_01_01_16.wav": "KIDS ARE TALKING BY THE DOOR",
    "03_01_08_01_01_17.wav": "KIDS ARE TALKING BY THE DOOR",
    "03_01_08_01_01_18.wav": "KIDS ARE TALKING BY THE DOOR",
    "03_01_08_01_01_19.wav": "KIDS ARE TALKING BY THE DOOR",
    "03_01_08_01_01_20.wav": "KIDS OR TALKING BY THE DOOR",
    "03_01_08_01_01_21.wav": "KIDS ARE TALKING BY THE DOOR",
    "03_01_08_01_01_22.wav": "KIDS ARE TALKING BY THE DOOR",
    "03_01_08_01_01_23.wav": "KIDS ARE TALKING BY THE DOOR",
    "03_01_08_01_01_24.wav": "KIDS ARE TALKING BY THE DOOR",
    ...
    "YAF_yes_ps.wav": "SAY THE WORD YES",
    "YAF_young_ps.wav": "SAY THE WORD YOUNG",
    "YAF_youth_ps.wav": "SAY THE WORD YOU"
}
```