USC Viterbi
School of Engineering
Ming Hsieh Department of
Electrical and Computer Engineering

Multimodal Humor Detection

Peilin Li, Jiaqi Lu, Yihang Yin, Jayavibhav Niranjan Kogundi

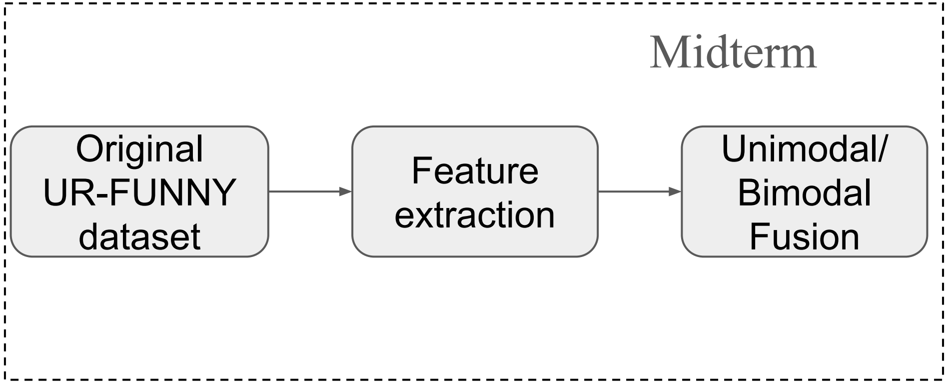
Recognizing Laughter: A Multimodal Challenge

Humor: key in human communication

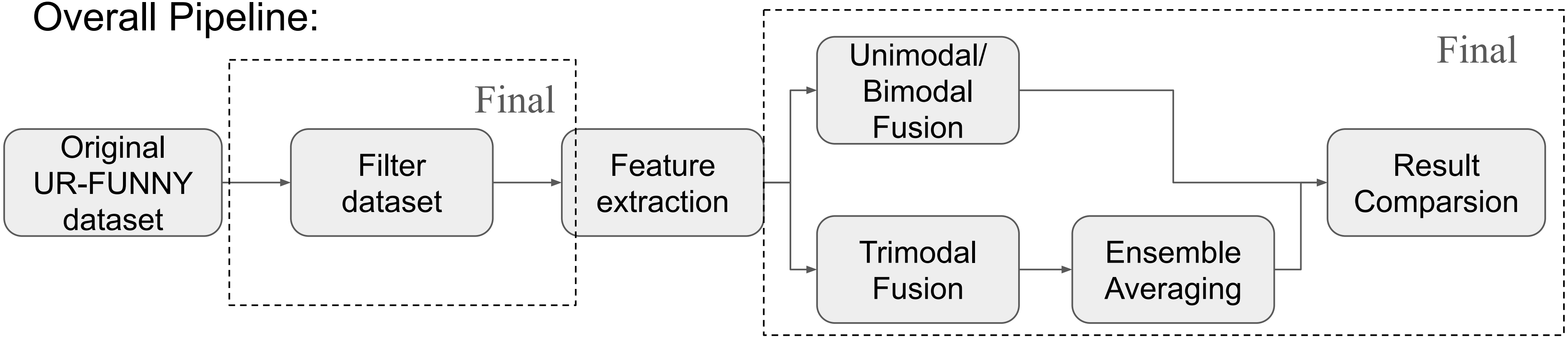
Types: sarcasm, irony, exaggeration

Multimodal deep learning approach

Detect humor from video data



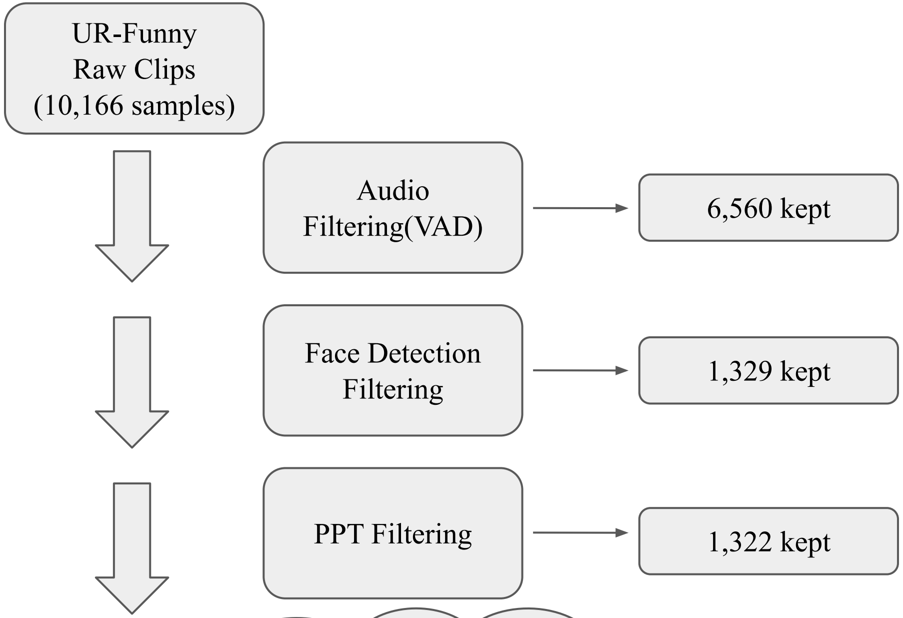
Overall Pipeline:



UR-FUNNY Dataset: Filtering and Preparation:

Data Filtering Pipeline Overview

- **Audio Speech Filtering**
  - Applied **Voice Activity Detection (VAD)** using WebRTC VAD.
  - Kept clips with **speech ratio > 85%**.
  - Batch processing enabled for 10,000+ videos.
- **Face Presence Filtering**
  - Used **OpenCV Haar Cascade** to detect faces.
  - Retained clips with **face detected in >60% of frames**.
  - Sampling 1 frame per second for efficiency.
- **PPT Frame Filtering**
  - Detected static "slide-like" frames based on **low edge density** and **low color variance**.
  - Mark frames as "PPT-like" if **edge ratio** < 2% and **color std** < 15.
  - Removed videos with **>30% PPT-like frames**.



Why PPT Filtering?

Some clips still had faces but they were frozen on slides or due to video freezing, so we applied PPT Frame Filtering to clean them up.

Unimodal Method:

Approach:

- Training multiple kinds of classifiers on each modality to try and achieve better performance.
- Tested on different sets of filtered dataset to compare results
- Text: Embedding extracted using ModernBERT, trained on both punchline only and full sentence.
- Audio: Feature (pre transformer) and last hidden states from Wav2Vec2.0
- Visual: Feature from ViT

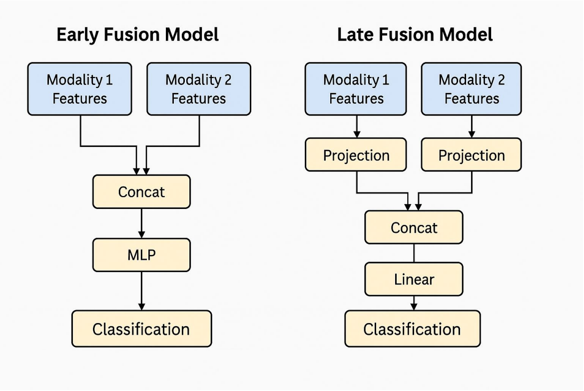
	Text (Punchline Only)		Text (Full Sentence)		Audio Features		Audio Last Hidden States		Visual	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
Unfiltered	65.44%	65.44%	66.85%	66.83%	67.97%	68.20%	71.05%	71.08%	58.05%	61.47%
Audio Filtered	64.04%	64.10%	71.80%	76.34%	67.32%	66.87%	71.05%	71.12%	51.24%	50.94%
Fully Filtered	60.65%	61.64%	71.36%	75.60%	65.16%	65.82%	68.92%	69.68%	57.29%	58.01%

Bimodal Fusion method:

Early Fusion Model: Concatenates features from both modalities, passes through a small MLP (Linear → ReLU → Linear) for final classification (input_dim → 128 → 2).

Late Fusion Model: Projects each modality separately (64-dim), then concatenates and classifies.

Modality Pair	Fusion	Accuracy (%)	AUC
Audio + Text	Early	**73.87**	0.765
Audio + Text	Late	70.85	0.755
Text + Visual	Early	66.83	0.761
Text + Visual	Late	**68.84**	0.761
Visual + Audio	Early	**55.78**	0.580
Visual + Audio	Late	54.65	0.572



Trimodal Fusion methods:

Cross-Modal Fusion:

- Load Extracted Features
- Cross-Modal Block
- Compute loss/ Update weights

Limitation:

- Only single-attention fusion
- Additive residual interaction may overlook multiplicative relations
- No robustness mechanism

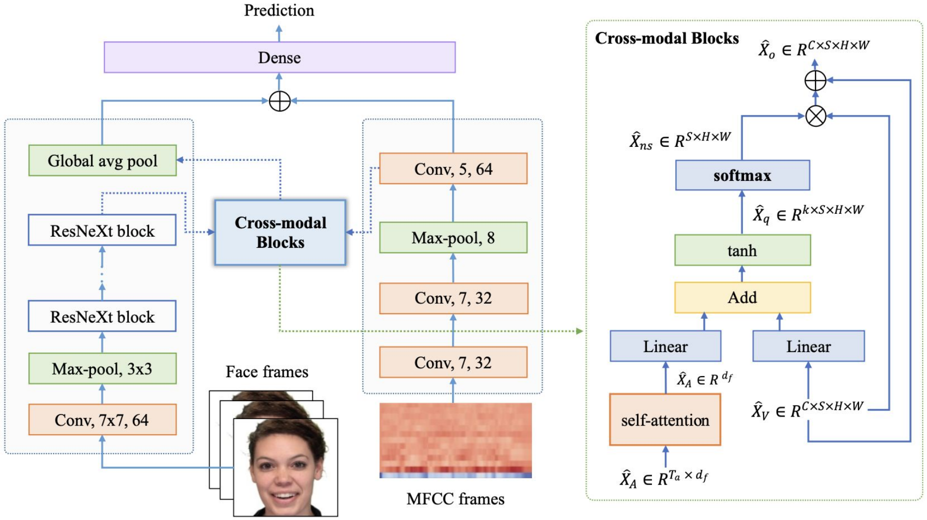


Fig. 1: The overall architecture of CFN-SR. **Left**: the flow structure of the whole model, extracting the higher-order semantic features of video and audio by ResNeXt and 1D CNN, respectively. **Right**: the cross-modal fusion blocks, which enables the complementarity and completeness of modal interactions to play a role through the introduction of self-attention mechanism and residual structure.

reference: https://arxiv.org/abs/2111.02172

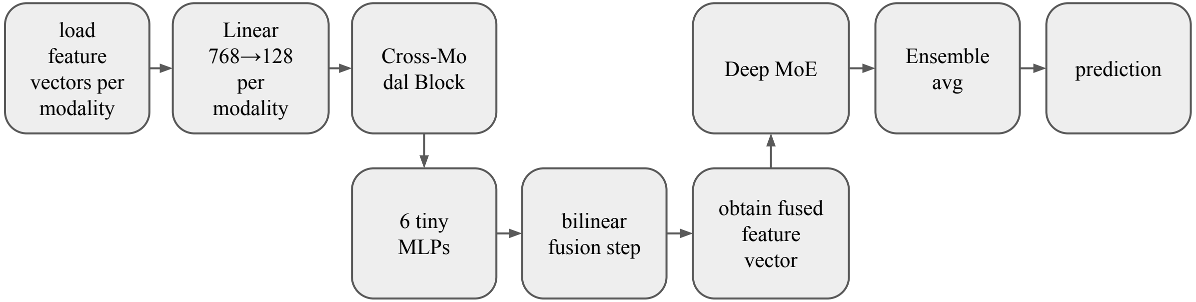
USC Viterbi
School of Engineering
Ming Hsieh Department of
Electrical and Computer Engineering

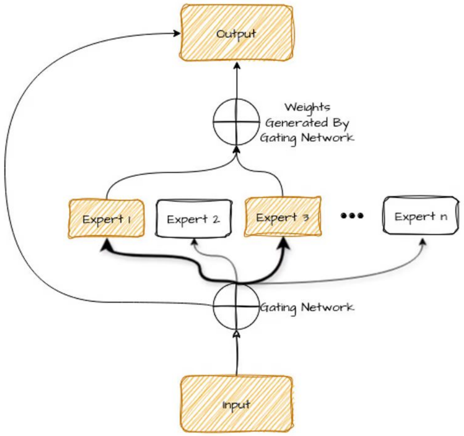
Trimodal Fusion methods:

Cross-Modal MoE++:

- Bilinear fusion
  - Six tiny MLPs to learn view-specific transforms
  - A pairwise bilinear interaction to capture multiplicative relations
  - Return a single fused feature vector for MoE
- Deep Mixture of Experts (MoE)
  - Gives the right sub-network the freedom to model each humor flavors
  - Added a ReLU layer inside of each experts
- Ensemble Averaging
  - Train five independent seeds and average their logits
  - Lowers variance and boosts generalization

```python
# --- Model Definitions ---
class CrossModalFusionPlusBoosted(nn.Module):
    def __init__(self, dim_visual, dim_audio, dim_text, output_dim=128):
        super().__init__()
        self.visual_proj = nn.Linear(dim_visual, output_dim)
        self.audio_proj = nn.Linear(dim_audio, output_dim)
        self.text_proj = nn.Linear(dim_text, output_dim)

        self.v_to_va = nn.Sequential(nn.Linear(output_dim, output_dim), nn.ReLU())
        self.a_to_va = nn.Sequential(nn.Linear(output_dim, output_dim), nn.ReLU())
        self.v_to_vt = nn.Sequential(nn.Linear(output_dim, output_dim), nn.ReLU())
        self.t_to_vt = nn.Sequential(nn.Linear(output_dim, output_dim), nn.ReLU())
        self.a_to_at = nn.Sequential(nn.Linear(output_dim, output_dim), nn.ReLU())
        self.t_to_at = nn.Sequential(nn.Linear(output_dim, output_dim), nn.ReLU())

        self.fusion_proj = nn.Linear(output_dim * 6, output_dim)

    def forward(self, v, a, t):
        v_proj = self.visual_proj(v)
        a_proj = self.audio_proj(a)
        t_proj = self.text_proj(t)

        va = self.v_to_va(v_proj) * self.a_to_va(a_proj)
        vt = self.v_to_vt(v_proj) * self.t_to_vt(t_proj)
        at = self.a_to_at(a_proj) * self.t_to_at(t_proj)

        concat = torch.cat([v_proj, a_proj, t_proj], dim=-1)
        fused = torch.cat([concat, va, vt, at], dim=-1)
        fused = self.fusion_proj(fused)
        return fused
```





reference: https://www.datacamp.com/blog/mixture-of-experts-moe

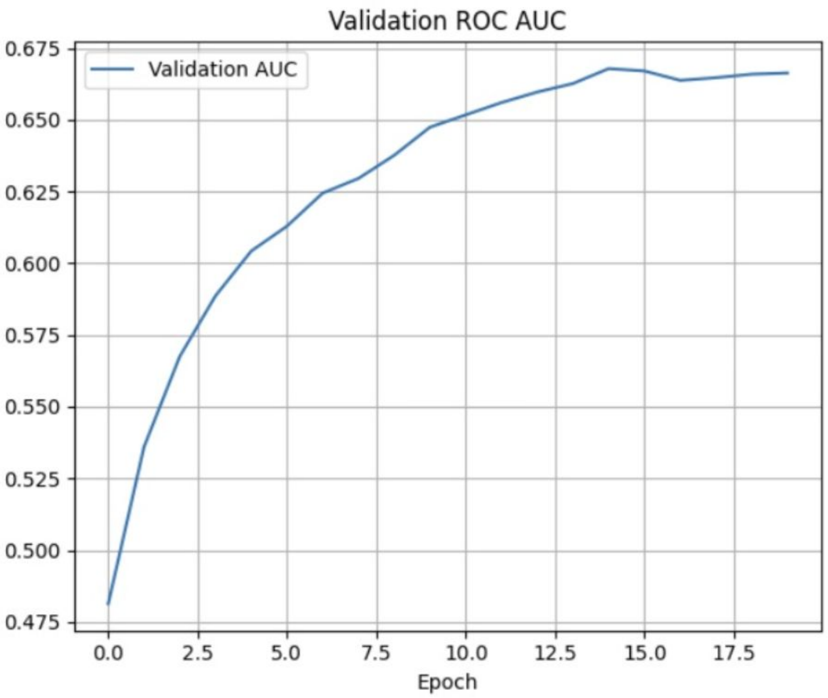
Results & Conclusion:

Trimodal fusion results:

- Validation ROC AUC steadily increased over epochs, indicating effective learning and convergence.

Conclusion:

- Trimodal fusion outperformed unimodal and bimodal methods
- Careful fusion design is more important than stronger single features



Trimodal Fusion results:
Test accuracy: 0.7437
Test ROC AUC: 0.7737

USC Viterbi School of Engineering
Ming Hsieh Department of Electrical and Computer Engineering

Thank you!

Team Contribution:
Peilin Li: Designed and implemented the Cross-Modal MoE++ model. Explored on the visual single modality fusion approaches.
Jiaqi Lu: Unimodal/Bimodal design and training, Feature Extraction
Yihang Yin: Data filtering pipeline design and implementation. Contributed to the design of the trimodal fusion strategy, including investigating Cross-Modal Fusion limitations,modifying the cross-modal block. Also explored the feasibility of using Deep Mixture of Experts.
Jayavibhav Niranjan Kogundi: Unimodal (Text, Visual) and Bimodal design and training