DigiClips Media Search Engine

Introduction

Our client, DigiClips Inc., is a media content analysis company that records and extracts data from different media sources to make it searchable for their clients, who might want to locate news clips containing their name, company name, a specific topic, and more.

Problem Statement - The data currently being extracted from the television recordings is only network-provided closed captions. These captions often miss words or phrases spoken within a broadcast, and the information that was not collected is lost within hours of recordings.

Solution - This project developed efficient speech-to-text and video-to-text modules that will take television and radio recordings as its inputs and record the timestamp-location of searchable keywords and phrases of interest in these recordings.

Requirements

Functional Requirements:

- Speech-to-text must convert mono and stereo audio recordings into plain text
- Video-to-text must detect multiple fonts/styles of text on bottom half of the recording frames
- All system results must have proper grammar and spelling

Non-functional Requirements:

- System will be built without using any costly API/cloud resources
- System will be built with documentation to explain usage
- System should scale with increased quantity of data

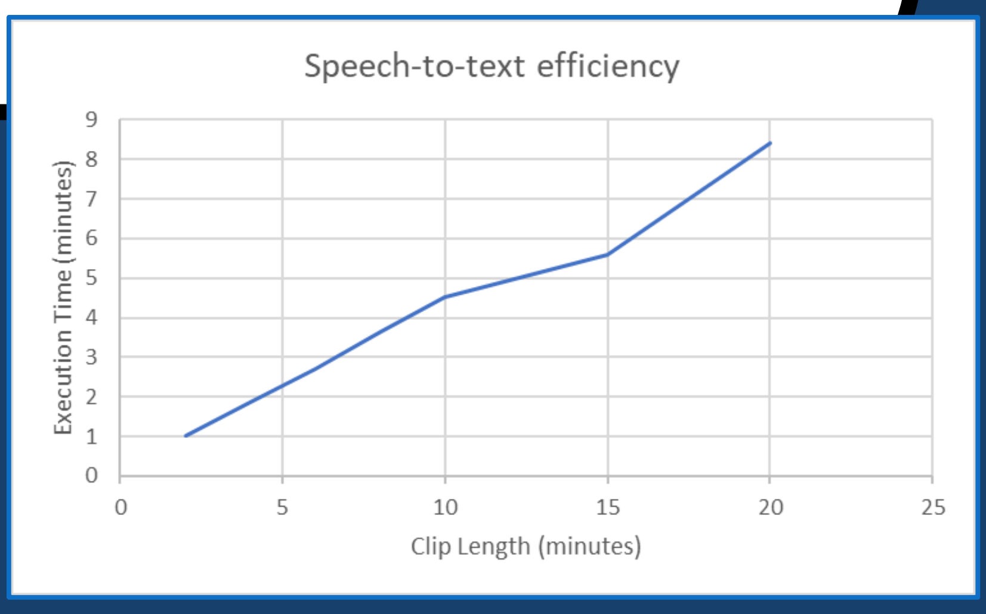
Constraints:

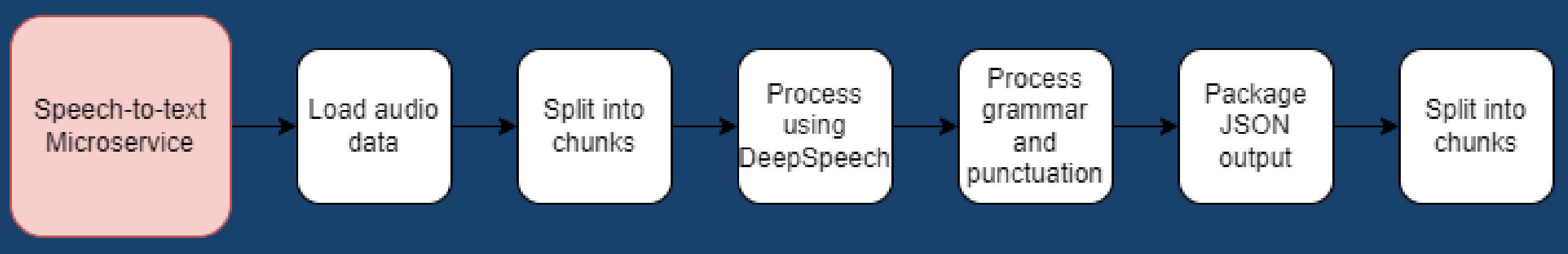
- Cannot utilize paid APIs for speech-to-text or optical character recognition
- Developed program must be able to run on an underpowered computer
- System must reliably output within the timespan of the input audio/video

Uses

This system is for use within DigiClips Inc. systems. Clients of DigiClips Inc. can use the DigiClips Media Search Engine to specifically query key words or phrases extracted from media recordings in a database.

DigiClips will use our part of the search engine to increase the amount of data available to be searched. This will help DigiClips' clients find information quickly and more accurately.

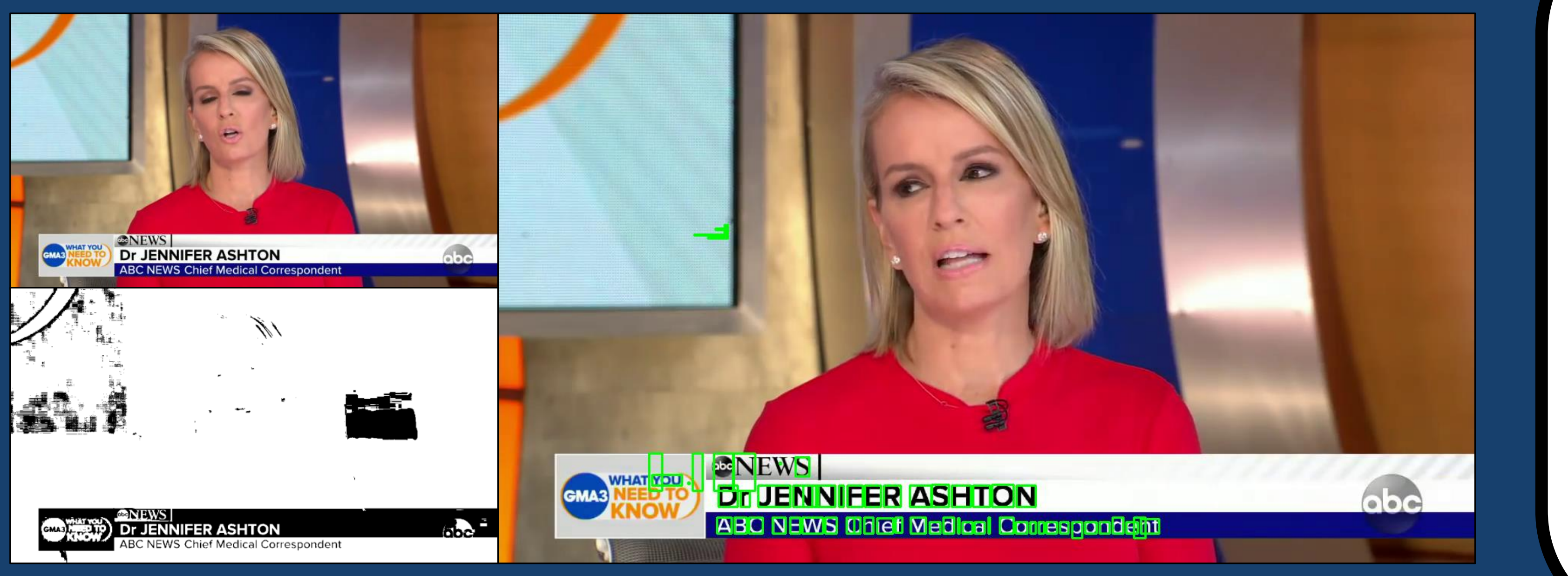




Speech-to-text Design

Tech: Python, FastAPI, Uvicorn, DeepSpeech, Docker

This service uses the Mozilla open-source project DeepSpeech to perform speech-to-text on audio and video files. Upon receiving a request, the service breaks the file into 20 second chunks and processes each chunk in parallel.

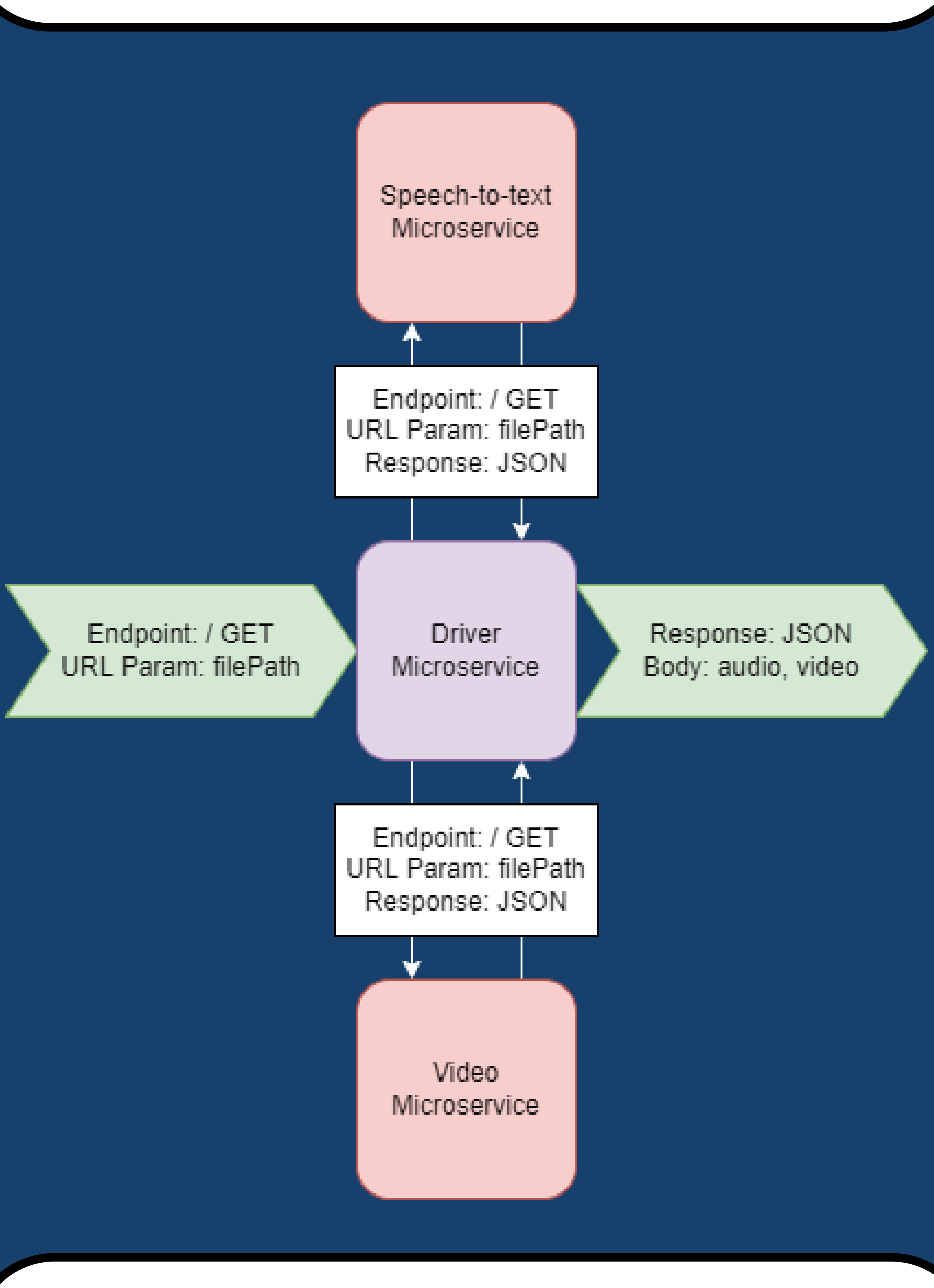


Original image (top left), result of pre-processing/morphology (bottom left), and detected text after passing through Tesseract OCR (right)

Video-to-text Design

Tech: Python, FastAPI, Uvicorn, Tesseract, OpenCV, Docker

This service uses OpenCV image morphology/pre-processing techniques alongside Google's Tesseract OCR to extract visible text from frames of TV recordings. Each recording is split into individual frames, which are then processed, timestamped, and checked for spelling and grammar errors.



Overall Design

Our project's architecture is focused on three **microservices** to perform speech-to-text and video text recognition on video files and speech-to-text on audio files.

We use Docker to simplify the setup of the services on the client's machine. Any machine with docker installed can run these three services quickly without complicated dependency installation. Docker also makes networking between services much cleaner.

Testing

Tech: Python

This service mainly uses the python library "difflib" to compare text that was generated with text that was transcribed by us. This checks for a few points of accuracy, including case sensitivity, overall length, and punctuation.

IOWA STATE UNIVERSITY
College of Electrical and Computer Engineering

Faculty Advisor: Dr. Ashfaq Khokhar
Client: DigiClips, Inc.

TEAM SDDEC21-06

Tyler Johnson
Samuel Massey
Max Van de Wille
Maxwell Wilson

Visit our team website!

http://sddec21-06.sd.ece.iastate.edu/