Goals

- □ Remove all parts of the lecture that do not give the user information. (silences, technical difficulties, filler words)
- □ Remove accents
- □ Improve the speakers tone, pacing, general oral quality
- □ Use learning theory to enforce good learning practices (such as intelligent breaks, active learning)
- □ Give lecturers the same speech qualty as professional speakers
- [?] Consume lectures regardless of language

Features

- ☒ Speeding up silent parts of the lecture
- ☒ Speed up lecture
- □ Read text from slides and outputs it
- □ Improve audio quality by denoising, this actually improves audio quality quite significantly. A 1hr20 minute lecture before audio denoising takes 20 minutes to process, while it takes 1 hr to process with denoising.
- [>] Cut out filler words from lecture
- □ Accent, filler word, and crappy explanation filter.
- □ Incorporate captions and transcripts
- [?] Potentially remove a lot of the frames if the video is just a slideshow. (reduce file size, increase processing speed)
- □ Transcript generation with whisper (1h20m video took 2h9m)
- □ TTS first iteration took 1h5m for 30 minutes of lecture
- □ TTS with 12 workers takes 4m38s for 5m30s of audio!

TODO: Smooth audio when using tts, don’t need to remove silence because all audio is synthesized. My current system with voice clonning is actually very good. It is probably 70% of the way on par with normal lecture. I need to remove the awkward pauses that arise with silences being sped up and improve transcript using AI

Then this will probably speed up lecture by 5x

Without the voice clonning, it is very hard to understnda the lecture because of the accent + speed, and also because of the “blips” that happen when it is supposed to be silent.

How do I use an LLM to take the transcript, improve it, and then stitch it together with the video?

Some nice properties of the transcript right now are that the silent parts are defined as parts outside the start and end boundaries. This is good because I can find the corresponding video at these boundaries and speed it up so it is the same length as the audio. I could entirely cut out these silences because they are generated “artifically”, which would be more efficient.

If I can split the transcript up "by idea" then I could match the newly generated transript with the previous one, and I can still entirely cut out fake silences.

I could pass in only the text from the

What happens when the LLM removes text?

- What happens with the video? I could make a minimum length decrease so it dosen't cut out too much of the video

What happens when the LLM adds text?

- What happens with the video?

Sync up captions with audio

Because I have access to when the professor says the word, I can use it to relate with the slides.

Retain audio that is not related to voice (music, sound effects, etc.) Enable watermark on whole audio Test to see how good whisper and RNNoise do when silence is severely cut out. -> 10% performance improvement

See if RNNoise -> whisper gives better results

```
TARGET_SPEED = 1.0  # TODO: See if changing this parameter gives better results then ex
```

When improving the lecture transcript, take the image from the lecture video as input for more/better data.

Find where questions are being asked and tag them. Use this later down in the lecture optimizer

How to make output text to speech sound more lively/realistic?

"You can read those papers" is pronouned "You can red those papers"

Improve the Openvoice library so audios can be passed in memory instead of reading and writing to/from disk

Make the professor's tone appropriate for the mateiral. Perefrably make him sound excited.

Add speech diarization if mulitple people are speaking

Use a better, more optimal version of whisper.

Pipeline Input lecture video -> denoise the video, increase audio quality (maybe removing silences will help whisper, needs to be tested) Whisper extracts transcript From transcript identify parts of text irrelevant to the material -> remove those From transcript, identify and remove filler words (whisper might not detect them, good!) Might have to clean up the transcript here as it won't be perfect. (i.e. some words captured might not make sense) Change the transcript so a complete thought is spoken every time. Right now, whisper splits it up somewhat aprbirarily. Also break up longer sentences to make it easier for tts packing More cool things can be done here! Improving explanations, rewording things, removing redundant information, adding content :) From the transcript, use a voice clonning AI to read the transcript from the person's voice (accent removal), or read it using an AI voice Speed up the voice clonned AI audio to the desired speed. Create many audio segments, they might not be the same length as before, if that is the case, take the video chunk that was played during the time segments, and then speed it up to match the audio segment's length. There are edge cases and problems with this because of video. There might be off-by-one errors that accumulate, the video might get out of sync with the audio Stitch audio and video segments together, and you should get an optimal lecture.

Remove text irrelevant to material: Example, during a lecture the professor was asking the class about the lighting, wasting a whole minute of our time

```
[01:00.000 --> 01:03.440]  Hi, everyone.
[01:03.800 --> 01:04.540]  Ooh, loud.
[01:08.300 --> 01:10.240]  Let's figure out the lighting.
[01:11.000 --> 01:12.840]  I'm going to give you three options for lighting.
[01:13.900 --> 01:14.680]  Let's see.
[01:17.180 --> 01:18.800]  This is option one.
[01:22.500 --> 01:23.880]  This is option two.
[01:25.560 --> 01:26.820]  Hey, not much difference.
[01:28.440 --> 01:29.720]  This is option three.
[01:31.000 --> 01:31.880]  Ooh.
[01:34.800 --> 01:36.020]  This is option four.
[01:37.920 --> 01:38.720]  Bad option.
[01:40.140 --> 01:41.620]  How many want option one?
[01:42.100 --> 01:42.600]  First one.
[01:44.240 --> 01:45.520]  How many want option two?
[01:46.840 --> 01:48.100]  How many want option three?
[01:48.820 --> 01:50.000]  Three? All right.
[01:50.380 --> 01:50.860]  Okay.
[01:52.620 --> 01:53.320]  All right.
[01:53.420 --> 01:55.620]  You guys are in mood for a movie, eh?
[01:56.020 --> 01:56.380]  Okay.
[01:56.380 --> 01:56.480]  Okay.
```

"or even microservices which is another buzzord that I did not define here" -> "or even microservices". This is a good example of a sentence that is not relevant to the material, and can be removed.

```
4170260 4172100 to function as a
4172100 4172320 service
4172320 4172660 or even
4172660 4173260 microservices
4173260 4173720 which is another
```

```
4173720 4174220 buzzword
4174220 4174860 that I did not
4174860 4175300 define here
4175300 4176460 how efficiently
4176460 4176780 you do it
4176780 4177160 if you do it
4177160 4177480 right
```

Whisper seems to natively take out some of the “phoneme” filler words (such as “um, ah”)

it’s a very open area and there is a very there’s a scarity of education resources in that area.

```
4220820 4221080 right
4221080 4221980 it's a very
4221980 4222620 open area
4222620 4223620 and there
4223620 4224060 is a very
4224060 4225400 there's a scarcity
4225400 4225860 of education
4225860 4226280 resources
4226280 4227280 in that area
```

Lectures are filled with this rambling. The below paragraph instead could be “Today, we’re discussing cloud computing, a recent buzzword. We’ll unearth what it’s all about and attempt to define the term.”

```
148460  155180  So, today we are discussing cloud computing, which is this new buzzword that
```

Interactive Q&A segments (“Yes, you had a question. . . ”) are jarring without audience context. Fix: Integrate questions into the narrative (e.g., “A common question is: What defines a rack? A rack is. . . ”).

```
1759720 1760700 that run inside of them.
1761800 1762760 Yes, you had a question.
1762760 1771020 What do I mean by a rack?
1771120 1772540 Yes, a rack is a collection
1772540 1773420 of computing nodes
```

Add Transitions: Use phrases like “Next,” “In contrast,” or “Historically,” to guide flow.

I am in CS510 information retrival, and right now the professor is struggling with the technology, this is somewhat unacceptable in an academic institution and this should be removed. The future of learning should be polished like youtube videos. Maybe I make a lecture polishing program to remove these. He spent about 10 mins.

Another thing, professors have surveys to understand what students want to learn. Maybe instead the professor can make a bunch of lectures and then the students can choose which one they want to watch. This is a more efficient way of learning.

There is a lot of overlap between courses. I.e. CS545 ML for Signal processing teaches LDA, EM, which is also taught in CS510.

Are professors just machines that take text books, extract knowledge, applying their perspective, and then trying to distill it down? What professors bring is their own perspective and thoughts on the field. THey also relate to all other systems.

Edge cases

- If a movie is being played without any audio, this program will zoom through it at a very fast speed. Maybe this could be detected by seeing how often the image changes from timestep to timestep and identify any videos playing
- If this is a lecture video where the professor rights on the board, there might be problems with taking the video and speeding it up/doing slideshow detection