Cross-Modal Video Anomaly Detection: Visual-Audio Alignment via Optical Flow and Fourier Analysis

Matt Wang, Lily Ru, Anna Dai

Johns Hopkins University

May 14, 2025

Overview

Video Anomaly Detection: The Cross-Modal Challenge

Problem Statement:

- Traditional video anomaly detection focuses primarily on visual content
- Our focus: Detecting *misalignment* between audio and visual streams
- Applications: Deepfake detection, media authentication, quality assurance

Types of Audio-Visual Anomalies:

- **Temporal misalignment:** Audio delay/advance
- **Audio corruption:** Muting, noise, distortion
- **Semantic Mismatch:** Incongruent audio and visual content

Our Approach to Audio-Visual Synchronization Detection

Key Challenges:

- Different sampling rates of modalities
- Feature representation discrepancy
- Limited labeled data for anomalies

Project Goals:

- Build feature extraction pipelines for both audio and video
- Generate audio-visual misalignment for training and testing
- Evaluate detection methods:
  - Thresholding
  - MLP classifier

Feature Representations

- **Audio Features (Mel-Frequency Cepstral Coefficients - MFCC):**
  - Uses Short-Time Fourier Transform (STFT) for time-frequency analysis
  - Applies perceptually motivated Mel-scale filter banks
  - Output: Vector for compact spectral representation per time window

- **Visual Features (Optical Flow):**
  - Computes motion between consecutive frames
  - Captures magnitude of movement
  - Output: Concatenated and flattened flow vectors

Dataset: Audio-Visual Event (AVE)

AVE Dataset Overview:

- 4,143 videos covering 28 audio-visual event categories
- Annotations for usable clips are saved in TXT file with complex delimiters: "Category&VideoID&Quality&StartTime&EndTime"

Data Preprocessing: Annotation Conversion

- **Original format in txt:** "Category&VideoID&Quality&StartTime&EndTime"
- **Conversion to CSV:**
  - Parsed annotations into structured format
  - Added derived fields (e.g., Duration)
  - Improved accessibility for downstream processing
- **Result:** `ave_annotations_preprocessed.csv`

Data Preprocessing: Synthetic Anomaly Generation

Challenge: Limited anomalous examples in original dataset $\Rightarrow$ generate synthetic misalignments by randomizing segment goodness.

Types of Synthetic Anomalies:

- *Time shift:* Audio delay relative to video
- *Noise:* White noise replacing audio
- *Mute:* Complete audio removal
- *Distort:* Sinusoidal temporal warp to the video playback speed

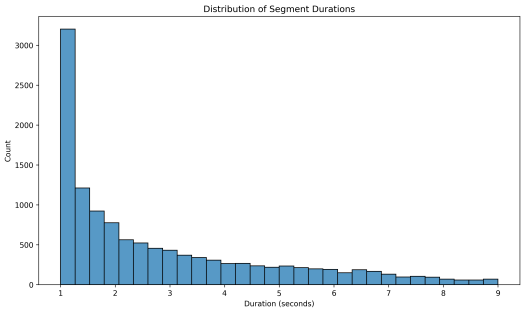
Result Summary:

- Total segments: 12,114
- Aligned: 6,049 (49.9%)
- Misaligned: 6,065 (50.1%)

Misalignment Breakdown:

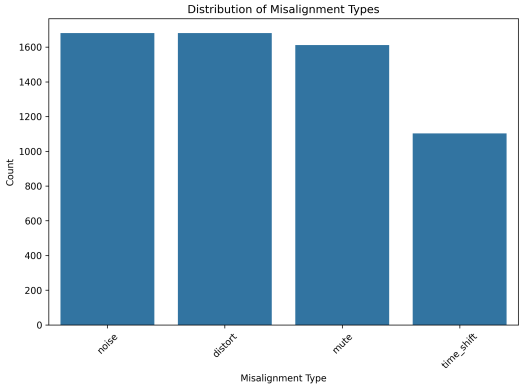
- Noise: 1,669 (27.5%)
- Mute: 2,061 (34.0%)
- Distort: 1,613 (26.6%)
- Time Shift: 722 (11.9%)

Data Preprocessing: Synthetic Data Statistics



Duration Distribution:

Most segments are short (1–2s), where long-tail durations add diversity.



Misalignment Distribution:

Synthetic misalignments are almost evenly distributed across types, ensuring balanced training examples.

Feature Extraction Pipeline

Visual Features:

- Grayscale frame extraction at 5 FPS
- Resize to 96×96 pixels
- Compute optical flow between frames
- Concatenate and flatten to feature vector

Audio Features:

- MFCC extraction (13 coefficients)
- 16kHz sampling rate
- Transpose to align with visual frames

Feature Processing & Storage

- **Temporal Alignment:**
  - Truncate audio and visual features to match in length
  - Ensures synchronized input for model
- **Label Extraction:**
  - Labels parsed from filename patterns (e.g., "misaligned", "aligned")
- **Storage Format:**
  - Store processed features and labels in `.npz` format
  - Enables fast loading and compact storage

Method 1: Thresholding

Key Idea: After extracting features, we detect misalignment according to the audio and visual features generated, by applying an "optimal" threshold to classify normal/anomaly.

Steps:

1. **Data Preparation**
   - Load visual and audio arrays
   - Split data into training (0.7) and testing (0.3) sets
2. **Compute Alignment Scores**
   - The alignment score is computed as the **normalized Euclidean distance** between audio (a) and visual (v) feature vectors:

$$\text{score} = \frac{\|a - v\|_2}{\sqrt{\text{min\_len}}}$$

   - Low scores mean more aligned, while high scores mean less aligned (anomaly)

Method 1: Thresholding

Steps (Continued):

3. **Determine Optimal Threshold (via ROC Analysis):**
   - Sweep over a range of threshold values to classify alignment scores as normal or anomalous.
   - Select the threshold that maximizes Youden's J statistic (i.e., TPR − FPR), achieving the best trade-off between true positive and false positive rates.
4. **Evaluate on Test Set:**
   - Apply the selected threshold to alignment scores computed on the test set.
   - Label a sample as anomalous if its score exceeds the threshold.
   - Compute evaluation metrics such as precision, recall, F1-score, etc.

Model Performance Evaluation (Thresholding)

Metric	Score
Accuracy	0.6271
Precision	0.6440
Recall	0.9361
F1 Score	0.7631

- Moderate Accuracy of 0.6271
- High recall indicates good sensitivity
- F1 (Harmonic mean of precision and recall) shows reasonable balance

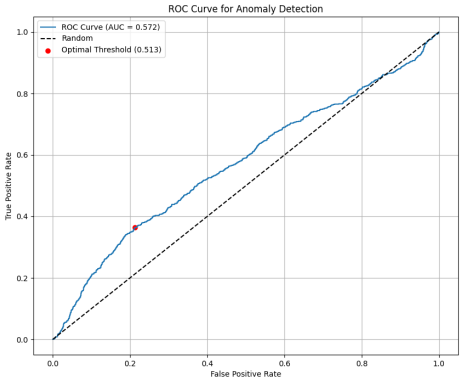


Figure: ROC curve showing threshold performance

Neural Network Method: Model Architecture

Model Type: Enhanced Multi-Layer Perceptron (MLP)

Architecture:

- **Input Layer:** Linear projection to 512-dim with BatchNorm, Swish activation, and Dropout (0.3).
- **Intermediate Layers:**
  - Residual blocks with linear transforms, BatchNorm, and Swish activation.
  - Layer dimensions: [512, 256, 64]
- **Output Layer:** Linear(64, 1) without activation (logits for `BCEWithLogitsLoss`).

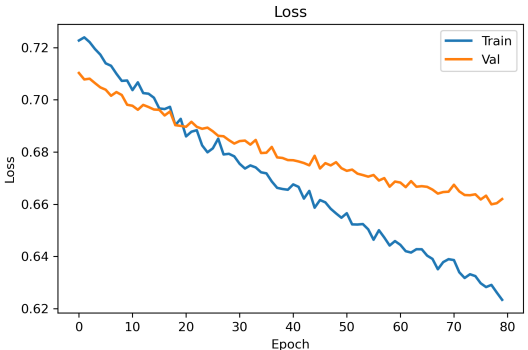
Neural Network Method: Training and Evaluation

Training Setup:

- **Loss Function:** Binary Cross Entropy with Logits
- **Optimizer:** Adam (LR = 1e-6, No Weight Decay)
- **Epochs:** 50, **Batch Size:** 128

Evaluation Metrics:

- Accuracy, F1-score



Concluding Remarks

- **Practical Impact**: The model demonstrates real-world applicability with strong recall, particularly valuable for audio-visual misalignment detection. We also produce a valuable dataset for future research.
- **Result Analysis**: MLP performs slightly better, but thresholding requires a lot less computational resources and complexity.
- **Promising Directions for Future Improvement:**
  - Feature engineering to boost precision
  - Optimization via alternative architectures and Hyperparameter Tuning
  - Exploration of alternative thresholding selection and neural network methods

Thank You!