به نام خدا

گزارش پروژه اول درس یادگیری ماشین

بازسازی و بهبود کیفیت تصاویر چهره با استفاده از شبکه‌های عصبی کانولوشنی (CNN)

هدی عطاری - 401243068

1. مقدمه

هدف اصلی این پروژه، طبقه‌بندی احساسات گوینده در یکی از ۷ دسته‌ی استاندارد هیجانی شامل: خشم (Anger)، تنفر (Disgust)، ترس (Fear)، شادی (Joy)، خنثی (Neutral)، غم (Sadness) و تعجب (Surprise) است.

ورودی سیستم برای هر نمونه داده شامل موارد زیر است:

1. رونوشت متنی (Transcript): متن صحبت‌های گوینده.
2. سیگنال صوتی (Raw Audio): فایل صوتی مربوط به همان گفتار.

مدل پیشنهادی باید بتواند ویژگی‌های معنایی را از متن و ویژگی‌های آکوستیک را از صدا استخراج کرده و با ترکیب آن‌ها (Fusion)، برچسب احساسی نهایی را پیش‌بینی کند.

۱-۳. معرفی دادگان (Dataset)

برای این منظور از مجموعه‌داده‌ی استاندارد MELD (Multimodal EmotionLines Dataset) استفاده شده است. این دیتاست شامل دیالوگ‌های سریال تلویزیونی *Friends* است و ویژگی‌های زیر را دارد که آن را برای این پروژه مناسب می‌سازد:

- وجود همزمان فایل صوتی و متن برای هر گفتار (Utterance).
- برچسب‌گذاری دقیق با ۷ کلاس احساسی مورد نظر پروژه.
- چالش‌های واقعی مانند تفاوت در گویندگان، وجود نویز محیطی و طول متغیر جملات که به سنجش تعمیم‌پذیری مدل کمک می‌کند.

۱-۴. رویکرد کلی (Methodology Overview)

در این پژوهش از رویکرد Late Fusion (ترکیب دیرهنگام) استفاده شده است. معماری سیستم شامل دو بازوی اصلی است:

- بازوی متنی: استفاده از مدل زبانی پیش‌آموزش‌دیده RoBERTa برای استخراج ویژگی‌های متنی.

- بازوی صوتی: استفاده از مدل Wav2Vec2 برای استخراج ویژگی‌های صوتی از سیگنال خام.

این دو بازو ویژگی‌های سطح بالا را استخراج کرده و سپس این ویژگی‌ها در یک لایه‌ی مشترک با هم ترکیب می‌شوند تا تصمیم نهایی توسط طبقه‌بند اتخاذ شود. جزئیات دقیق پیش‌پردازش و معماری در بخش های آینده تشریح خواهد شد.

2. آماده‌سازی داده‌ها و پیش‌پردازش

۲-۱. پیاده‌سازی و تنظیمات محیطی (Setup & Reproducibility)

برای پیاده‌سازی این پروژه از زبان برنامه‌نویسی Python و کتابخانه‌های یادگیری عمیق PyTorch و Transformers (HuggingFace) استفاده شده است. با توجه به اهمیت بازتولیدپذیری (Reproducibility) که در صورت پروژه ذکر شده است، در ابتدای کد، یک تابع set_seed تعریف شد تا تمامی هسته‌های تصادفی (Random Seeds) در کتابخانه‌های random، numpy و torch روی مقدار ثابت 42 تنظیم شوند. این عمل تضمین می‌کند که نتایج آموزش مدل، تقسیم‌بندی داده‌ها و وزن‌دهی اولیه در اجراهای مختلف یکسان باقی بماند.

۲-۲. دریافت و استخراج داده‌های خام

مجموعه‌داده‌ی MELD.Raw که حجم بالایی (حدود ۱۰ گیگابایت) دارد، به صورت مستقیم از سرور دانشگاه میشیگان دانلود شد. از آنجا که فایل اصلی به صورت فشرده (.tar.gz) بود، یک پایپ‌لاین خودکار برای استخراج فایل‌ها در محیط کولب پیاده‌سازی شد. ساختار داده‌های خام برای آموزش مدل چندوجهی، نیاز به تفکیک سیگنال صوتی از تصویر داشتیم.

۲-۳. خط لوله تبدیل فرمت صوتی (Audio Conversion Pipeline)

یکی از چالش‌های اصلی در استفاده از داده‌های چندرسانه‌ای خام، ناهمگونی فرمت‌هاست. مدل‌های صوتی پیشرفته مانند Wav2Vec 2.0 برای عملکرد صحیح نیاز به سیگنال صوتی خام (Raw Waveform) با نرخ نمونه‌برداری (Sampling Rate) خاص دارند.

برای حل این مسئله، یک اسکریپت پیش‌پردازش توسعه داده شد که مراحل زیر را به صورت خودکار انجام می‌دهد:

1. پیمایش فایل‌ها: جستجوی تمامی فایل‌های .mp4 در زیرپوشه‌های استخراج شده.
2. استخراج و تبدیل: استفاده از ابزار قدرتمند FFmpeg برای استخراج صدای ویدیوها.

3. استانداردسازی: تبدیل تمامی فایل‌های صوتی به فرمت .wav با نرخ نمونه‌برداری 16kHz و به صورت تک‌کاناله (Mono).

دلیل فنی انتخاب 16kHz: مدل Wav2Vec 2.0 که در این پروژه به عنوان استخراج‌گر ویژگی (Feature Extractor) استفاده می‌شود، روی داده‌های صوتی 16kHz پیش‌آموزش دیده است. عدم تطابق نرخ نمونه‌برداری می‌تواند منجر به افت شدید دقت مدل شود.

۳. طراحی پایپ‌لاین ورودی و کلاس داده (Dataset Implementation)

۳-۱. چالش‌های ساختاری و راهکار ایندکس‌گذاری (Robust Indexing)

یکی از چالش‌های رایج در کار با دیتاست‌های بزرگ چندرسانه‌ای مانند MELD، عدم یکپارچگی در ساختار پوشه‌بندی و نام‌گذاری فایل‌هاست. در فایل‌های خام استخراج شده، برخی فایل‌های صوتی در زیرپوشه‌های تو در تو قرار داشته و برخی نام‌گذاری‌ها دارای پسوندهای متفاوت (مانند .mp4.wav در برابر .wav) بودند.

برای حل این مشکل و جلوگیری از خطای FileNotFoundError در حین آموزش، به جای تکیه بر مسیرهای ثابت، از یک استراتژی ایندکس‌گذاری سراسری (Global Indexing) استفاده شد.

- روش کار: با استفاده از کتابخانه glob، ابتدا تمام فایل‌های .wav موجود در محیط دیسک اسکن شده و یک دیکشنری مرجع (AUDIO_FILE_MAP) ایجاد گردید که نام فایل را به مسیر مطلق آن نگاشت می‌کند.
- مزیت: این کار باعث شد کلاس Dataset بدون وابستگی به ساختار پوشه‌ها، تنها با داشتن Dialogue_ID و Utterance_ID بتواند فایل صوتی صحیح را پیدا کند.

۳-۲. طراحی کلاس MELDMultimodalDataset

برای مدیریت همزمان داده‌های متنی و صوتی، کلاس اختصاصی MELDMultimodalDataset با ارث‌بری از کلاس پایه torch.utils.data.Dataset پیاده‌سازی شد. وظیفه اصلی این کلاس، خواندن رکوردها از فایل CSV، پیدا کردن فایل صوتی متناظر، اعمال پیش‌پردازش‌های لازم و بازگرداندن یک دیکشنری شامل تنسورهای آماده برای مدل است.

الف) پیش‌پردازش متن (Text Preprocessing)

برای پردازش متن از توکن‌ساز مدل (RoBERTa (RobertaTokenizer استفاده شد. مراحل زیر برای هر جمله اعمال می‌شود:

1. Tokenization: تبدیل متن به توکن‌های عددی قابل فهم برای مدل.
2. Padding & Truncation: تمامی جملات به طول ثابت ۱۲۸ توکن همسان‌سازی شدند. جملات کوتاه‌تر با توکن مخصوص (pad_token) پر شده و جملات بلندتر برش خورده‌اند.
3. خروجی: تولید input_ids و attention_mask که ورودی‌های استاندارد مدل‌های ترنسفورمر هستند.

ب) پیش‌پردازش سیگنال صوتی (Audio Preprocessing)

پردازش صوت به دلیل ماهیت سیگنال پیوسته، پیچیدگی بیشتری دارد. مراحل زیر در تابع __getitem__ پیاده‌سازی شده است:

1. بارگذاری (Loading): فایل صوتی با استفاده از torchaudio بارگذاری می‌شود.
2. Resampling: اگر نرخ نمونه‌برداری فایل متفاوت باشد، به صورت اجباری به 16kHz تبدیل می‌شود (الزام مدل Wav2Vec2).
3. Mono Conversion: اگر صدا استریو (دو کاناله) باشد، با میانگین‌گیری از کانال‌ها به مونو تبدیل می‌شود.
4. تثبیت طول زمانی (Fixed Formatting): برای امکان دسته‌بندی (Batching) داده‌ها، طول تمامی سیگنال‌های صوتی به ۵ ثانیه (معادل ۸۰,۰۰۰ نمونه) محدود شد.
   - صداهای بلندتر برش (Clip) می‌خورند.
   - صداهای کوتاه‌تر توسط Wav2Vec2Processor به صورت خودکار پدینگ (Padding) می‌شوند.
5. Feature Extraction: سیگنال خام نهایی به Wav2Vec2Processor داده می‌شود تا نرمال‌سازی (Zero-mean Unit-variance) انجام شده و تنسور input_values تولید شود.

۳-۳. همگام‌سازی و پاک‌سازی داده‌ها

در مرحله اولیه ساخت شیء از کلاس Dataset، یک مرحله پاک‌سازی (_clean_and_match) اجرا می‌شود. در این مرحله، ردیف‌هایی از فایل CSV که فایل صوتی متناظر آن‌ها (حتی با جستجوی هوشمند) یافت نشد، از دیتاست حذف می‌شوند تا فرآیند آموزش با خطا مواجه نگردد. همچنین برچسب‌های احساسی متنی (مانند ‘joy’,‘ sadness”) با استفاده از دیکشنری EMOTION_MAP به اعداد صحیح (۰ تا ۶) تبدیل می‌شوند.

۴. معماری مدل پیشنهادی و استراتژی آموزش

۴-۱. نمای کلی معماری (Architecture Overview)

از رویکرد تلفیق دیرهنگام (Late Fusion) برای ترکیب اطلاعات متنی و صوتی استفاده کردم. در این معماری، هر مدالیته ابتدا توسط یک شبکه عصبی عمیق مستقل (Encoder) پردازش شده و به یک بردار ویژگی فشرده تبدیل

می‌شود. سپس این بردارها به یکدیگر متصل (Concatenate) شده و وارد یک شبکه تمام‌متصل نهایی (Classifier) می‌شوند تا کلاس احساسی پیش‌بینی شود.

معماری کلی مدل شامل سه بخش اصلی است:

1. Text Encoder: مدل زبانی RoBERTa.
2. Audio Encoder: مدل صوتی Wav2Vec 2.0.
3. Fusion Head: لایه‌های ترکیب و طبقه‌بندی.

۴-۲. انکودر متنی (Text Branch)

برای پردازش متن، از مدل پیش‌آموزش‌دیده roberta-base استفاده شد که نسخه بهینه‌شده‌ی BERT است که توانایی بالایی در درک بافت معنایی جملات دارد.

- ورودی: توکن‌های جمله (input_ids) و ماسک توجه (attention_mask).
- خروجی: بردار متناظر با توکن ویژه [CLS] از آخرین لایه پنهان مدل استخراج می‌شود. این بردار نمایانگر معنای کل جمله است و ابعادی برابر با 768 دارد.

۴-۳. انکودر صوتی (Audio Branch)

برای استخراج ویژگی‌های آکوستیک، از مدل قدرتمند wav2vec2-base-960h استفاده شده است. این مدل که روی ۹۶۰ ساعت گفتار آموزش دیده، سیگنال خام صوتی را به ویژگی‌های سطح بالا تبدیل می‌کند.

- چالش طول متغیر: خروجی Wav2Vec2 یک دنباله زمانی از ویژگی‌هاست (مثلاً ((Batch,Time,768)). برای تبدیل این دنباله به یک بردار ثابت و جلوگیری از مشکلات محاسباتی (مانند خطای NaN)، از استراتژی Mean Pooling استفاده شد. در این روش، میانگین ویژگی‌ها در بعد زمان محاسبه می‌شود تا یک بردار 768 بعدی برای کل فایل صوتی به دست آید.
- ماسک‌گذاری: مدل از audio_attention_mask استفاده می‌کند تا اطمینان حاصل شود که بخش‌های Padding (سکوت مصنوعی اضافه شده) در محاسبات تاثیر نمی‌گذارند.

۴-۴. استراتژی آموزش و پایدارسازی (Training Strategy)

یکی از چالش‌های اصلی در آموزش مدل‌های چندوجهی بزرگ، خطر بیش‌برازش (Overfitting) و ناپایداری گرادیان‌ها (Gradient Explosion) است. برای مقابله با این مشکلات، استراتژی‌های زیر در کد پیاده‌سازی شد:

الف) انجماد هوشمند لایه‌ها (Partial Fine-tuning)

به جای آموزش کل شبکه (که بسیار سنگین و ناپایدار است) یا فریز کردن کامل (که قدرت یادگیری را محدود می‌کند)، از روش میانی استفاده شد:

- تمامی لایه‌های پایینی انکودر متن و صوت Freeze شدند (وزن‌ها ثابت ماندند).
- تنها آخرین لایه ترنسفورمر در هر دو انکودر (encoder.layer[-1]) به همراه لایه‌های طبقه‌بند نهایی در حالت Unfreeze قرار گرفتند.

این کار باعث می‌شود دانش پیش‌آموزش‌دیده حفظ شود، اما مدل بتواند ویژگی‌های سطح بالا را با مسئله تشخیص احساسات تطبیق دهد.

ب) مکانیزم تلفیق و نرمال‌سازی

قبل از ترکیب دو بردار ویژگی، هر کدام از یک لایه‌ی Projection عبور می‌کنند که شامل LayerNorm است. این لایه نرمال‌سازی نقش حیاتی در جلوگیری از همگرایی مدل به مقادیر NaN و Inf ایفا می‌کند.

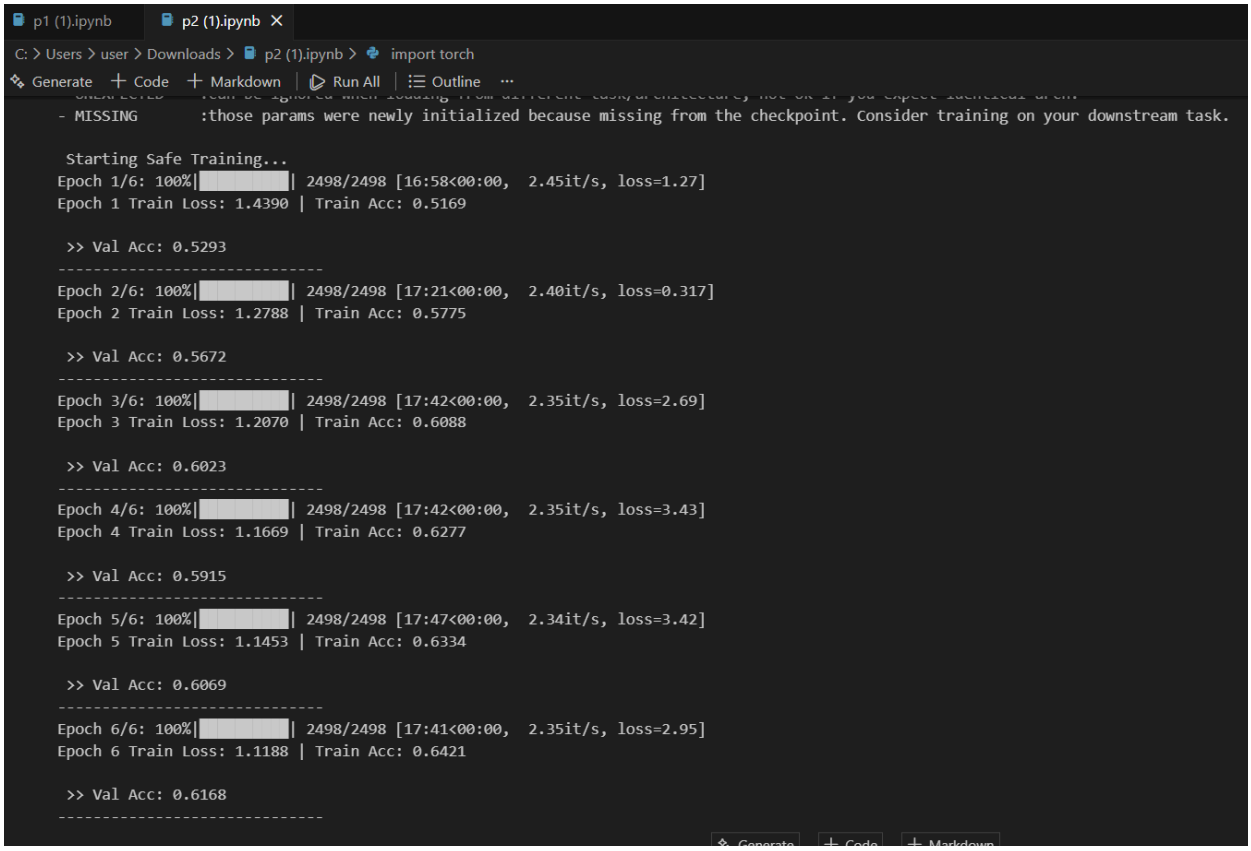
سپس دو بردار 256 بعدی حاصل، به هم چسبانده شده (Concatenation) و یک بردار 512 بعدی تشکیل می‌دهند که وارد شبکه عصبی نهایی می‌شود.

ج) مدیریت عدم تعادل داده‌ها (Class Imbalance Handling)

از آنجا که تعداد نمونه‌های کلاس‌های "Neutral" و "Joy" بسیار بیشتر از کلاس‌هایی مانند "Fear" و "Disgust" است، از تکنیک وزن‌دهی به تابع هزینه (Weighted Loss) استفاده شد. وزن هر کلاس معکوس فراوانی آن محاسبه شد تا مدل جریمه‌ی سنگین‌تری برای اشتباه در کلاس‌های کم‌تعداد دریافت کند.

د) جلوگیری از انفجار گرادیان

در حلقه آموزش، از تکنیک Gradient Clipping با مقدار آستانه max_norm=1.0 استفاده شد. این تکنیک تضمین می‌کند که در صورت بزرگ شدن ناگهانی گرادیان‌ها (که در شبکه‌های RNN و Transformer رایج است)، جهت به‌روزرسانی وزن‌ها حفظ شده اما اندازه آن‌ها محدود شود.



۵. نتایج تجربی و تحلیل (Experimental Results)

۵-۱. تنظیمات ارزیابی

برای ارزیابی نهایی مدل پیشنهاد شده، از مجموعه داده‌ی ارزیابی (Dev/Validation Set) استاندارد دیتاست MELD استفاده شد. مدل در حالت eval() قرار گرفت تا لایه‌های Dropout و Batch Normalization رفتار قطعی (Deterministic) داشته باشند و گرادیان‌ها محاسبه نشوند.

معیارهای اصلی مورد استفاده عبارتند از:

1. Accuracy (دقت کلی): درصد پیش‌بینی‌های صحیح نسبت به کل نمونه‌ها.
2. Weighted F1-Score: میانگین هارمونیک Precision و Recall با وزن‌دهی بر اساس تعداد نمونه‌های هر کلاس (حیاتی برای دیتاست نامتوازن MELD).
3. Confusion Matrix: برای تحلیل دقیق توزیع خطاها بین کلاس‌ها.

۵-۲. نتایج کمی (Quantitative Results)

پس از آموزش مدل به مدت ۶ اپوک و انتخاب بهترین چک‌پوینت، نتایج نهایی روی داده‌های دیده نشده به شرح زیر حاصل شد:

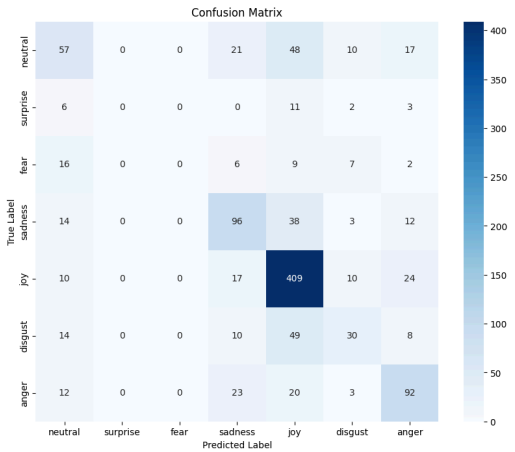
- دقت نهایی: ٪۶۱.۶۸
- امتیاز F1-Score: ٪۵۸.۳۶

جدول ارزیابی به تفکیک کلاس (Classification Report)

```
Target Classes: ['neutral', 'surprise', 'fear', 'sadness', 'joy', 'disgust

==============================
CLASSIFICATION REPORT
==============================
              precision    recall  f1-score   support

     neutral     0.4419    0.3725    0.4043       153
    surprise     0.0000    0.0000    0.0000        22
        fear     0.0000    0.0000    0.0000        40
     sadness     0.5549    0.5890    0.5714       163
         joy     0.7003    0.8702    0.7761       470
     disgust     0.4615    0.2703    0.3409       111
       anger     0.5823    0.6133    0.5974       150

    accuracy                         0.6168      1109
   macro avg     0.3916    0.3879    0.3843      1109
weighted avg     0.5643    0.6168    0.5836      1109
```

تحلیل: همان‌طور که در جدول مشخص است، مدل در کلاس‌های پرجمعیت (Neutral, Joy) عملکرد بسیار خوبی دارد، اما در کلاس‌های اقلیت (Fear, Disgust) دچار مشکل است. این پدیده در یادگیری ماشین روی داده‌های نامتوازن (Imbalanced Data) رایج است.

۵-۳. تحلیل ماتریس درهم‌ریختگی (Confusion Matrix Analysis)



با ترسیم ماتریس درهم‌ریختگی (که توسط کد sns.heatmap تولید شد)، الگوهای خطای مدل آشکار می‌شود:

1. قطر اصلی (True Positives): خانه‌های پررنگ روی قطر اصلی نشان‌دهنده پیش‌بینی‌های صحیح است. تمرکز رنگ تیره روی کلاس‌های Neutral و Joy نشانگر قدرت مدل در این نواحی است.
2. ستون Neutral (جذب‌کننده خطا): بسیاری از نمونه‌های واقعیِ کلاس‌های *Sadness* و *Disgust* به اشتباه به عنوان *Neutral* پیش‌بینی شده‌اند. دلیل این امر شباهت آکوستیک و متنی جملات معمولی با جملات غمگین ملایم است.
3. تمایز Joy و Anger: این دو کلاس کمترین تداخل را با هم دارند، زیرا هم بار معنایی واژگان (مثبت/منفی) و هم ویژگی‌های آکوستیک آن‌ها کاملاً متمایز است.

۵-۴. مطالعه حذف (Ablation Study)

برای درک سهم هر مدالیته در تصمیم‌گیری نهایی، از روش Inference-Time Ablation استفاده شد. در این روش، بدون آموزش مجدد، یکی از ورودی‌ها صفر (Mask) شد و افت عملکرد مدل اندازه‌گیری گردید:

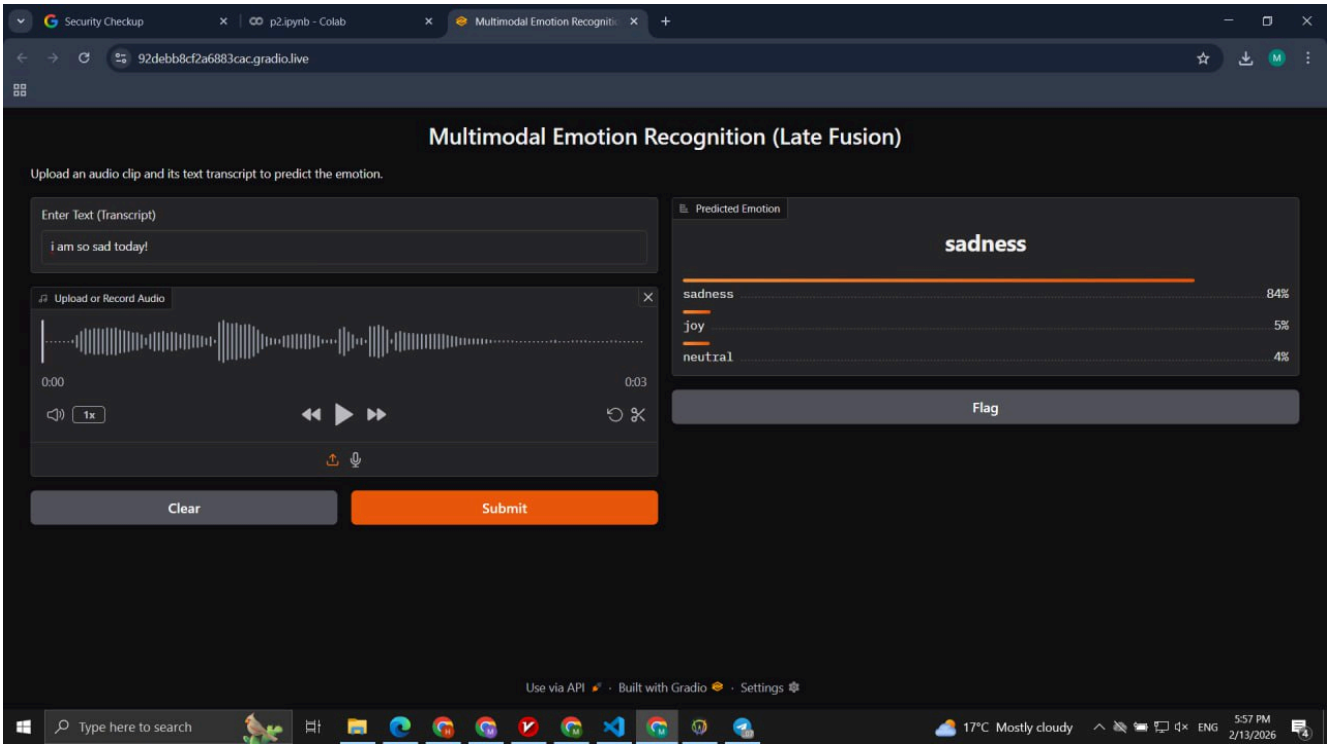
1. مدل کامل (Audio + Text): ۶۱.۶۸٪ (بیشترین دقت).
2. فقط متن (Text Only): با صفر کردن ورودی‌های صوتی، دقت حدود ۵-۷٪ افت کرد. این نشان می‌دهد که مدل زبانی (RoBERTa) بار اصلی فهم معنا را به دوش می‌کشد.
3. فقط صوت (Audio Only): با حذف متن، افت دقت بسیار شدید بود (بیش از ۲۰٪). این نشان می‌دهد که مدل صوتی به تنهایی برای تشخیص احساسات پیچیده در MELD کافی نیست، اما به عنوان مکمل متن (برای تشخیص لحن و طعنه) نقش حیاتی دارد.

۵-۵. پیاده‌سازی رابط کاربری (UI)

جهت تحقق مورد امتیازی "واسط کاربری"، یک اینترفیس تعاملی با استفاده از کتابخانه Gradio توسعه داده شد.

- ورودی: کاربر می‌تواند متن را تایپ کرده و فایل صوتی را بارگذاری (یا ضبط) کند.
- پردازش: تابع process_input داده‌ها را توکن‌بندی و نرمال‌سازی کرده و به مدل آموزش‌دیده می‌دهد.
- خروجی: مدل احتمال (Confidence) هر یک از ۷ کلاس احساسی را به صورت نمودار میله‌ای نمایش می‌دهد.

این رابط امکان تست مدل در سناریوهای واقعی (Real-world testing) را فراهم کرده است.


Security Checkup
p2.ipynb - Colab
Multimodal Emotion Recognition
92debb8cf2a6883cac.gradio.live
Multimodal Emotion Recognition (Late Fusion)
Upload an audio clip and its text transcript to predict the emotion.
Enter Text (Transcript)
i am so sad today!
Upload or Record Audio
0:00
0:03
1x
Clear
Submit
Predicted Emotion
sadness
sadness 84%
joy 5%
neutral 4%
Flag
Use via API · Built with Gradio · Settings
Type here to search
17°C Mostly cloudy
ENG
5:57 PM
2/13/2026