Multimodal Deepfake Detection

Nick Cheliotis
MSc in AI

July 3, 2025

Abstract

Deepfake technologies have become increasingly sophisticated, posing serious threats to information integrity, online trust, and digital security. In this project, we propose a multimodal approach to deep-fake detection by combining both visual and audio features. Visual features are extracted using MediaPipe to capture facial landmarks and compute metrics such as blink rate and variance in head motion. Audio features are derived from prebuilt libraries. We train traditional machine learning models, including logistic regression, on the combined features to classify videos as real or fake. Our results demonstrate that fusing modalities significantly improves detection performance, with a final accuracy of 75% in our evaluation data set. This work highlights the effectiveness of low-cost interpretable feature engineering techniques in combating fake content.

1 Introduction

The rise of deep-fake technology - synthetic media generated using AI techniques such as GANs - has sparked both fascination and concern. While it enables creative possibilities in film, gaming, and accessibility, it also introduces critical ethical and security challenges. Deepfakes can be used to spread misinformation, commit fraud, or damage reputations by impersonating real individuals with alarming realism.

Despite a growing number of research efforts in deepfake detection, many rely heavily on large public datasets or deep neural networks with limited interpretability. In this project, we take a more grounded approach: rather than relying on standard data or black-box deep learning models, we design and build our entire pipeline from scratch.

A key contribution of this work is the creation of a custom dataset consisting of real and deep-fake video samples, carefully collected and labeled to fit our experimental needs. This hands-on dataset construction allows us full control over the modalities and formats involved.

From there, we focus on interpretable, hand-crafted feature engineering, a deliberate contrast to opaque deep models. Visual features are extracted using MediaPipe, a library that identifies facial landmarks in each frame. From these landmarks, we compute domain-relevant features such as blink rate and head motion variance, hypothesizing that subtle inconsistencies in eye movement and stability might betray synthetic content.On the audio side, we utilize the Librosa library to extract statistical features from the speech signal.

By training traditional machine learning models (e.g. logistic regression) on these multimodal features, we evaluate the effectiveness of our approach. The goal is not just to detect deep-fakes, but to do so in a way that is transparent, reproducible, and interpretable - laying the groundwork for low-resource, high-trust detection systems that could one day be deployed in real-world settings.

2 Dataset Generation

2.1 Video collection

Given the lack of existing datasets tailored to our multimodal deepfake detection goals, we manually constructed a custom dataset by collecting short video clips from YouTube Shorts. To ensure consistency and reduce noise in the data, all selected videos satisfied the following criteria:

- Exactly one person is present in the video.
- No facial obstructions (e.g., hands, objects, hair).
- No sudden camera angle changes or scene cuts.
- No background music or added sound effects.
- Minimum duration of 10 seconds of uninterrupted footage.

Using these constraints, we curated a dataset of 100 high-quality videos. Out of these:

- 50 videos were retained in their original form and labeled as **real**.
- The remaining 50 were synthetically manipulated and labeled as **deepfake**.

The deepfake videos were further categorized into three types:

- **20 videos** were modified in both **audio and video** to create fully multimodal deepfakes.
- **15 videos** were altered only in the **video** modality, preserving the original audio.
- **15 videos** were altered only in the **audio** modality, preserving the original video.

This controlled setup allowed us to test not only general deepfake detection performance, but also the individual and combined impact of visual and auditory manipulations.

2.2 Fake Audio Generation and Lip Synchronization

To generate realistic and convincing deepfake audio, we first extracted transcripts from each video using OpenAI's Whisper ASR model. Whisper provided accurate and language-agnostic transcriptions, enabling us to preserve the original speech content while replacing the speaker's voice.

Next, we used the ElevenLabs speech synthesis platform to generate fake voices for each transcript. Multiple synthetic voice profiles were utilized to introduce variation in speaker tone, gender, and emotional delivery — increasing the realism and diversity of the manipulated samples.

After generating the fake audio, we applied **Wav2Lip**, a state-of-the-art lip synchronization model, to align the synthesized speech with the speaker's lip movements in the original video. This step was critical in maintaining audiovisual consistency and avoiding obvious mismatches between mouth motion and spoken content.

The result was a set of highly realistic deepfake videos where only the audio modality was altered, as well as samples where both audio and video were manipulated in conjunction.

2.3 Visual Deepfake Generation

To generate the manipulated video modality for our deepfake samples, we employed advanced AI-based face-swapping techniques. Specifically, we utilized state-of-the-art pre-trained face replacement models available through online deepfake toolkits. These models leverage generative adversarial networks (GANs) and facial identity mapping to create photorealistic face swaps while preserving head pose, lighting, and expression consistency.

This method allowed us to create two sets of visual deepfakes: one where only the face was swapped (video-only deepfake), and one where the face swap was combined with altered audio using lip-sync models, resulting in a fully multimodal deepfake.

3 Feature engineering

3.1 Visual Features

To capture facial dynamics and identify inconsistencies typical of deepfakes, we extracted several interpretable visual features using the MediaPipe Face Mesh model, which detects 468 facial landmarks per frame. These landmarks allowed us to compute high-level behavioral signals that are difficult for synthetic models to replicate. The extracted features include blink rate, mouth open ratio, head motion variance, and expression entropy.

Blink Rate

Eye blinking is a spontaneous, unconscious behavior that deepfake models often fail to replicate accurately, especially in older or lower-quality generations. To measure blink rate, we used eye aspect ratio (EAR), which captures the relative distance between vertical and horizontal eye landmarks.

For each frame, we computed EAR as:

$$EAR = \frac{||p_2 - p_6|| + ||p_3 - p_5||}{2 \cdot ||p_1 - p_4||}$$

where p_i are specific landmark points around the eye. A blink is detected when the EAR drops below a predefined threshold (typically 0.2) for a short sequence of frames. The final blink rate is computed as:

$$\text{Blink Rate} = \frac{\text{Number of Blinks}}{\text{Video Duration (in seconds)}}$$

Mouth Open Ratio

Unnatural mouth movements may indicate poor lip-sync or facial reenactment artifacts. To quantify this, we computed the mouth open ratio based on the vertical and horizontal distances between lip landmarks.

$$\text{Mouth Ratio} = \frac{||p_{\text{upper}} - p_{\text{lower}}||}{||p_{\text{left}} - p_{\text{right}}||}$$

where p_{upper} and p_{lower} are top and bottom lip landmarks, and p_{left}, p_{right} are the mouth corners.

Head Motion Variance

Deepfake faces are often less stable and may exhibit unnatural jitter or lack organic micro-expressions. To capture this, we modeled the face in pseudo-3D space using landmark-based pose estimation and computed head motion variance in yaw and pitch across the video.

For each frame, we estimated the yaw (θ_y) and pitch (θ_p) angles. Then, we calculated the variance over all frames:

$$\text{Yaw Variance} = \text{Var}(\theta_y), \quad \text{Pitch Variance} = \text{Var}(\theta_p)$$

High or extremely low variance values can both indicate manipulation, depending on the type of forgery.

Expression Entropy

To capture facial expression dynamics, we used a pre-trained DeepLab model that classifies facial expressions per frame (e.g., happy, sad, angry, neutral). From this sequence, we computed the Shannon entropy:

$$H = -\sum_{i=1}^{N} p_i \log p_i$$

where p_i is the normalized frequency of expression class i across the video. Deepfakes may exhibit unnaturally static expressions, leading to lower entropy.

This suite of features allowed us to represent fine-grained behavioral cues, essential for detecting manipulated visual content.

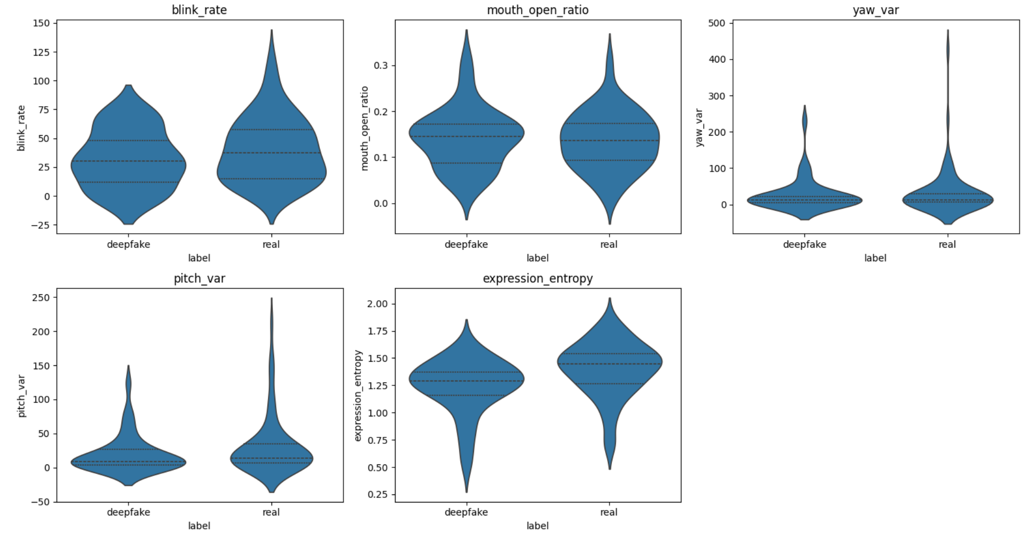

Figure 1: Violin plots of visual features extracted from real and deepfake videos. The plots reveal characteristic distributional differences between modalities.

3.2 Audio Features

To complement the visual cues, we extracted a series of interpretable audio features from the speech track of each video using the `Librosa` library. These features were chosen to capture prosodic and statistical patterns in vocal delivery that may distinguish real from synthetic speech.

Pitch Statistics (Mean and Standard Deviation)

Pitch, or fundamental frequency, is a key characteristic of human speech. Synthetic voices often struggle to replicate natural variation in pitch over time. Using Librosa's `piptrack` function, we extracted the pitch contour and computed the mean and standard deviation of pitch per audio clip:

$$\text{Pitch}_\mu = \frac{1}{N}\sum_{i=1}^{N} f_i, \quad \text{Pitch}_\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(f_i - \text{Pitch}_\mu)^2}$$

where f_i is the pitch frequency at frame i, and N is the number of frames.

Energy Statistics and Entropy

Energy reflects the loudness and dynamics of speech. Flat or overly uniform energy is a potential indicator of synthetic generation. We extracted the root mean square (RMS) energy for each frame:

$$\text{Energy}_\mu = \frac{1}{N}\sum_{i=1}^{N} E_i$$

In addition, we calculated the Shannon entropy of the energy distribution to capture variability and expressiveness:

$$H_E = -\sum_{j=1}^{B} p_j \log p_j$$

where p_j is the normalized histogram of energy over B bins.

Mel-Frequency Cepstral Coefficients (MFCCs)

MFCCs are widely used to represent the spectral properties of speech and are sensitive to timbre and vocal tract shape. We extracted the first three MFCCs per frame and computed their mean across time. However, analysis revealed that only the first MFCC exhibited noticeable variation between real and fake speech, and was thus retained in the final feature set.

Excluded Features

We initially considered additional features such as silence ratio, zero crossing rate, and higher-order MFCCs. However, distributional analysis via violin plots showed no significant distinction between real and fake samples for these features. As a result, they were excluded to reduce dimensionality and avoid introducing noise into the classification process.

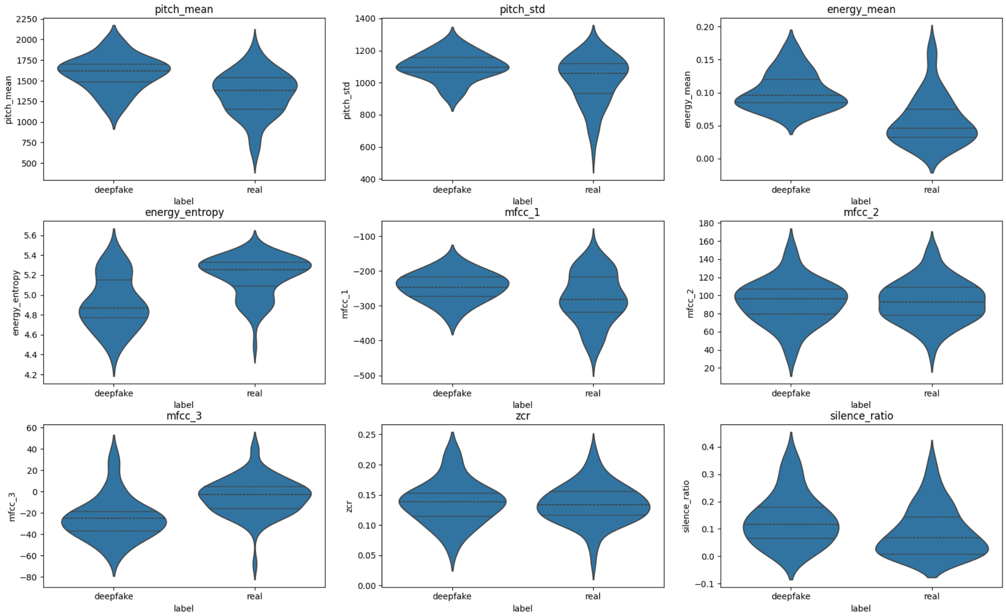

Figure 2: Violin plots of features for real and deepfake samples.

4 Experiments

We evaluated the effectiveness of our extracted features using two widely-used, interpretable classification models: Support Vector Machines (SVM) and Logistic Regression. Both models were chosen for their robustness, simplicity, and ability to perform well on relatively low-dimensional, engineered feature sets.

To optimize model performance, we performed hyperparameter tuning using a grid search strategy. Specifically, we used `GridSearchCV` with 5-fold cross-validation to identify the best combination of hyperparameters for each model. The search space included:

- **Logistic Regression:** Regularization strength (`C`), solver type
- **SVM:** Kernel type (linear, RBF), regularization parameter (`C`), and gamma

After identifying the optimal parameters, each model was retrained on the full training set using the selected configuration. We then evaluated their performance on a held-out test set using standard classification metrics, including accuracy, precision, recall, and F1-score.

This experimental setup allowed us to assess how well simple classifiers could differentiate between real and deepfake content using only interpretable multimodal features.

5 Results

Both models — Logistic Regression and SVM — achieved comparable performance in detecting deepfakes using the engineered multimodal features. Below, we report detailed classification metrics, confusion matrices, and performance across different deepfake types.

Classification Performance

Table 1 presents the precision, recall, and F1-score for both models on the test set.

Model	Class	Precision	Recall	F1-score
Logistic Regression	Real	0.80	0.73	0.76
	Deepfake	0.70	0.78	0.74
SVM	Real	0.88	0.64	0.74
	Deepfake	0.67	0.89	0.76

Table 1: Precision, recall, and F1-scores for Logistic Regression and SVM models. Both achieve similar macro-average F1 around 0.75.

Both models achieved an overall accuracy of 75%, indicating that even simple classifiers can perform reasonably well with carefully engineered features.

Confusion Matrices

Figures 3 and 4 show the confusion matrices for Logistic Regression and SVM, respectively.

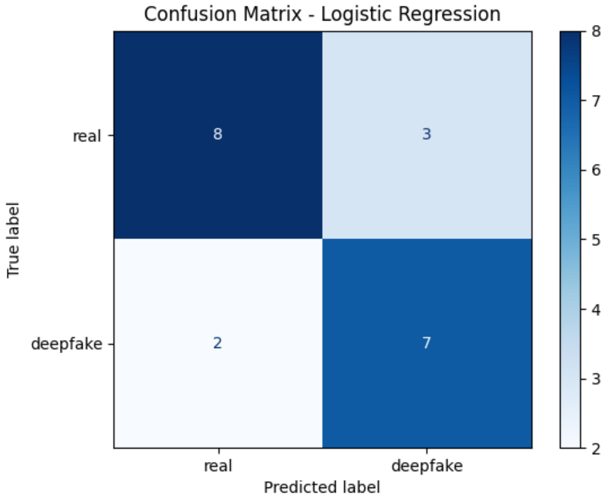

Figure 3: Confusion matrix for Logistic Regression. The model misclassified 3 real and 2 fake samples.

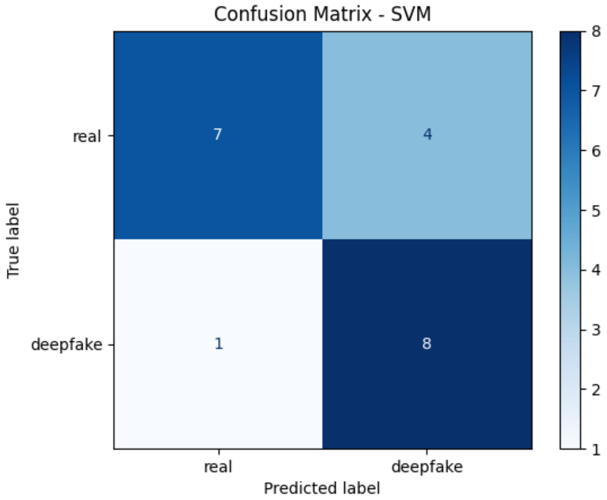

Figure 4: Confusion matrix for SVM. It showed higher recall for fake samples, misclassifying only 1 deepfake but 4 real videos.

Performance by Deepfake Type

To better understand model behavior across manipulation types, we evaluated accuracy separately for each fake type: audio-only, video-only, and audio+video.

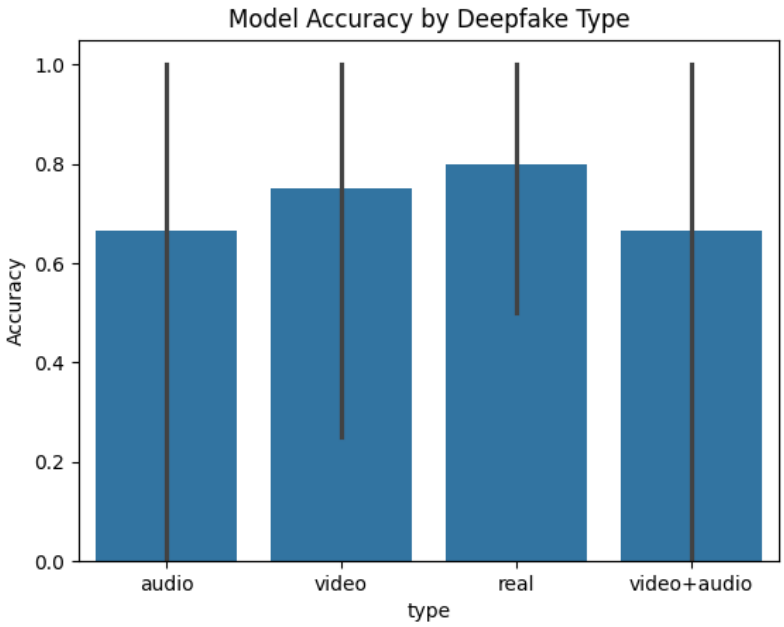

Figure 5: Logistic Regression accuracy across different deepfake types. The model performed best on real and video-only samples, while struggling with audio and fully multimodal fakes.

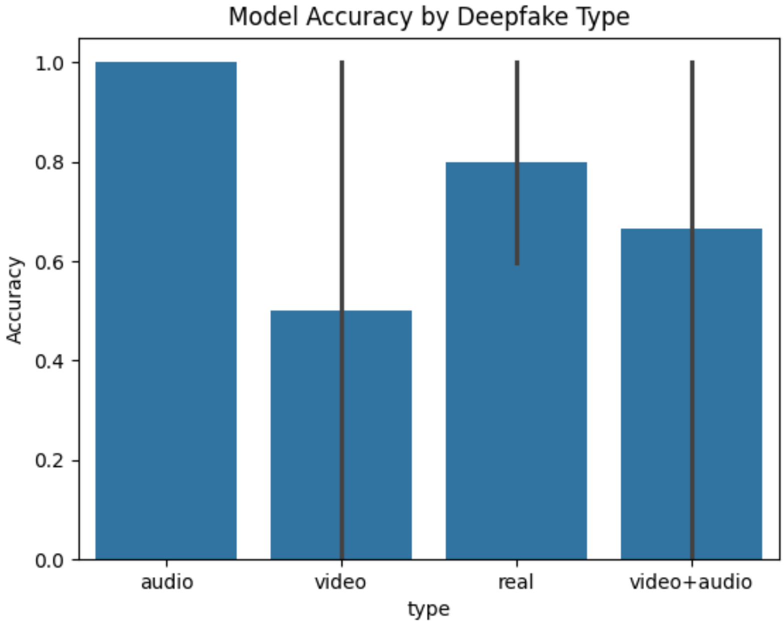

Figure 6: SVM accuracy per deepfake type. The model excelled at detecting audio-only fakes but underperformed on video-only manipulations.

These plots highlight that audio-only fakes are easier to detect than fully multimodal deepfakes. This suggests that the audio features, particularly pitch and energy entropy, are strong indicators, while combined manipulations introduce more noise and uncertainty.

6 Conclusion

In this project, we presented a multimodal deepfake detection pipeline built entirely from scratch — including custom dataset creation, feature engineering, and traditional machine learning modeling. By combining interpretable audio and visual features, we demonstrated that even relatively simple classifiers like Logistic Regression and SVM can achieve solid performance in detecting deepfakes.

Our results showed that certain features, such as expression entropy, pitch entropy, and energy mean, were particularly useful in differentiating real from fake content. Both models reached an overall accuracy of 75%, with performance varying across deepfake types. Audio-only deepfakes were the easiest to detect, while fully multimodal manipulations posed the greatest challenge.

Overall, our work highlights that meaningful, hand-crafted features — when carefully chosen — still have significant value in the deepfake detection landscape.

References

[1] MediaPipe by Google: Face Mesh Model. https://github.com/google-ai-edge/mediapipe

[2] Librosa: Python Audio Analysis Library. https://librosa.org/

[3] OpenAI Whisper: Automatic Speech Recognition. https://github.com/openai/whisper

[4] ElevenLabs Text-to-Speech API. https://www.elevenlabs.io/

[5] Wav2Lip: Accurate Lip-sync for Audio-Video. https://github.com/Rudrabha/Wav2Lip