MSA Challenge @ The 4th Pazhou AI Competition

Cross-Lingual Multimodal Sentiment Analysis System

Team Members: YanSong Hu, YangLe Ma, ZhouYang Wang, RiJie Hao

CONTENTS

01 Cross-Lingual System

Predicts fine-grained sentiment scores from raw video inputs across languages

02 Dual Architecture

Upstream feature extraction and downstream multimodal fusion framework

03 User Interface

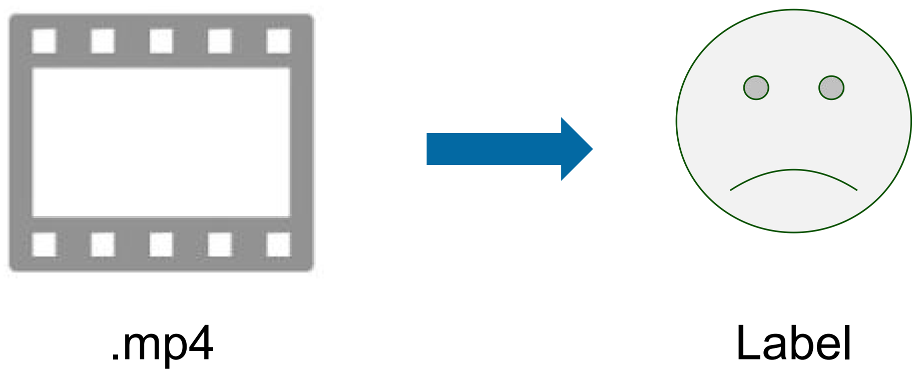
Flexible execution modes for reproducibility, testing, and human-computer interaction

04 Interpretability

Transparency modules provide insight into modality contributions for trustworthy results

01. Project Overview

- Objective: Predict fine-grained sentiment scores (SNEG/WNEG/NEUT/WPOS/SPOS) from raw video inputs.
- Architecture: Dual-architecture design with upstream feature extraction and downstream Transformer cross-attention fusion.
- Capabilities: Bilingual (Chinese/English) processing, feature reuse, and user-friendly interaction.



02. Upstream Feature Extraction

Text Modality

A: XLM-R → 768-D multilingual embeddings.

B: Whisper transcribes; language BERTs → 768-D → per-language projection → shared 256-D + lang embedding.

Audio Modality

Extracts MFCCs, pitch, spectral properties, harmonic-to-noise ratio. Aggregated into 40-D representation for prosodic cues.

Visual Modality

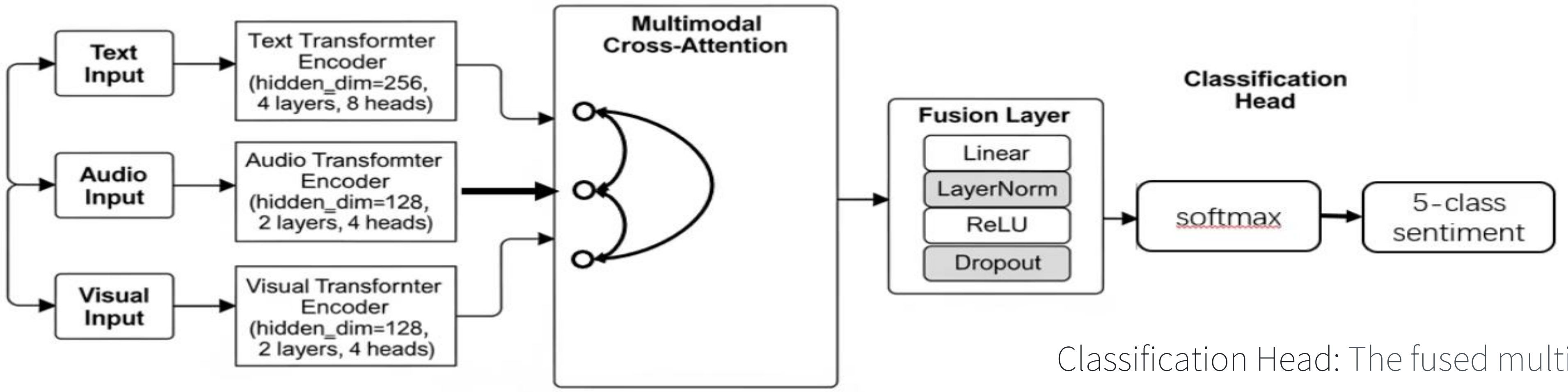
MediaPipe face tracking + HoG texture analysis + Facial Action Units (AU1-AU20) (Resnet18) create 512-D facial expression descriptor.

All features standardized to fixed dimensions, stored in .pkl format with dataset metadata for seamless integration.

03. Downstream Fusion Framework

01 Transformer Encoding: Text features are encoded with a Transformer (256 dimensions, 4 layers, 8 heads). Audio and Visual features share a smaller Transformer (128 dimensions, 2 layers, 4 heads).

02 Cross-Attention Fusion: A bidirectional cross-attention mechanism allows each modality to query and attend to the others, generating rich, context-aware joint vectors.



03 Classification Head: The fused multi-modal representations are passed through a final classification layer to predict the 5-class sentiment scores (SNEG/WNEG/NEUT/WPOS/SPOS).

04. Training Settings & Optimization

- Optimization: AdamW optimizer combined with CosineAnnealingLR learning rate scheduler for stable convergence.
- Training Control: 20 epochs total, with early stopping (patience=10) to prevent overfitting and gradient clipping (max norm 1.0) to stabilize training.
- Evaluation: Comprehensive metrics including Accuracy, Macro F1, Confusion Matrix analysis, and Ablation Studies to validate component contributions.

Key Hyperparameters

Parameter	Value
Learning Rate	1×10^{-4}
Batch Size	32
Dropout Rate	0.3
Optimizer	AdamW
Scheduler	CosineAnnealingLR

05. Human-Computer Interaction

Two-Step Workflow (Recommended)

video2pkl.py → main.py

Feature reuse for efficiency

One-Click Execution

test_scripts.py

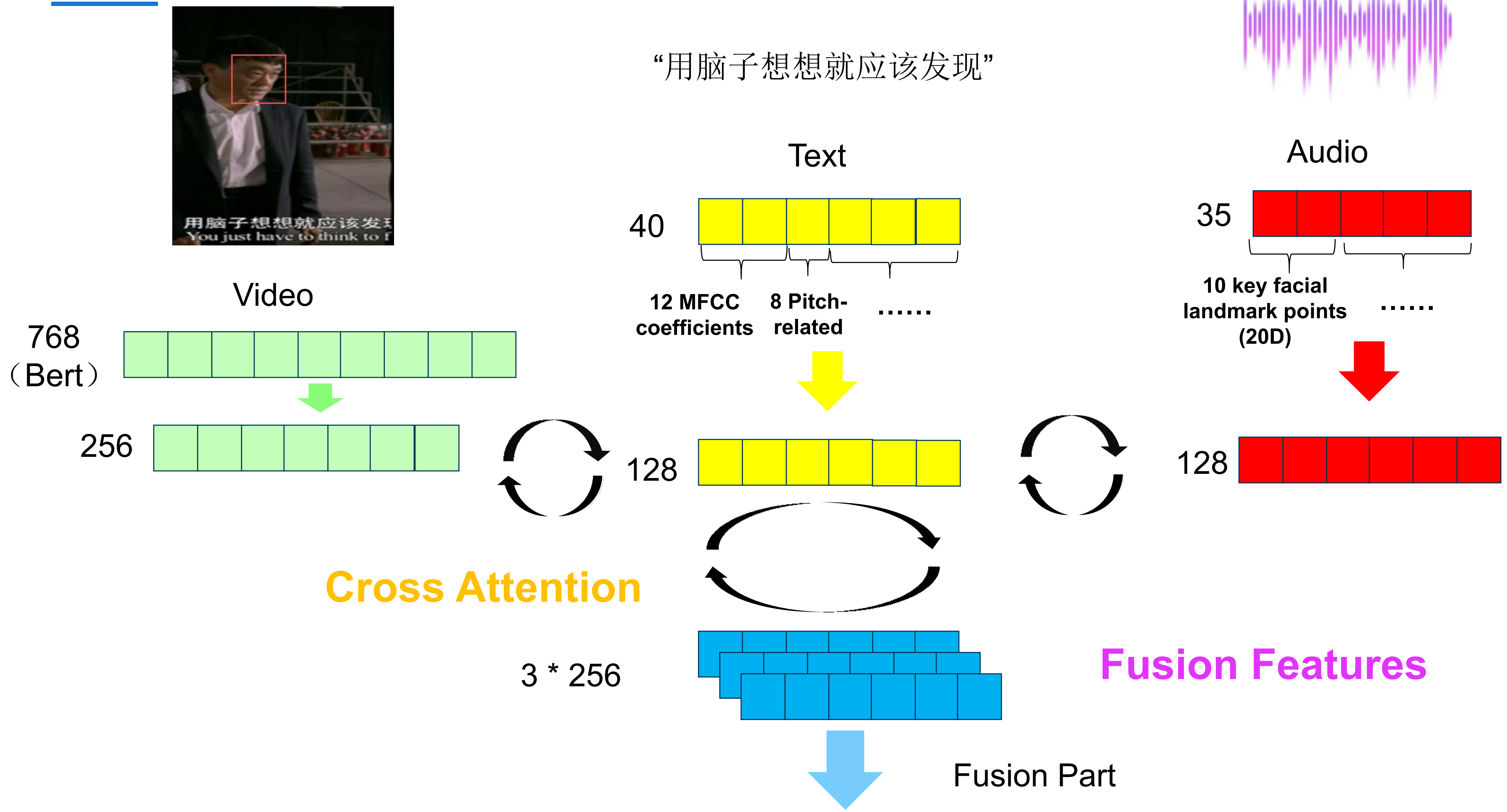
Small-scale, rapid testing

Direct Video Inference

MSAbyvideo/main.py

End-to-end prediction

06. Model Interpretability



06. Model Interpretability

- Attention Visualization:

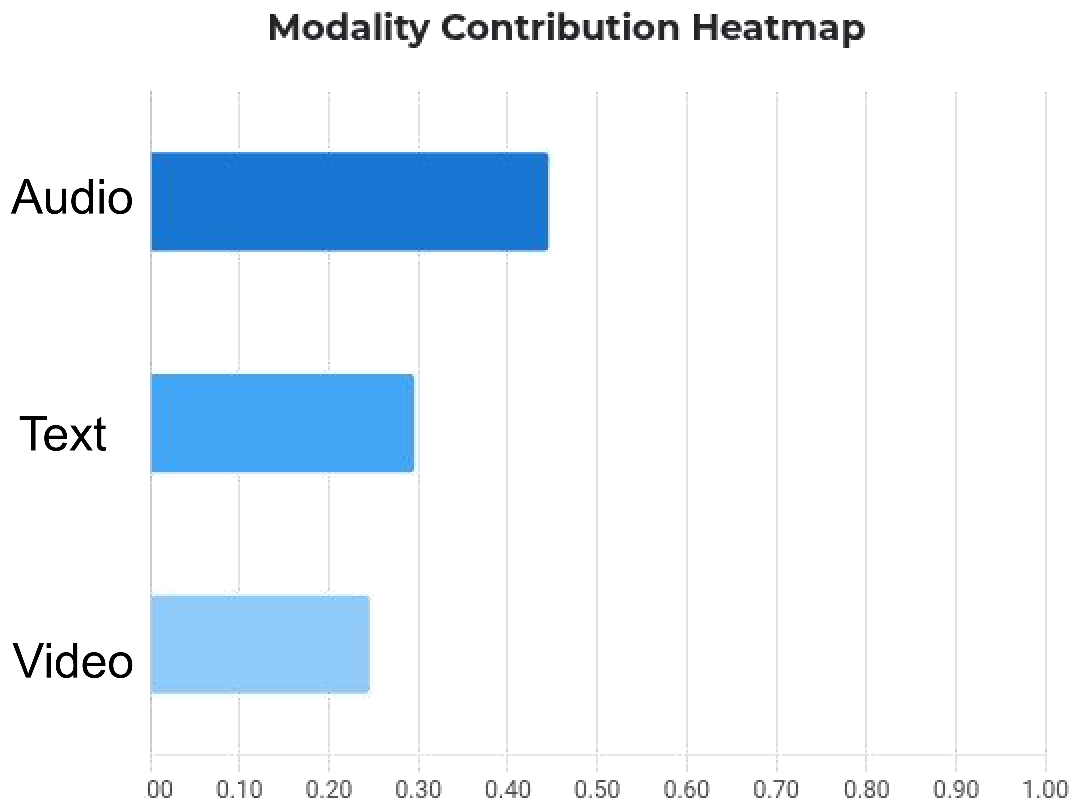
  Records cross-attention layer weight matrices to show how the model focuses on different modalities.

- Heatmap Visualization:

  Intuitively displays the contribution of each modality (text, audio, visual) to the final prediction.

- Value:

  Provides transparent integration logic, facilitating error analysis and guiding future model optimizations.



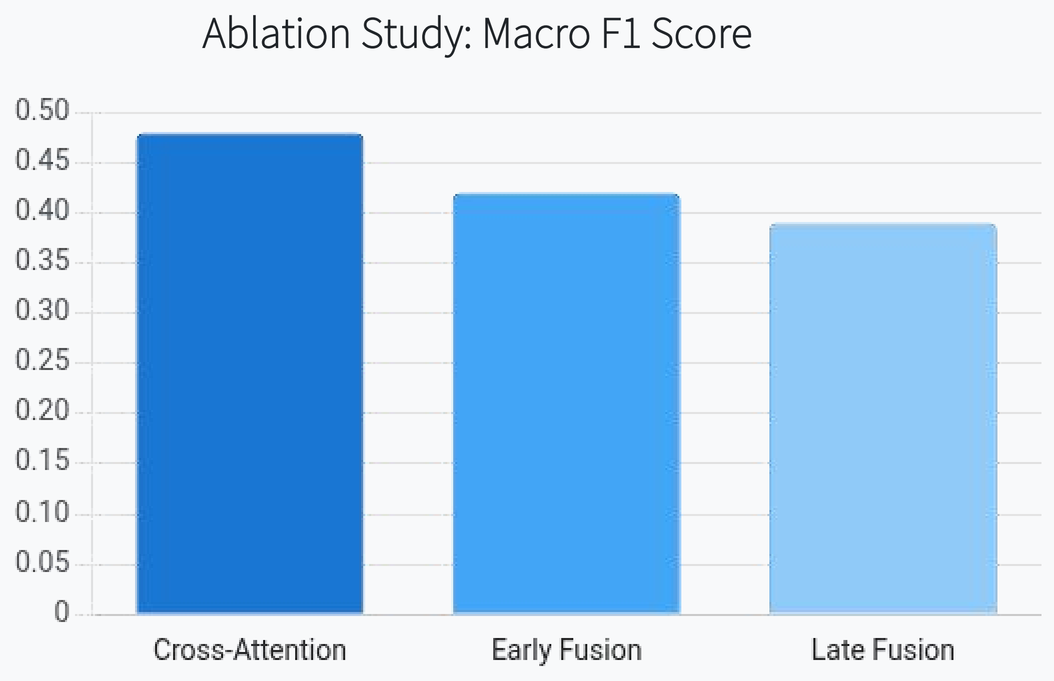
07. Performance Results & Analysis

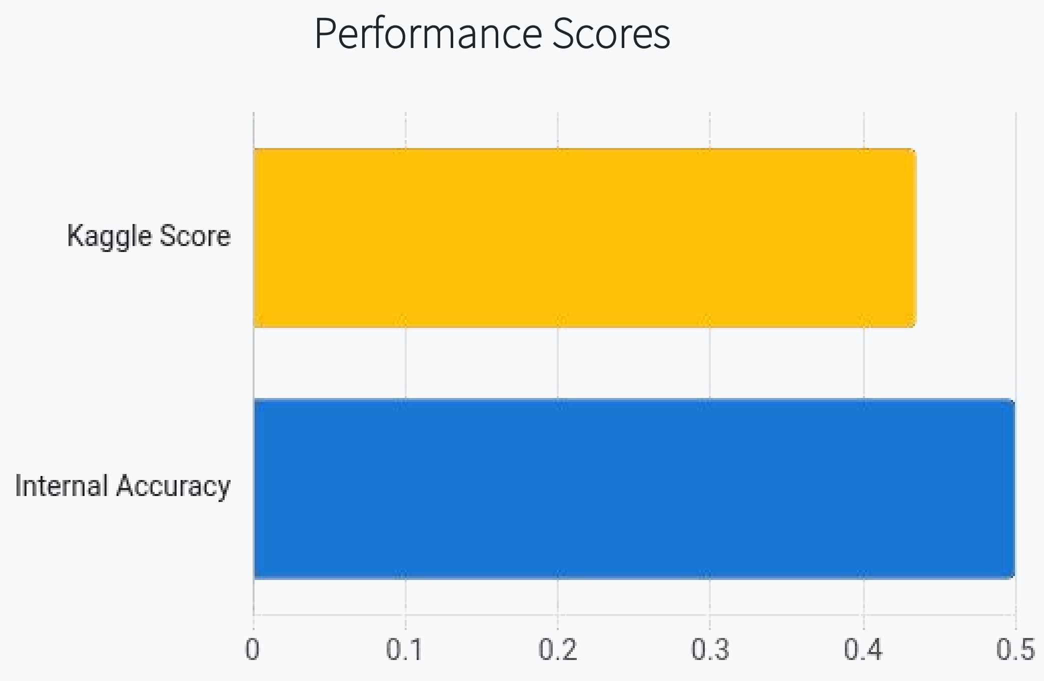
Kaggle Score
0.4350
(baseline, CPU-trained)

Internal Accuracy
>50%

Ablation Study
Cross-Attention
> Early > Late Fusion

Bottleneck
CPU Limits
GPU expected to improve





08. Project Structure & Quick Start

Project Structure

```
MSA/
  ├──── test_script.py (one-click test)
  ├──── MSAbypkl/ (PKL workflow)
  ├──── MSAbyvideo/ (direct video workflow)
  ├──── video2pkl/ (feature extractor)
  └──── best_models/ (pretrained weights)
```

https://github.com/19376357/Bilingual-Multimodal-Sentiment-Analysis

01 Install Dependencies:

`pip install -r requirements.txt`

02 One-Click Execution:

`python test_script.py`

03 Two-Step Workflow:

Extract Features →Train/Evaluate

09. Problem Solving & Future Outlook

Troubleshooting

FFmpeg Installation: Resolve missing FFmpeg errors on Windows with the command:

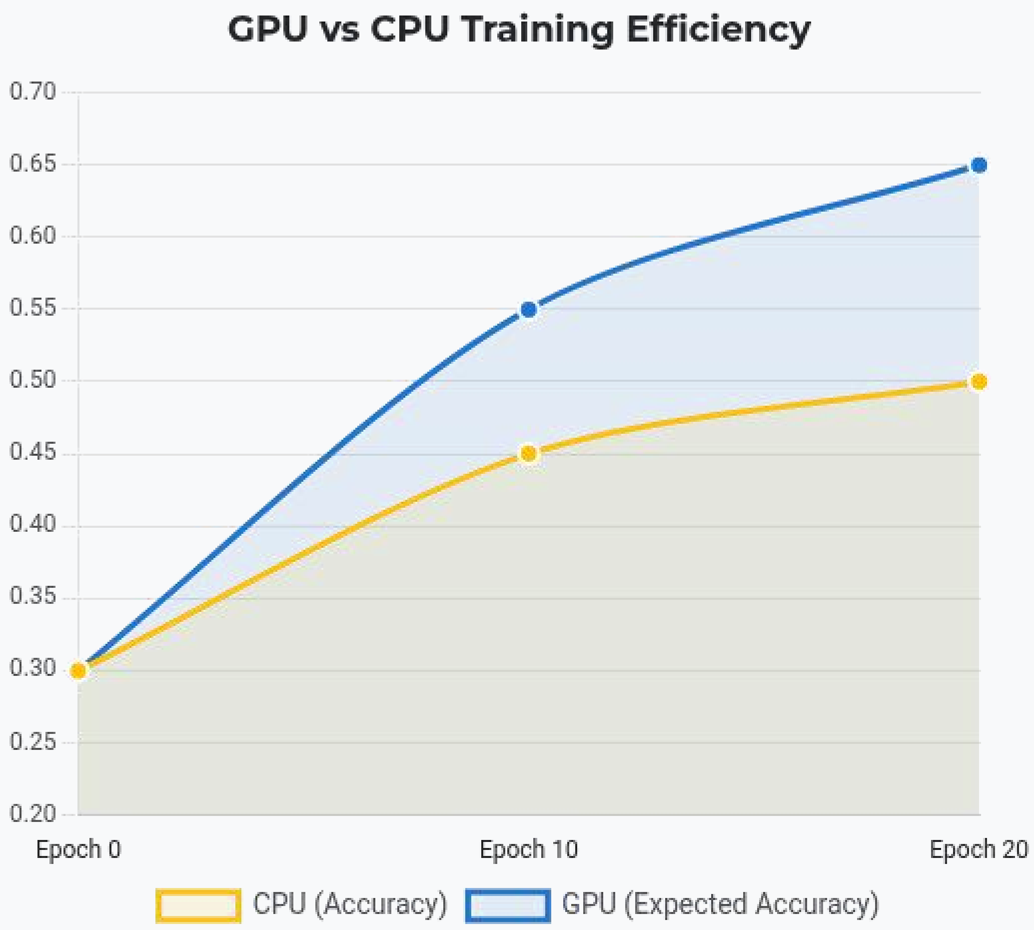
`winget install Gyan.FFmpeg`

Model Files: Ensure pretrained weights are downloaded and placed in the correct directory:

`best_models/`

Future Plans

- GPU training for faster convergence and improved model performance.
- Optimize cross-lingual transfer learning to better handle multilingual inputs.
- Expand to finer-grained tasks, such as emotion detection or sarcasm identification.



Acknowledgements

Special thanks to the organizing committee of The 4th Pazhou AI Competition.
Gratitude to team members for collaborative development.
Appreciation for open-source communities supporting Whisper, BERT, and MediaPipe.