T.R.

GEBZE TECHNICAL UNIVERSITY

FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

DELAY REMOVAL ON TELEVISION INTERVIEWS

ŞEYDA ÖZER

SUPERVISOR
PROF. DR. YUSUF SINAN AKGÜL

GEBZE
2023

T.R.
GEBZE TECHNICAL UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

DELAY REMOVAL ON TELEVISION INTERVIEWS

ŞEYDA ÖZER

SUPERVISOR
PROF. DR. YUSUF SINAN AKGÜL

2023
GEBZE

This study has been accepted as an Undergraduate Graduation Project in the Department of Computer Engineering on 15/01/2023 by the following jury.

JURY

Member
(Supervisor) : Prof. Dr. Yusuf Sınan Akgül

Member : Prof. Dr. İbrahim Soğukpınar

ABSTRACT

Today, with the developing technology, people can reach the things they want easily and quickly. They don't want to waste their time. It makes them uncomfortable to watch the delays that have occurred while watching a recorded conversation.

In this project, the focus is on how to eliminate delays caused by various reasons. Videos were taken as input and converted to audio using video processing libraries. Using audio processing libraries, operations were carried out to help detect delays on audios. Delays were detected with the help of voice activity detection libraries. Detected delays have been deleted from the videos.

Keywords: video processing, audio processing, voice activity detection.

ÖZET

Günümüzde gelişen teknolojiyle birlikte insanlar istedikleri şeylere kolay ve hızlı ulaşabiliyorlar. Vakitlerini boşa harcamak istemiyorlar. Kaydedilmiş bir karşılıklı konuşmayı izlerken meydana gelmiş gecikmeleri tekrar izlemek onları rahatsız ediyor.

Bu projede, çeşitli nedenlerden dolayı meydana gelmiş gecikmelerin nasıl ortadan kaldırılacağına odaklanılmıştır. Videolar input olarak alınmış, video işleme kütüphaneleri kullanılarak sese dönüştürülmüştür. Sesler üzerinde ses işleme küpüthaneleri kullanılarak gecikme tespitine yardımcı olacak işlemler yapılmıştır. Ayrıca konuşma tespiti yapan kütüphanelerin de yardımıyla gecikmeler tespit edilmiştir. Tespit edilen gecikmeler videolardan silinmiştir.

Anahtar Kelimeler: video işleme, ses işleme, konuşma tespiti.

ACKNOWLEDGEMENT

I would like to my special thanks of gratitude to my supervisor Prof. Dr. Yusuf Sinan Akgül for his guidance and support in completing my project.

I am also grateful to my family and friends for the support they have given me during my education.

Şeyda Özer

LIST OF SYMBOLS AND ABBREVIATIONS

Symbol or Abbreviation	:	**Explanation**
VAD	:	Voice Activity Detection

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Delays can occur in TV interviews. When studio presenter questions a remote interviewee, the two can't hear one another instantly in real time. There is a delay between the time the studio presenter speaks to a remote interviewee and when the response arrives, because the signal (digital and/or satellite link) takes time to get there and get back.

Also, when broadcasting across a distance data needs to be compressed and it is buffered for this purpose. When broadcasting live this buffering can cause overhead in the processing of the signal and can cause the signal to be delayed. Reducing the buffer shortens the delay. However, when a remote interviewee is on location they don't always have access to necessary equipment to be able to stream without delay.

In some cases there is a deliberate delay called broadcast delay. In radio and television, broadcast delay refers to the practice of intentionally delaying broadcast of live material. A short delay is often used to prevent profanity, bloopers, violence, or other undesirable material from making it to air, including more mundane problems such as technical malfunctions or coughing.

For these and similar reasons, the remote interviewee smile while they wait to hear a question the audience has already heard, because he/she can't hear it immediately. The audience is not satisfied with these delays and does not want to wait for the answer of the remote interviewee. In this case, delays need to be reduced. When the delays are reduced, the audience will watch the broadcast as if there were no delays, they will not be aware of the delays. By reducing the delays, the audience will be prevented from getting bored and impatient.

Figure 1.1: TV interview

1.1. Project Description

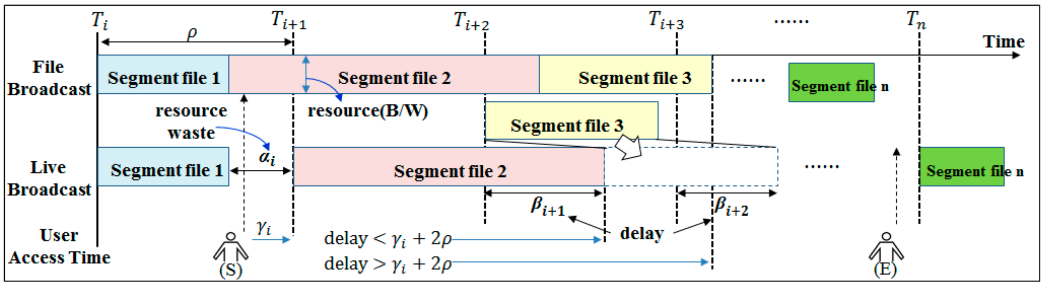

Figure 1.2: Example of a delay

We are able to satisfy the audience by removing the delays caused by various reasons. However, we do not prevent delays. We remove these delays that have already occurred so that they do not disturb when watched later. In other words, this project takes 2 videos containing conversation as input and detects delays. When the delay is detected, we delete these delays from the related videos, to provide the pleasure of watching videos without delay.

1.2. Project Purpose

The aim of the project is to remove delays from videos. The audience should watch the video as if there were no delays and should not be aware of the delays.

2. LITERATURE REVIEW

Delay detection can be done via audio. In this case, audio processing will be done. Video processing will also be done for operations such as extracting audio from the video and cutting the video. Many Python libraries are available for such operations.

2.1. Video Processing

MoviePy is a Python module for video editing, which can be used for basic operations (like cuts, concatenations, title insertions), video compositing (a.k.a. non-linear editing), video processing, or to create advanced effects. It can read and write the most common video formats, including GIF.

2.1.1. Video to Audio Convert

In order to process the audio, the video must be converted to audio. There are several libraries and techniques available in Python for the conversion of Video to Audio. One such library is Movie Editor. This library reads the video and then writes the audio extracted from the video to the file.

2.1.2. MoviePy - cutout

Cut out video is the trimmed video or we can say it as some few seconds get cut from the original video, it is used to skip some part of the video. Cut out means to get the video from original video with skip of some time in between the video.

2.1.3. MoviePy - subclip

Sub Clip, divide the video into sub-clips. The video is divided at specific intervals. So the sub-clips are processed.

2.1.4. MoviePy - concatenate_videoclips

Concatenate video clips, combine the videos that wanted and provides a single video.

2.2. Voice Activity Detection

SpeechBrain 2.1 is an open-source all-in-one speech toolkit based on PyTorch. It is designed to make the research and development of speech technology easier. Speechbrain's pre-trained VAD model can be used for voice activity (speech) detection. The goal of Voice Activity Detection (VAD) is to detect the segments containing speech within an audio recording. The VAD could provide in the output the boundaries where speech activity is detected.

Figure 2.1: Speechbrain Logo

A VAD plays a crucial role in many speech processing pipelines. It is used when we would like to apply the processing algorithms to the speech parts of the audio recording only.

2.2.1. VAD pipeline details

The pipeline for detecting the speech segments is the following:

1. Compute posteriors probabilities at the frame level.
2. Apply a threshold on the posterior probability.
3. Derive candidate speech segments on top of that.
4. Apply energy VAD within each candidate segment (optional). This might break down long sentences into short one based on the energy content.
5. Merge segments that are too close.
6. Remove segments that are too short.
7. Double-check speech segments (optional). This could is a final check to make sure the detected segments are actually speech ones.

Figure 2.2: VAD pipeline details

2.3. Audio Processing

Torchaudio is a library for audio and signal processing with PyTorch. It provides I/O, signal and data processing functions, datasets, model implementations and application components. With this library, audio can be loaded, resampled and saved.

3. METHOD AND SYSTEM ARCHITECTURE

3.1. System Requirements

Software and hardware requirements:

- Sufficient storage space,
- Sufficient computer powering,
- Pytorch $\geq$ 1.7,
- Python $\geq$ 3.7,
- Linux-based distributions and macOS for SpeechBrain

3.2. Architecture and Implementation Details

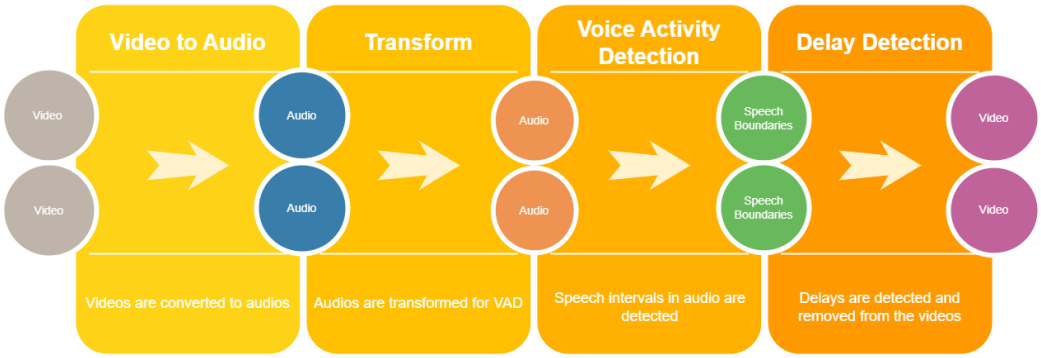

Figure 3.1: Delay Detection

3.2.1. Video to Audio

Audio delay in videos should be detected. In order for the delay in the audio to be detected, the video must be converted to audio. Delay detection will be done on audio. MoviePy module was used to convert video to audio. The extracted audio was saved with ".wav" extension for later use.

```
def video2audio(video_file, audio_file):
  video = moviepy.editor.VideoFileClip(video_file)
  audio = video.audio
  audio.write_audiofile(filename=audio_file, fps = VAD.sample_rate)
  return video
```

Figure 3.2: video to Audio

3.2.2. Transform

Speechbrain's VAD (Voice Activity Detection) model was used for the voice activity detection. Audio converted from video could not be used directly for the voice activity detection. Because the audio had to has the VAD's sample rate and one channel for VAD. torchaudio.transforms.Resample was used for transform. It is used to resample a signal from one frequency to another. The transformed audios were re-recorded.

```
def voice_transform(audio_file):
  waveform, sample_rate = torchaudio.load(audio_file, normalize=True)
  transform = transforms.Resample(sample_rate, VAD.sample_rate)
  waveform = transform(waveform)
  waveform = waveform[0].unsqueeze(0)
  torchaudio.save(audio_file, waveform, VAD.sample_rate)
```

Figure 3.3: Transforming Audio

3.2.3. Voice Activity Detection

Speechbrain's VAD model was used for the voice activity detection.

```
from speechbrain.pretrained import VAD
VAD = VAD.from_hparams(source="speechbrain/vad-crdnn-libriparty", savedir="pretrained_models/vad-crdnn-libriparty")
```

Figure 3.4: Using VAD model

The VAD provided the limits at which speech activity was detected at the output.

```
first_boundaries = VAD.get_speech_segments(voice_path+"first.wav")
second_boundaries = VAD.get_speech_segments(voice_path+"second.wav")
```

Figure 3.5: Getting Speech Activity

3.2.4. Delay Detection

The limits provided by the VAD model were used for delay detection.

```
delay_detection(first_boundaries.tolist(), second_boundaries.tolist(), video1, video2)
```

Figure 3.6: Using speech limits

By looking at the speech boundaries, delay was detected from the difference between the speech boundaries. If the delay was less than 2 seconds, the delay was ignored. 3.7 If the delay is longer than 2 seconds, that part of the video is cropped. The trimmed video was saved with a new name. 3.8

```
delay = second_boundaries[i][0] - first_boundaries[i][1]
if(delay > 2) :# there is a delay
  clipDelay(video2, video_path+"new_second.mp4", first_boundaries[i][1], second_boundaries[i][0])
```

Figure 3.7: Delay detection

```
def clipDelay(clip, filename, start, end):
    clip1 = clip.subclip(0, start)
    clip2 = clip.subclip(end, clip.duration)
    clip = moviepy.editor.concatenate_videoclips([clip1, clip2])
    clip.write_videofile(filename)
```

Figure 3.8: Clipping delay

4. EXPERIMENTS

The goal is to process 2 videos that contain conversations with delays and remove these delays. The videos of the speakers are inputs.

4.1. Results

4.1.1. Display Audios

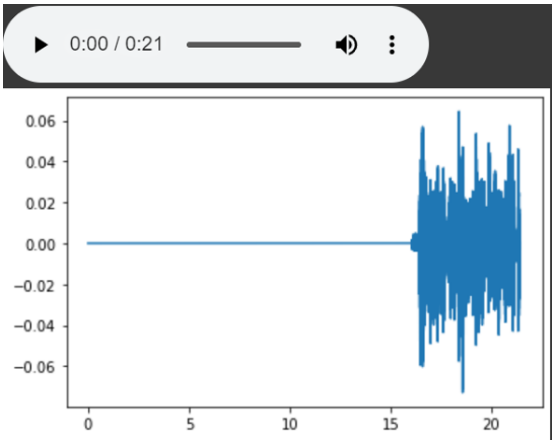

Figure 4.1: First audio

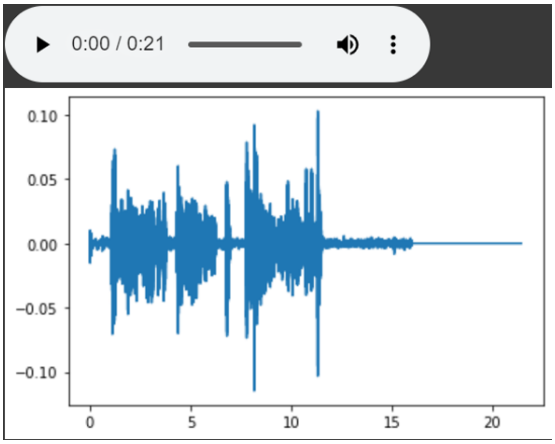

Figure 4.2: Second audio

4.1.2. Speech Boundaries in Audios

```
segment_001  0.00  16.32 NON_SPEECH
segment_002  16.32  19.99 SPEECH
```

Figure 4.3: Speech boundaries of the first audio

```
segment_001  0.00  11.82 SPEECH
```

Figure 4.4: Speech boundaries of the second audio

4.1.3. Display Audios by Overlapping

If we show it by overlapping, we can easily see the delay.

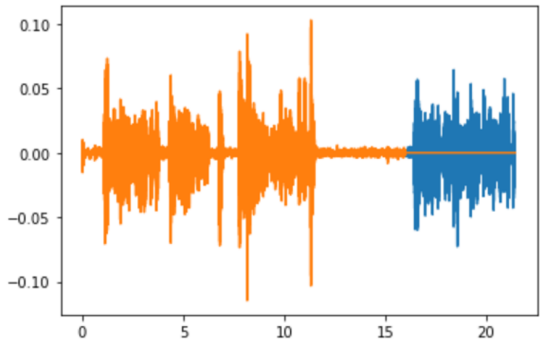

Figure 4.5: Display audios by overlapping

4.1.4. Remove Delay from the Video

When a delay is detected, it removed from the video. 4.6 The clipped video is saved by giving it a new name. 4.7

If we show it again by overlapping, we can easily see that the delay has removed. 4.8

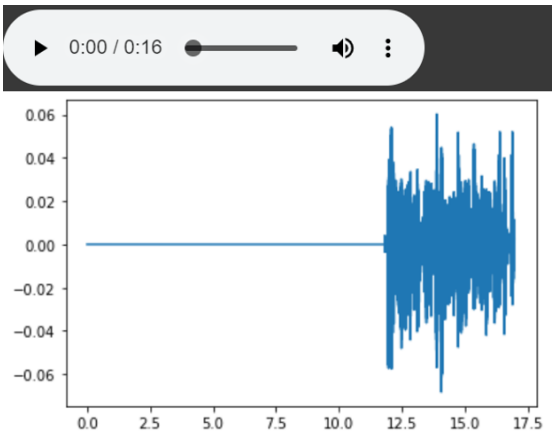

Figure 4.6: First audio with no delay

```
[MoviePy] Writing audio in /content/gdrive/MyDrive/Graduation_Project/Inputs/Voices/new_first.wav
100%|██████████| 136/136 [00:00<00:00, 1049.83it/s][MoviePy] Done.
```

Figure 4.7: Saving clipped video

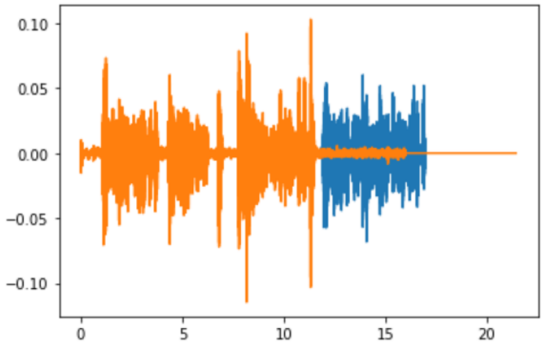

Figure 4.8: Audios by overlapping with no delay

5. DISCUSSION AND CONCLUSION

Delays have been removed from videos. However, trimming and re-writing the video took longer than expected.

A dataset containing 25 videos was created to test the project.

6. BIBLIOGRAPHY

[1] TV interview image link2.2

[2] Seo, H. Kim, G. "DASH Live Broadcast Traffic Model: A Time-Bound Delay Model for IP-Based Digital Terrestrial Broadcasting Systems" Applied Sciences 2021, 11(1), 247, https://doi.org/10.3390/app11010247

[3] Zhang, C. Liu, J. "On Crowdsourced Interactive Live Streaming: A Twitch. TV-Based Measurement Study" in NOSSDAV '15: Proceedings of the 25th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video, March 2015, Pages 55-60, https://doi.org/10.1145/2736084.2736091.

[4] "Achieving Broadcast-Grade Low Latency in Live Streaming" in Streaming Media. Nov/Dec 2018, Vol. 15 Issue 8, p16-25. 10p.

[5] Tang, K. Huo, L.-J. "Optimizing Synchronization of Tennis Professional League Live Broadcast Based on Wireless Network Planning" in Hindawi, Mobile Information Systems, Volume 2021, Article ID 8732115, https://doi.org/10.1155/2021/8732115.

[6] Kelsey, K. D. "Participant interaction in a course delivered by interactive compressed video technology". American Journal of Distance Education 12, no: 1 (2000): 63-74.

[7] Wu, Shan-Hung. Chen, Chung-Min. "Minimizing Broadcast Delay in Location-Based Channel Access Protocols". 2011 Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN), 1-6 Jul, 2011.

[8] https://zulko.github.io/moviepy/

[9] https://en.wikipedia.org/wiki/Voice_activity_detection

[10] https://speechbrain.github.io

[11] Voice Activity Detection

[12] Tan, Z. Sarkar, A. Dehak, N. ”rVAD: An unsupervised segment-based robust voice activity detection method”. Computer Speech Language, Volume 59, January 2020, Pages 1-21. https://doi.org/10.1016/j.csl.2019.06.005

[13] https://speechbrain.readthedocs.io/en/latest/installation.html

[14] https://pointerclicker.com/how-long-is-the-delay-in-live-tv/

[15] moviepy-getting-cut-out-of-video-file-clip

[16] https://huggingface.co/speechbrain/vad-crdnn-libriparty