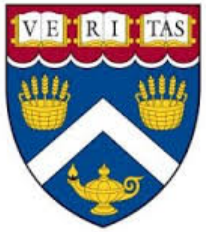
Enhanced Lecture Slides with "Closed Caption" Style Text using Luigi (Tool and/or maybe a Pset)

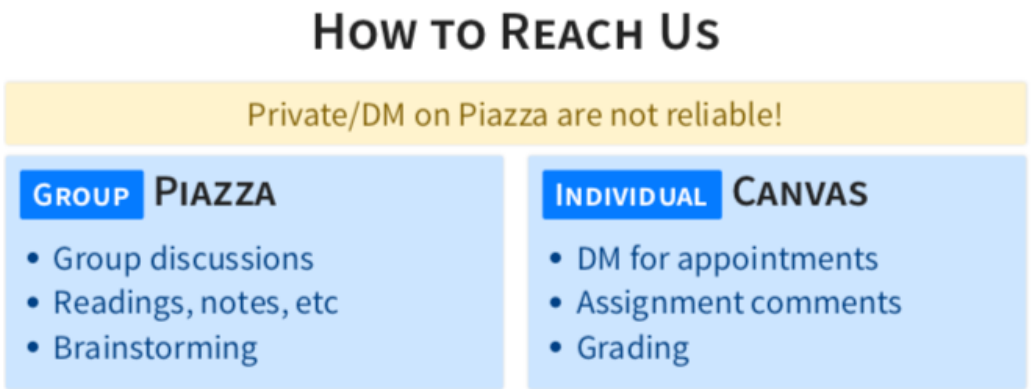
Amit R. Gupta



CSCI E-29 Advanced Python for Data Science
Spring 2019
Harvard University Extension School
Prof. Scott Gorlin

The Project

- Convert a lecture or presentation to text, and map the spoken text to each relevant slide
- One can lookup the entire lecture notes at a later point for easy reference, and keyword search
- Example from Scott Gorlin's Monday lecture (08_dask.pdf, slide4)



Converted Text from Scott's presentation

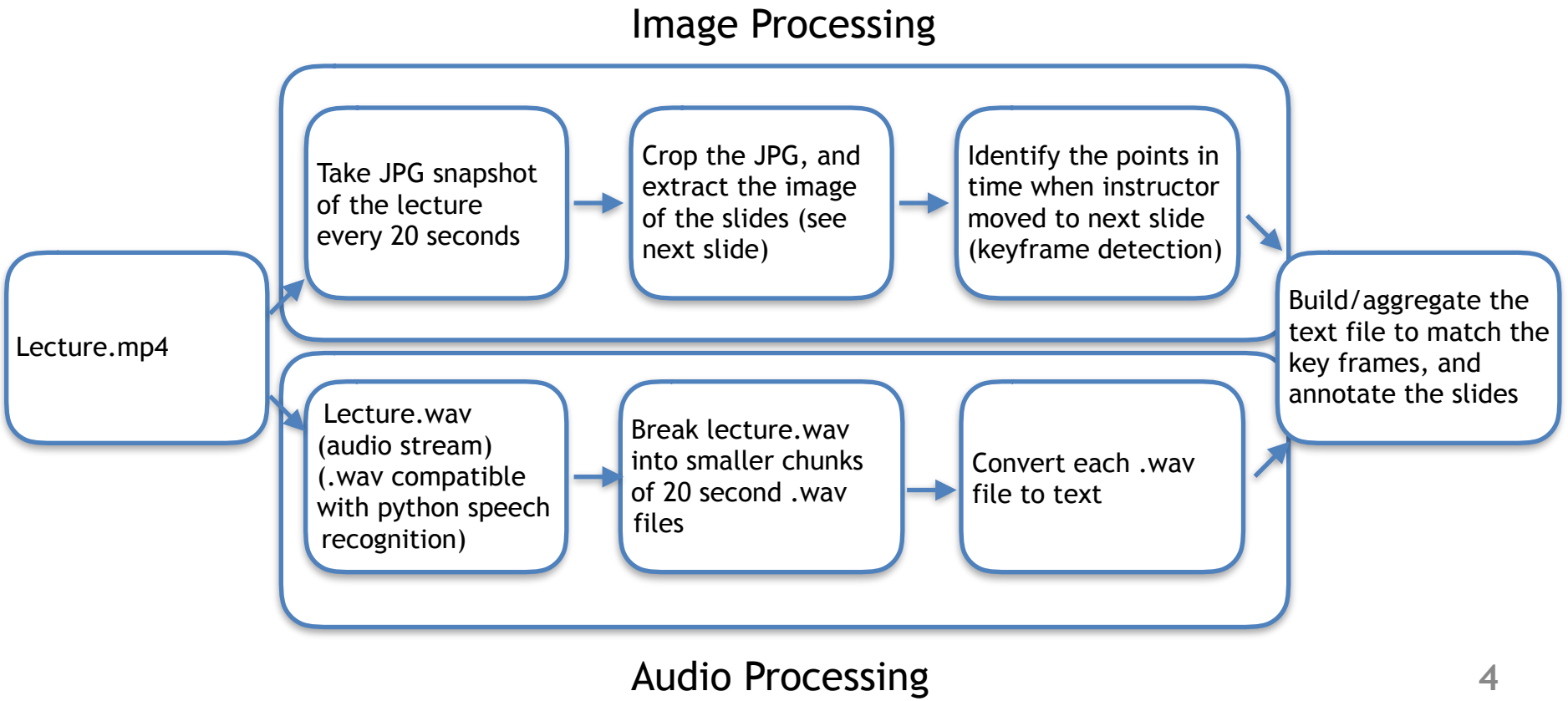
```
one thing that I wanted to point out
is that I think there's been a few
misses in terms of communication so
Piazza is where we convey as a group
and student feedback between each
other is Great is that are visible to
everyone I think it's working pretty
well for that but we haven't been
super successful in Direct
Communications on on Piazza number
post where people try to contact you
know a specific TA or they ask for
grading feedback and because there's
so much stuff on Piazza…
```

Goal

- Generate annotated lecture slides with speaker/instructor text
- Please see an example of a sample pdf here

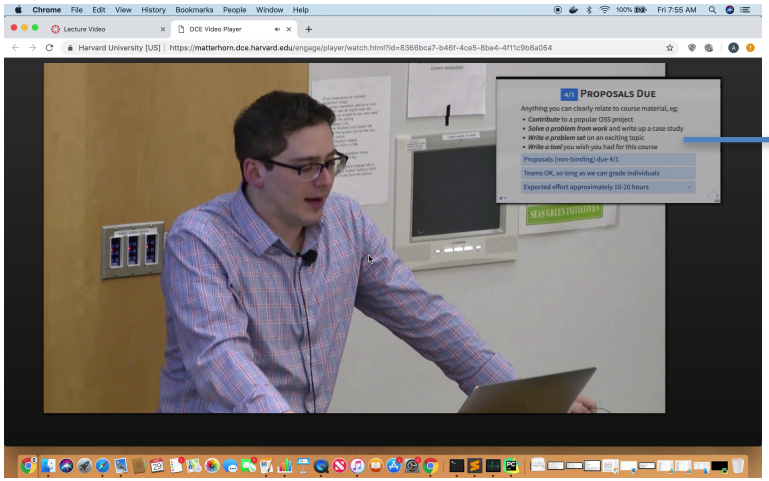
The Flow

- Convert lecture to .mp4 format
- Extract audio and video stream
- Every 20 seconds take a JPG snapshot (downsampling)
- Convert audio to text
- Build Keyframes and map text to each slide

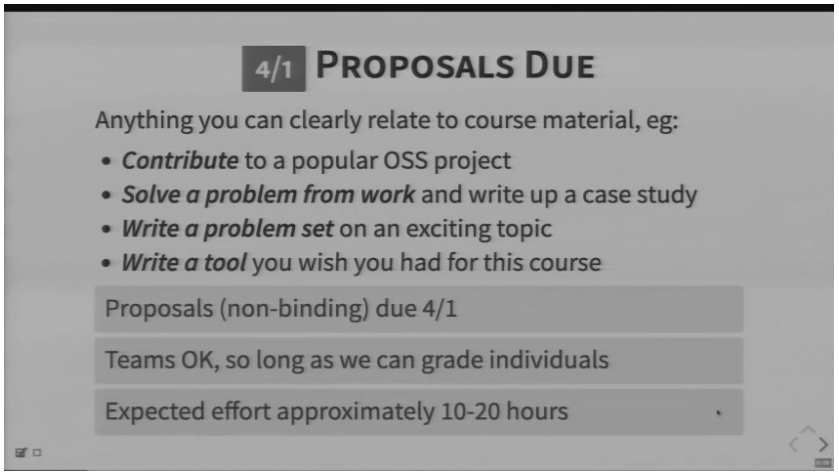


Slide Extraction

JPG snapshot



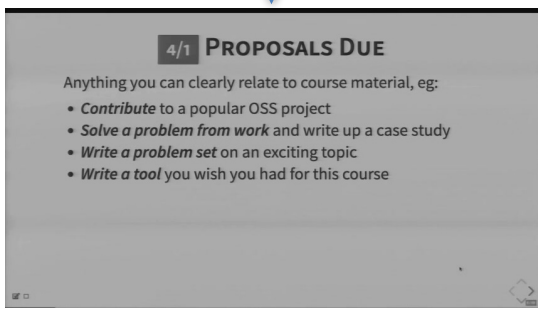
Extracted slide from the JPG snapshot



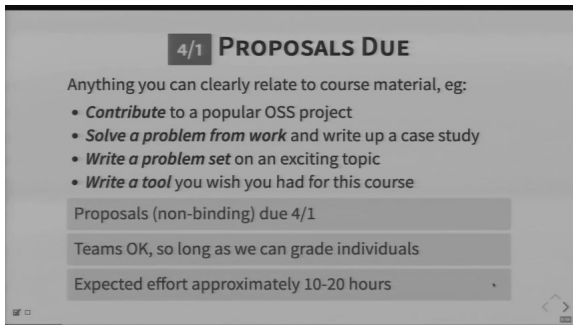
What is a KeyFrame

- A KeyFrame is where the 2 JPG snapshots are dissimilar.
- Leveraged a measure called structural similarity index (http://www.cns.nyu.edu/pub/eero/wang03-reprint.pdf) (to identify the keyframes)
- KeyFrames are essentially the various complete slides shown in the presentation

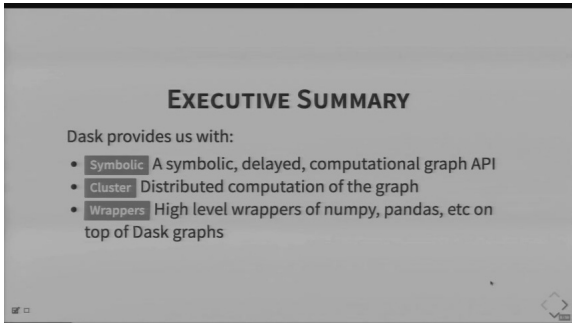
Non-KeyFrame
(it is similar to the next snapshot)



KeyFrame

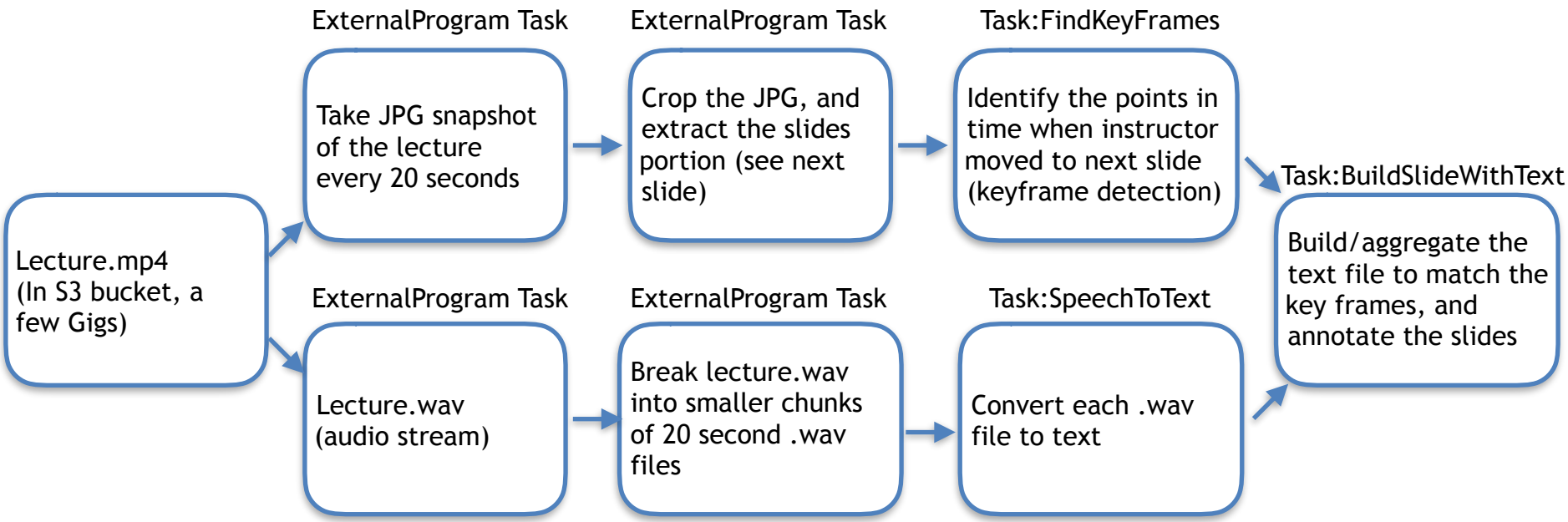


KeyFrame



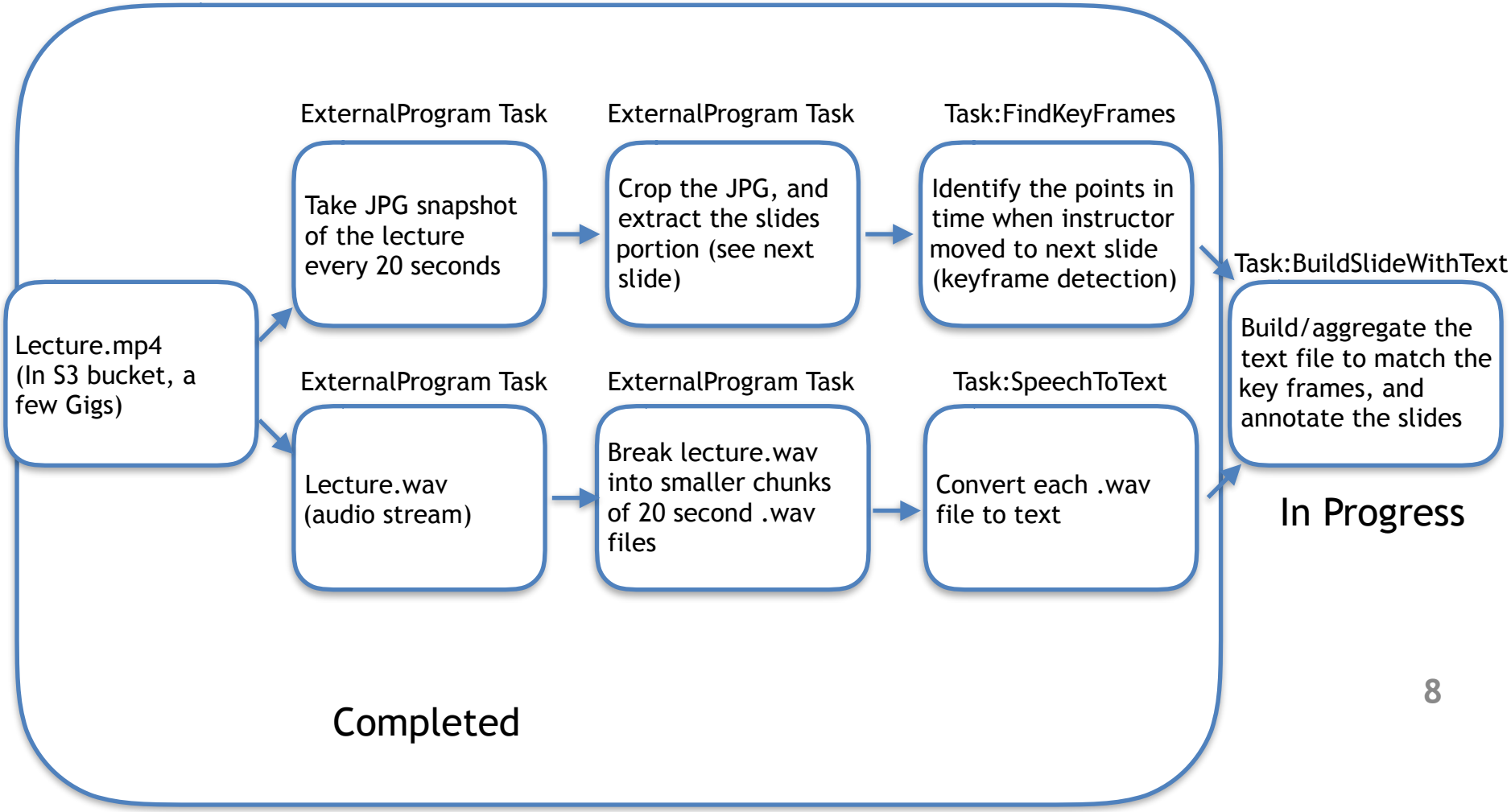
JPG snapshots in time

Mapping “The Flow” to Luigi



ExternalProgram Task - FFMPEG for audio/video extraction, and for JPG/WAV snapshots

Status - Coding Progress



Status and Areas for Improvement

- Status

- Coding for image and audio processing tasks completed with Luigi framework
- Final task (build text with slides) coding remains - requires a few iterations to get the algorithm right.

- Future areas of improvement

- The speech recognition software translates words like Dask to desk, parquet to RK. Essentially the software doesn't recognize technical jargon.
- The annotated text output in the slides needs to be improved. Right now it is a clutter.

Luigi Challenges

- Number of tasks in the flow are generating a large number (in hundreds) of JPG and WAV files.
- I had to pre-calculate the last snapshots of JPG and WAV files that are generated by FFMPEG, and use these as a dependency into the next task to stay compatible with Luigi's paradigm. (The collections concept like dask targets may have worked better for this specific use case)
- I didn't use the self.input and self.output in the code given the large number of files needed in each task. In the run method, I directly picked the files from the relevant directories.

Packages, and Code

- luigi
- scikit-learn (for ssim - structural similarity index)
- opencv-python (for image cropping and extraction)
- speechrecognition (wavfiles to text)
- cookiecutter template developed for psets was used for this project
- Code checked into https://github.com/csci-e-29/2019sp-final-project-amitathex

For any further questions

- Please feel free to reach me at amitrgupta27@gmail.com