ACTIVE SPEAKER DETECTION

“HOW TECHNOLOGY HEARS THE LOUDEST VOICE”

ALEESHA AHMED

CS-19013

CGPA:3.47

KASHAF KHAN

CS-19002

CGPA:3.80

HAFSA ZAFAR

CS-19012

CGPA:3.87

GROUP MEMBERS

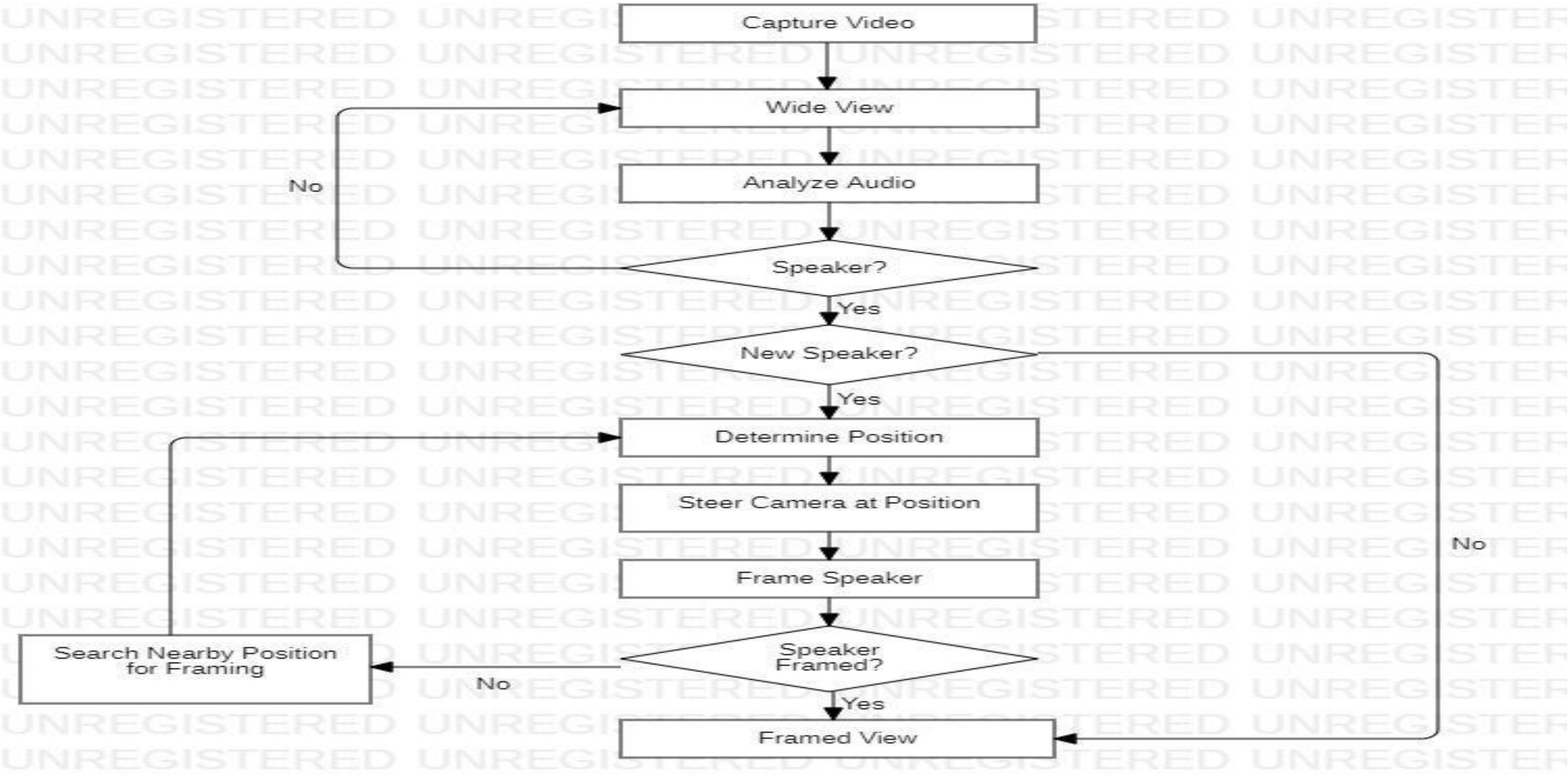
PROJECT INTRODUCTION

- Welcome to our presentation on an innovative system for active speaker detection in challenging environments.
- In scenarios where multiple individuals speak and move, our system offers a robust solution by leveraging both audio and visual cues.
- Unlike conventional methods relying solely on audio data, our approach combines the power of audio-visual information from multiple sensors.
- Our integrated system combines audio and visual modules, utilizing labeled facial features for video and enhancing audio accuracy with labeled data.

THE POWER OF AUDIO-VISUAL FUSION

- **Comprehensive Perception:** Integrating audio and visual inputs provides a more holistic understanding of the environment.
- **Enhanced Robustness:** Multiple modalities ensure greater accuracy, especially in cluttered environments with multiple speakers.
- **Synergistic Insights**: Audio and visual cues complement each other, enriching the feature set for more accurate identification.
- **Adaptive Learning:** Our system learns from both auditory and visual patterns, adapting to variations in speech and movement.

FLOW OF THE PROJECT



GANTT CHART OF THE PROJECT

	7TH SEMESTER				BREAK		8TH SEMESTER			
PROJECT PLAN	1ST MONTH	2ND MONTH	3RD MONTH	4th MONTH	5TH MONTH	6TH MONTH	7TH MONTH	8TH MONTH	9TH MONTH	10th MONTH
LITERATURE REVIEW										
SELECTION OF ALGORITHMS										
DATA PREPROCESSING										
IMPLEMENTATION OF AUDIO ALGORITHM										
TESTING OF AUDIO ALGORITHM										
IMPLEMENTATION VIDEO ALGORITHM										
TESTING OF VIDEO ALGORITHM										
INTEGRATION										
TESTING OF PROJECT										
DOCUMENTATION										

OVERALL PROJECT WORK

AUDIO MODEL IMPLEMENTATION

DATA PREPROCESSING

FEATURE EXTRACTION

CONVOLUTIONAL NEURAL NETWORK

PREDICTED RESULTS

SPEAKER MODEL IMPLEMENTATION

VOICE ACTIVITY DETECTION

CLUSTERING

GAUSSIAN MODEL

SPEAKER DIARIIZATION

OVERALL PROJECT WORK

VIDEO MODEL IMPLEMENTATION

- VIDEO PROCESSING
- FACE DETECTION
- FACIAL FEATURES EXTRACTION
- RETINA FACE IMPLEMENTATION
- FRAMING
- RESULTS

PROJECT INTEGRATION

- AUDIO MODELS INTEGRATION
- AUDIO VISUAL FUSION

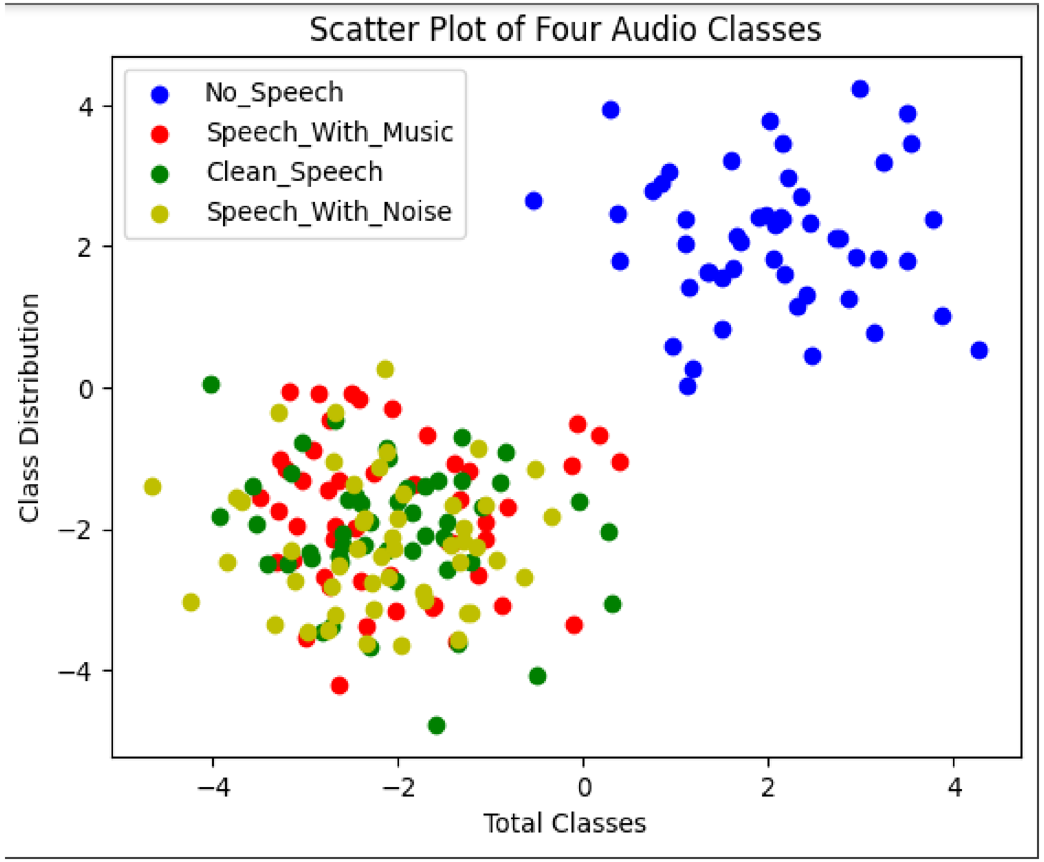
MID YEAR OVERVIEW

- Audio Data Preprocessing
- Audio Extraction From Dataset
- Feature Etraction from the Audio
- Video Model Implementation

	YouTube Identifier	label_start_timestamp_seconds	label_end_timestamp_seconds	Speech
0	JNb4nWexD0I	900.00	901.15	NO_SPEECH
1	JNb4nWexD0I	901.15	902.20	CLEAN_SPEECH
2	JNb4nWexD0I	902.20	902.66	SPEECH_WITH_NOISE
3	JNb4nWexD0I	902.66	904.79	NO_SPEECH
4	JNb4nWexD0I	904.79	905.40	CLEAN_SPEECH
...	...	...	...	...
7437	2fwni_Kjf2M	1780.59	1789.95	SPEECH_WITH_NOISE
7438	2fwni_Kjf2M	1789.95	1791.27	NO_SPEECH
7439	2fwni_Kjf2M	1791.27	1795.23	SPEECH_WITH_NOISE
7440	2fwni_Kjf2M	1795.23	1796.31	NO_SPEECH
7441	2fwni_Kjf2M	1796.31	1800.00	SPEECH_WITH_NOISE

7442 rows × 4 columns

AUDIO DATASET



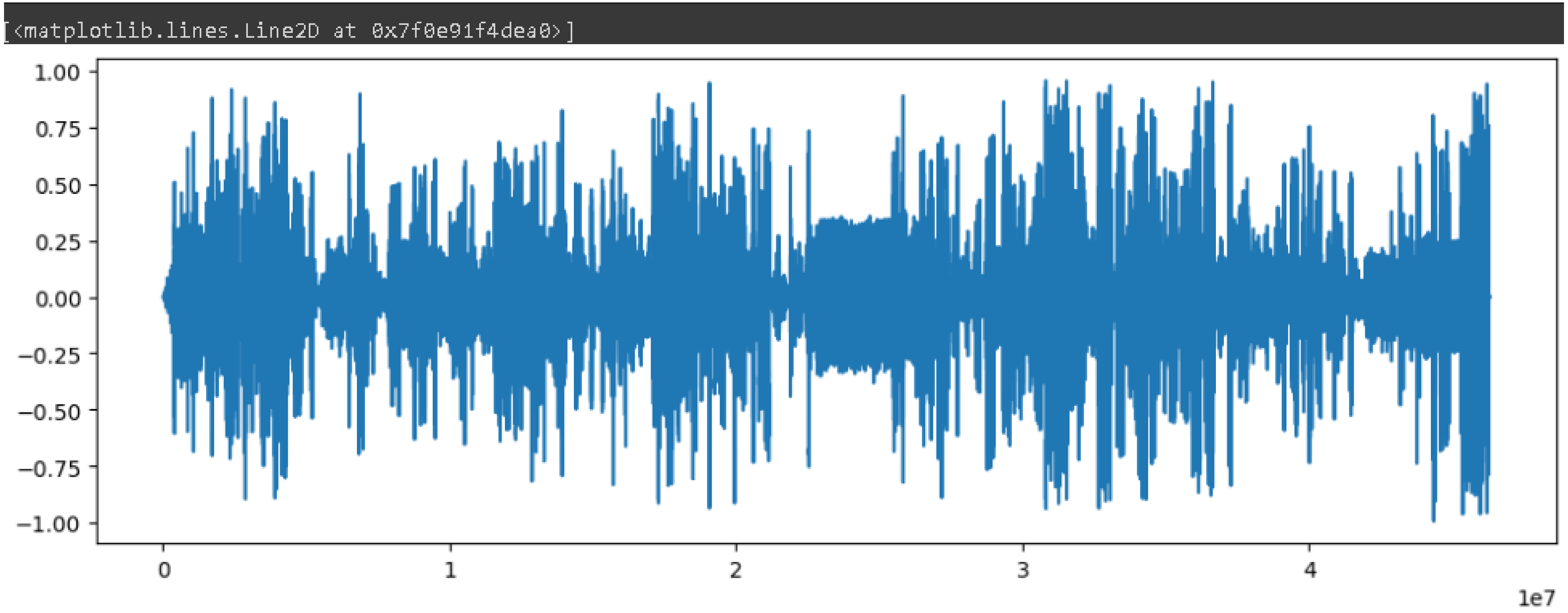
No Speech:
Segments with no apparent human speech, encompassing ambient sounds, silence, or absence of vocal content.

Speech with Music:
Audio samples merging human speech with musical elements, seen in songs, podcasts, or presentations with combined speech and music.

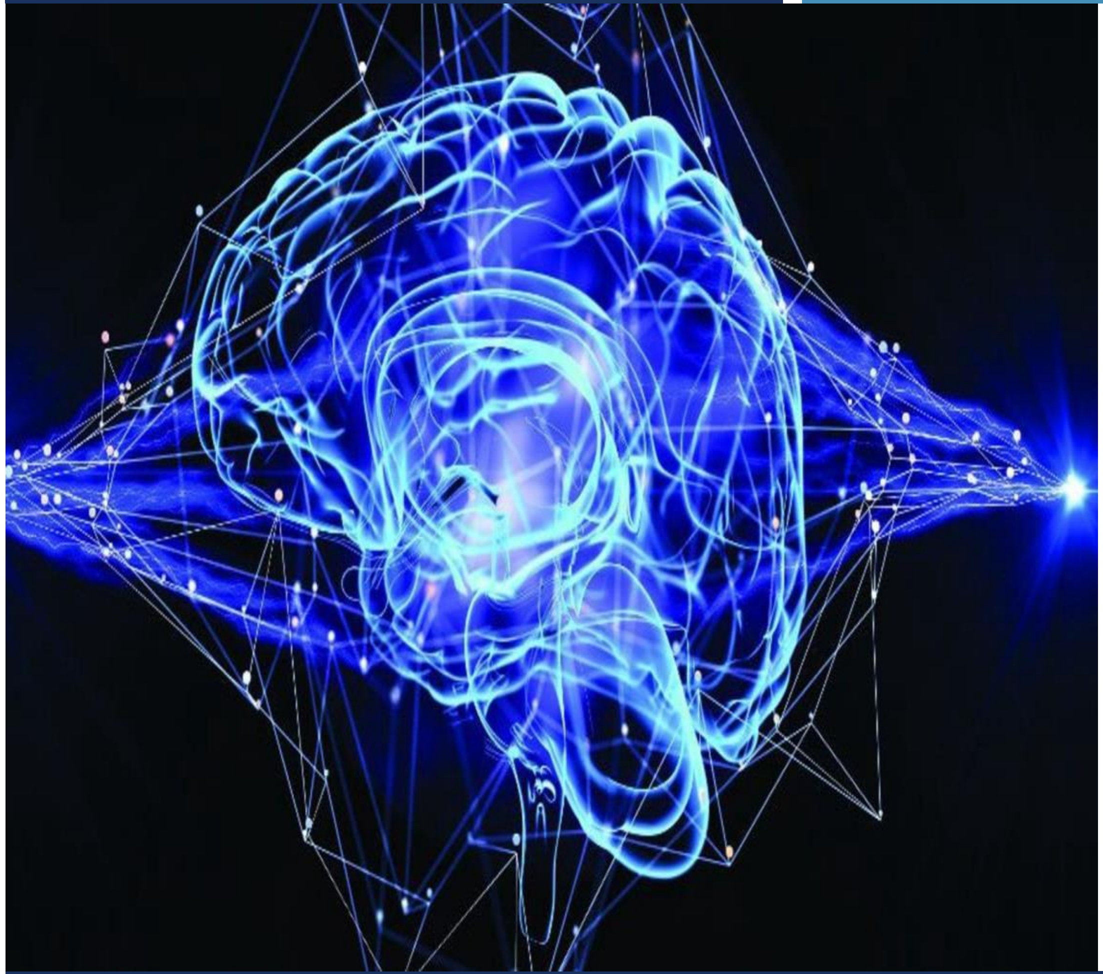
Clean Speech:
Pristine audio recordings featuring isolated and clear human speech, crucial for tasks like speech recognition and transcription.

Speech with Noise:
Audio instances where human speech is accompanied by various interferences or background noise, relevant for noise reduction and speech enhancement.

AUDIO CLASS DIVISION

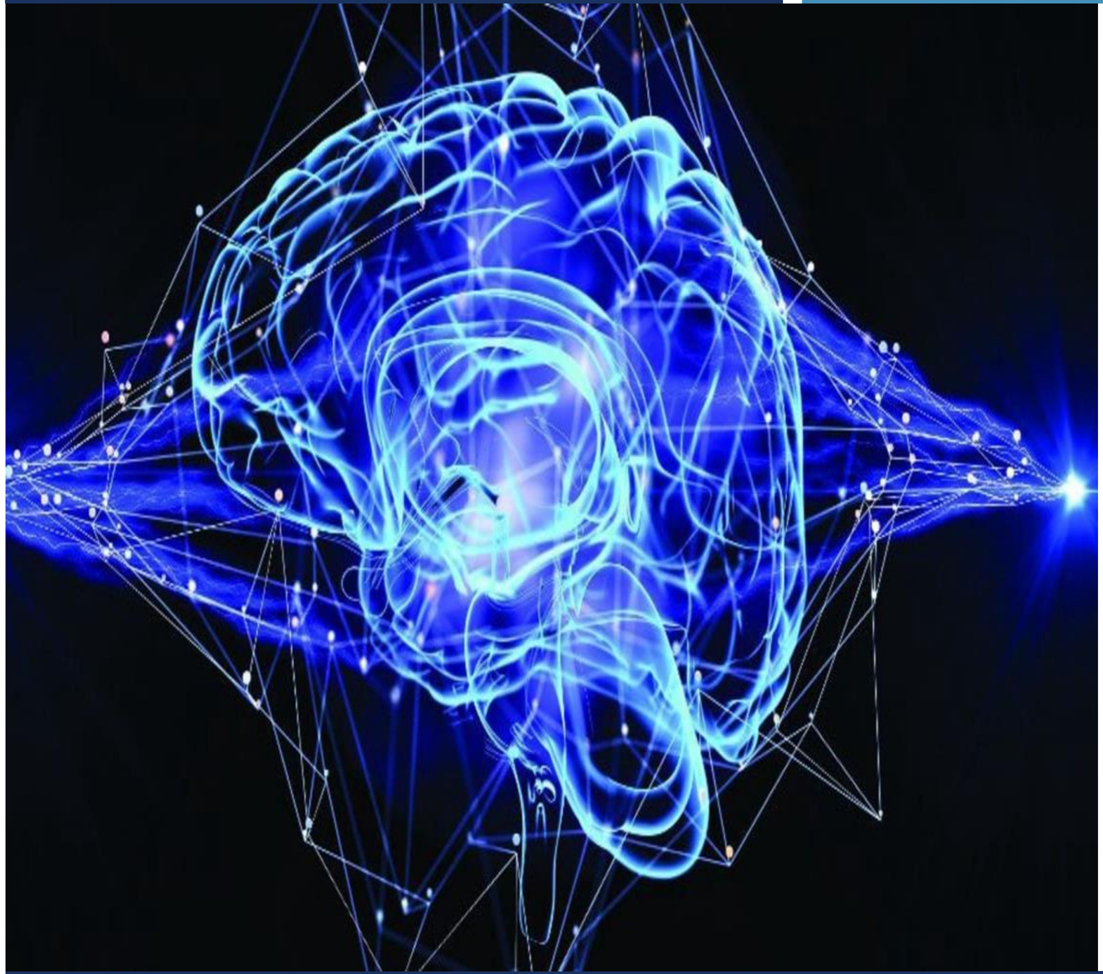


AUDIO WAVE FORM



CONVOLUTIONAL NUERAL NETWORK(CNN)

AUDIO MODEL IMPLEMENTATION



Convolutional Neural Network (CNN) Implementation

Architecture: Employed CNN designed for audio pattern recognition.

Efficiency: Captures intricate audio features using Conv1D and MaxPooling layers.

Training: Trained on 5,500 samples across 4 classes with 'adam' optimizer.

Accuracy: Achieved strong accuracy in nuanced audio attribute classification.

AUDIO MODEL IMPLEMENTATION

```
Epoch 1/5
172/172 [==============================] - 25s 139ms/step - loss: 1.6414 - accuracy: 0.2560
Epoch 2/5
172/172 [==============================] - 25s 143ms/step - loss: 1.3249 - accuracy: 0.3916
Epoch 3/5
172/172 [==============================] - 24s 143ms/step - loss: 1.0811 - accuracy: 0.6102
Epoch 4/5
172/172 [==============================] - 24s 139ms/step - loss: 0.6037 - accuracy: 0.8373
Epoch 5/5
172/172 [==============================] - 24s 137ms/step - loss: 0.2256 - accuracy: 0.9647
```

CNN RESULTS

```
# Example usage:
audio_file = '/content/drive/MyDrive/audios/Agents of Secret Stuff.mp4'
predicted_class = predict_audio_class(audio_file)

print('Predicted class:', predicted_class)
```

```
<ipython-input-38-f188030f32f6>:15: UserWarning: PySoundFile failed. Trying audioread instead.
  audio, sr = librosa.load(audio_file, sr=sample_rate, duration=duration)
/usr/local/lib/python3.10/dist-packages/librosa/core/audio.py:184: FutureWarning: librosa.core.audio.__audioread_load
	Deprecated as of librosa version 0.10.0.
	It will be removed in librosa version 1.0.
  y, sr_native = __audioread_load(path, offset, duration, dtype)
1/1 [==============================] - 0s 109ms/step
Predicted class: ['SPEECH WITH MUSIC']
```

CNN RESULTS

Objective:

Detect changes in speakers during audio segments.
Improve speaker diarization accuracy for multi-speaker audio analysis.

Approach:

Develop a computational model to automatically identify transitions between different speakers.
Utilize advanced signal processing and machine learning techniques.

SPEAKER CHANGE MODEL

Key Steps:

Feature Extraction:

- Extract relevant audio features, such as Mel-Frequency Cepstral Coefficients (MFCCs).
- Convert audio data into a suitable representation for analysis.

Clustering:

- Apply clustering algorithms, such as Gaussian Mixture Models (GMM), to segment audio frames.
- Group frames into clusters based on similarity in feature space.

Change Detection:

Analyze transitions between clusters to identify speaker changes.
Establish thresholds or rules for determining significant speaker shifts.

SPEAKER CHANGE MODEL

Benefits:

- Enhance accuracy in recognizing different speakers within audio segments.
- Facilitate applications like transcription, voice recognition, and content indexing.

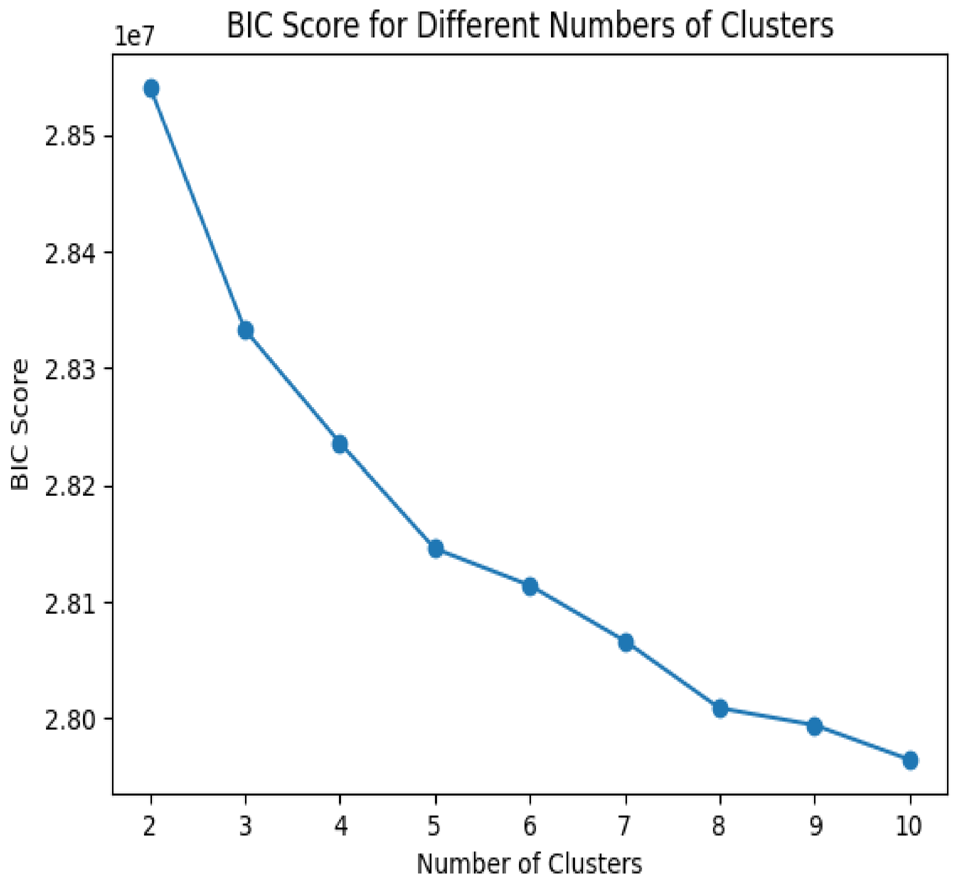
Challenges:

- Handling overlapping speech or abrupt changes.
- Fine-tuning model parameters for optimal performance.

Future Work:

- Integration with deep learning techniques for improved speaker change detection.
- Exploration of real-time applications for instant speaker recognition.

SPEAKER CHANGE MODEL




```
Optimal Number of Clusters: 10
```

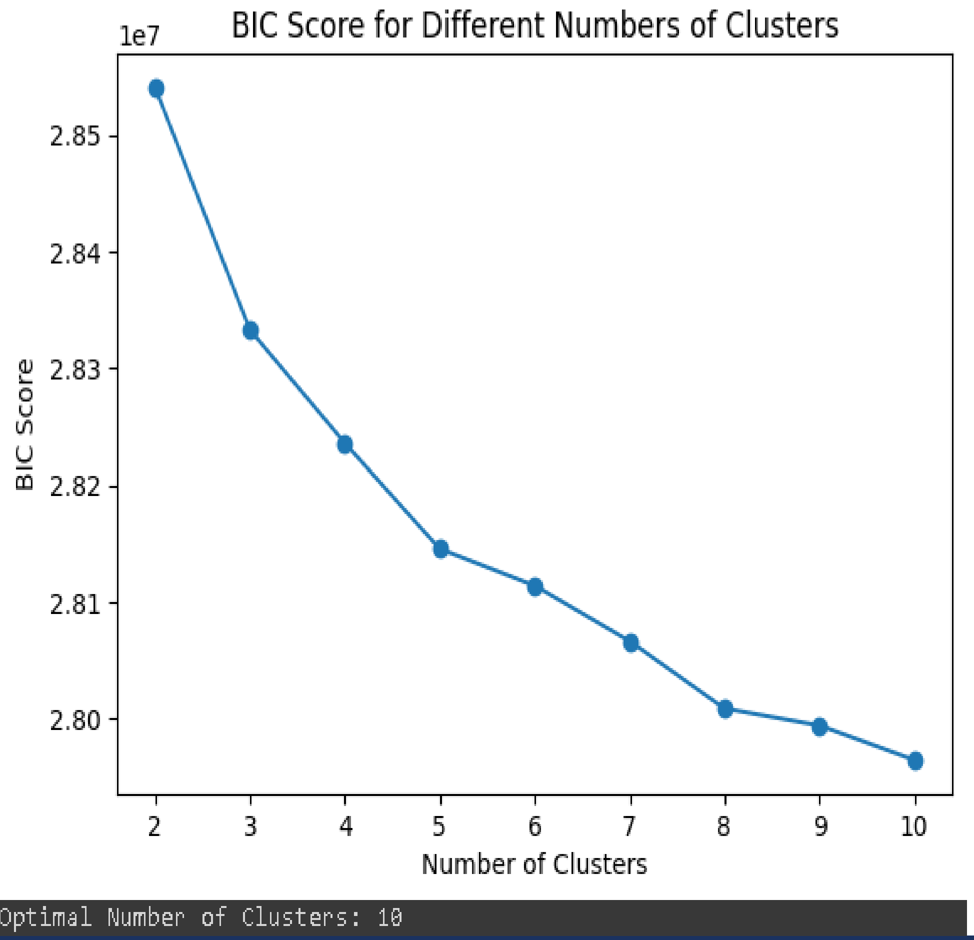
Voice Activity Detection (VAD):

- Utilized Voice Activity Detection to identify segments with speech presence in the audio.
- Ensured alignment of VAD results with MFCC features for accurate processing.

MFCC Feature Extraction:

- Extracted Mel-Frequency Cepstral Coefficients (MFCC) features from the audio data.
- Adapted feature dimensions to match VAD results for consistent analysis.
- Gaussian Mixture Model (GMM) Clustering.

SPEAKER DIARIZATION USING GMM



Gaussian Mixture Model (GMM) Clustering:

- Employed GMM-based clustering for audio segmentation into speaker-like segments.
- Chose the number of mixtures based on the desired complexity.

Frame-Level Clustering:

- Clustered audio frames using the GMM-based model to identify speaker boundaries.
- Resulted in a sequence of clustered frames.

Speaker Diarization:

- Created an initial hypothesis for speaker diarization using the clustered frames.
- Generated pass 1 hypothesis aligned with VAD segments.

SPEAKER DIARIZATION USING GMM

```
[ ] spkdf = speakerdiarisationdf(pass1hyp, frameRate, wavFile)
    spkdf["TimeSeconds"] = spkdf.EndTime - spkdf.StartTime
```

```
print(spkdf)
```

```
                    Audio SpeakerLabel  StartTime  EndTime  TimeSeconds
0    Checkpoint 1956.wav    Speaker 3       3.04    22.08        19.04
1    Checkpoint 1956.wav    Speaker 2      22.08    34.08        12.00
2    Checkpoint 1956.wav    Speaker 1      34.08    37.08         3.00
3    Checkpoint 1956.wav    Speaker 7      37.08    43.08         6.00
4    Checkpoint 1956.wav    Speaker 4      43.08    46.08         3.00
..                   ...          ...        ...      ...          ...
669  Checkpoint 1956.wav    Speaker 6    4779.18  4782.22         3.04
670  Checkpoint 1956.wav    Speaker 3    4782.22  4788.52         6.30
671  Checkpoint 1956.wav    Speaker 6    4788.52  4793.86         5.34
672  Checkpoint 1956.wav    Speaker 4    4793.86  4799.48         5.62
673  Checkpoint 1956.wav    Speaker 1    4799.48  4797.24        -2.24

[674 rows x 5 columns]
```

SPEAKER DIARIZATION USING GMM

Enhanced Audio Analysis:

- Seamlessly combine an advanced audio classification model with a robust speaker change detection algorithm.
- Achieve a comprehensive solution for extracting meaningful insights from audio data.

Unified Processing:

“Harness CNN audio classification to categorize segments (e.g., 'Speech with Music' or 'Clean Speech'). Effortlessly shift to the speaker change model for precise speaker identification during speech-related sections.“

Efficient Workflow:

- Automatic routing based on audio classification ensures targeted processing for speech-related content.
- Speaker diarization algorithm accurately pinpoints speaker changes, enriching analysis results.

INTEGRATION OF AUDIO MODELS

Real-World Applications:

- Optimize multimedia content indexing and organization by identifying both audio content types and speaker transitions.
- Enhance transcription services, voice assistants, and meeting analytics with contextual insights.

Responsive Adaptation:

- Dynamically tailor processing based on the presence of speech-related content.
- Efficiently handle scenarios with or without speech for seamless and reliable performance.

Future Prospects:

- Potential for further integration with deep learning and real-time techniques.
- Pave the way for more sophisticated applications and adaptive processing.

INTEGRATION OF AUDIO MODELS

```
<ipython-input-59-69695e937ad6>:18: UserWarning: PySoundFile failed. Trying audioread instead.
  audio, sr = librosa.load(audio_file, sr=sample_rate, duration=duration)
/usr/local/lib/python3.10/dist-packages/librosa/core/audio.py:184: FutureWarning: librosa.core.audio.__audioread_load
        Deprecated as of librosa version 0.10.0.
        It will be removed in librosa version 1.0.
  y, sr_native = __audioread_load(path, offset, duration, dtype)
1/1 [==============================] - 0s 97ms/step
Predicted class: ['SPEECH WITH MUSIC']
Speaker diarization output:                    Audio SpeakerLabel  StartTime  EndTime  TimeSeconds
0    Checkpoint 1956.wav    Speaker 3       3.04    22.08        19.04
1    Checkpoint 1956.wav    Speaker 2      22.08    34.08        12.00
2    Checkpoint 1956.wav    Speaker 1      34.08    37.08         3.00
3    Checkpoint 1956.wav    Speaker 7      37.08    43.08         6.00
4    Checkpoint 1956.wav    Speaker 4      43.08    46.08         3.00
..                   ...          ...        ...      ...          ...
669  Checkpoint 1956.wav    Speaker 6    4779.18  4782.22         3.04
670  Checkpoint 1956.wav    Speaker 3    4782.22  4788.52         6.30
671  Checkpoint 1956.wav    Speaker 6    4788.52  4793.86         5.34
672  Checkpoint 1956.wav    Speaker 4    4793.86  4799.48         5.62
673  Checkpoint 1956.wav    Speaker 1    4799.48  4797.24        -2.24

[674 rows x 5 columns]
```

INTEGRATION OF AUDIO MODELS

INTEGRATION OF AUDIO MODELS

```
<ipython-input-50-ad5fb74739d4>:18: UserWarning: PySoundFile failed. Trying audioread instead.
  audio, sr = librosa.load(audio_file, sr=sample_rate, duration=duration)
/usr/local/lib/python3.10/dist-packages/librosa/core/audio.py:184: FutureWarning: librosa.core.audio.__audioread_load
        Deprecated as of librosa version 0.10.0.
        It will be removed in librosa version 1.0.
  y, sr_native = __audioread_load(path, offset, duration, dtype)
1/1 [==============================] - 0s 84ms/step
Predicted class: ['NO SPEECH']
No speech detected in the audio.
```

Objective: Implement an advanced video model leveraging Retina Face for accurate and efficient face detection and tracking within video streams.

RetinaFace Framework: Leveraging the power of Retina Face, a state-of-the-art face detection and alignment framework, to achieve precise face localization even in challenging video scenarios.

Key Features:

- **Multi-Tasking:** Simultaneously detects faces and aligns them for consistent orientation, contributing to reliable face tracking.
- **Robustness:** Handles variations in scale, pose, and lighting conditions, enhancing model's adaptability across diverse video content.
- **Efficiency:** Achieves real-time performance, allowing seamless integration into video processing pipelines.

VIDEO MODEL

Workflow:

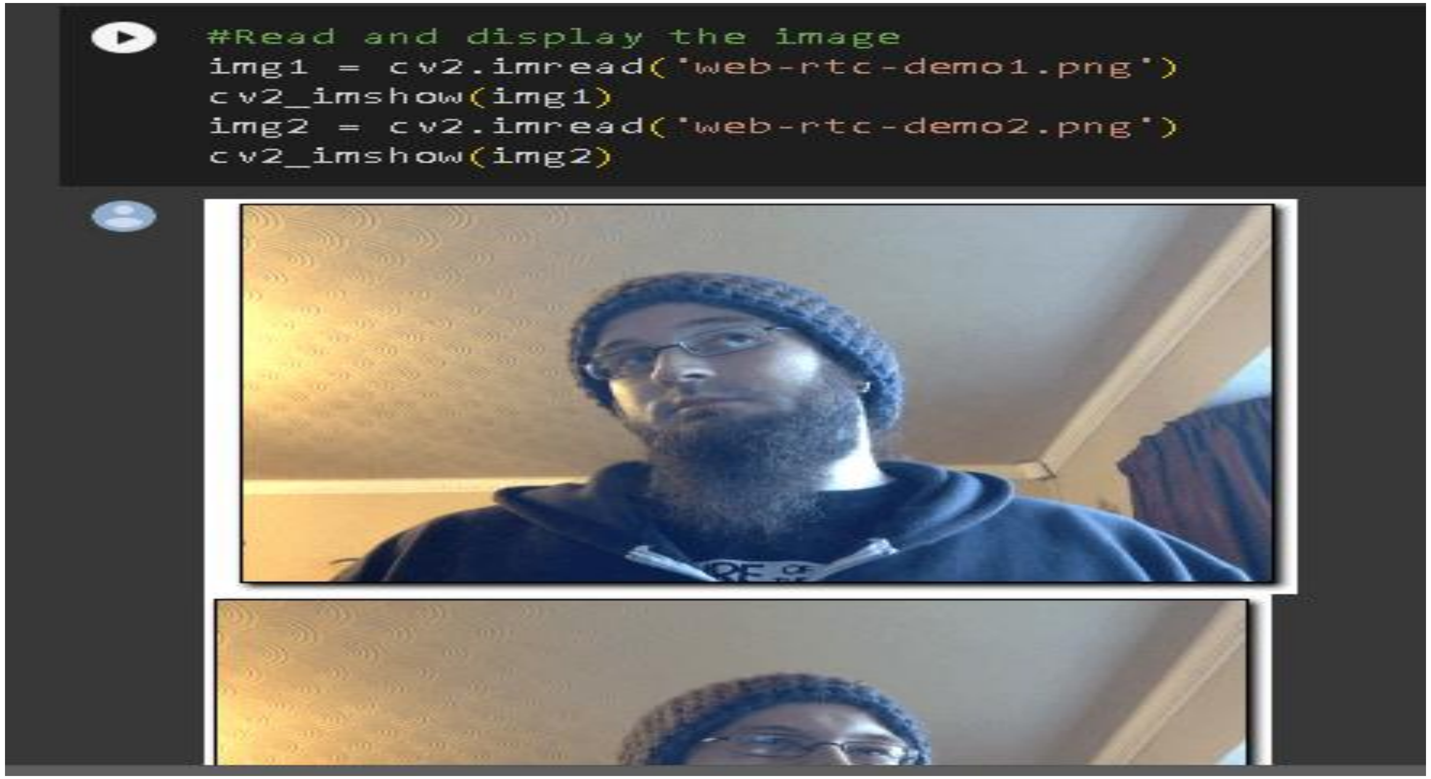
Frame Extraction: Break down video into frames for individual analysis.
Retina Face Inference: Apply Retina Face on each frame to detect and align faces.

Tracking Integration: Integrate detection results for continuous face tracking across frames.

Benefits:

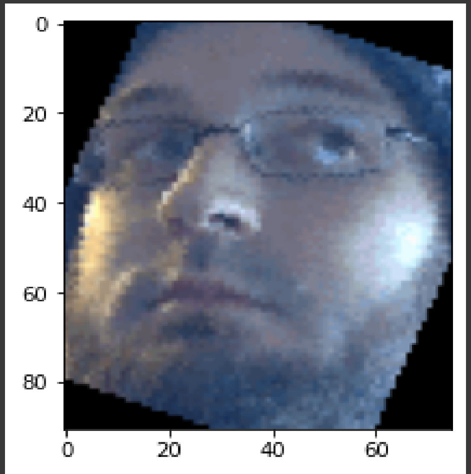
Elevate video content analysis with precise and responsive face tracking.
Unlock creative possibilities for video enhancement and customization.
Contribute to enhanced user experiences in diverse video-centric applications.

VIDEO MODEL



VIDEO MODEL IMPLEMENTATION RESULTS

```
[ ] #Extract individual faces from the image
    import matplotlib.pyplot as plt
    ext_faces = RetinaFace.extract_faces(img_path = "web-rtc-demo1.png", align = True)
    for face in ext_faces:
      plt.imshow(face)
      plt.show()
```



VIDEO MODEL IMPLEMENTATION RESULTS

- Finding scores of audio and video models.
- Fusion of the model’ s score.
- Thresholding the score to find out the active speaker, to eradicate the conflict between two active speakers
- Find max active speaker score.
- Pan the camera to the active predicted speaker based on the max score.

SCORE BASED FUSION OF AUDIO VISUAL MODEL

Subtitle Support: Provide Subtitles below the video

Deep Learning Architectures: Explore CNNs, RNNs, and Transformers for enhanced speaker detection.

Real-time Processing: Optimize for low latency to ensure real-time active speaker detection.

Robust Noise Handling: Develop noise-resistant models for accurate detection in varying environments

.

Adaptation and Transfer Learning: Fine-tune models for specific environments to boost performance.

FUTURE ENHANCEMENT

Personalized Models: Create user-specific profiles for adapting to individual speaking styles.

Emotion Analysis: Integrate emotion detection for added contextual insights.

Privacy Measures: Implement secure data handling to address privacy concerns.

User Interface Integration: Develop a user-friendly real-time interface for easy visualization.

FUTURE ENHANCEMENT

PROJECT TASK DISTRIBUTION

KASHAF KHAN	HAFSA ZAFAR	ALEESHA AHMED
LITERATURE REVIEW	LITERATURE REVIEW	LITERATURE REVIEW
VIDEO MODEL IMPLEMENTATION	SPEAKER CHANGE MODEL IMPLEMENTATION	AUDIO MODEL IMPLEMENTATION
AUDIO-VISUAL MODEL FUSION	DOCUMENTATION	AUDIO MODELS INTEGRATION

THANK YOU