WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Implementation of Video and Audio to Text Converter

Dr. M. Saraswathi[1], VVSV Ronit[2], S Sai Pranav[3]

[1]Assistant Professor, Department of CSE, SCSVMV, Kanchipuram
[2]B. Tech graduate (IV year), Department of IT, SCSVMV, (Deemed to be University) Kanchipuram
[3]B. E Graduate (IV year), Department of CSE, SCSVMV (Deemed to be University), Kanchipuram

ABSTRACT:

In the real world, where the biggest workplace issues are resolved, the necessity for a video and audio to text converter exists. This converter can be used for documentation reasons by a wide range of software firms, educational institutions, and other organizations. This is mostly used by software businesses to access notes, project details, project presentation materials, etc. We choose Google Speech Recognition for our system due to its superior accuracy and user-friendly interface. Python was even chosen by us for its ease of learning. When there is only one audio file in the system, the audio to text converter should be used to make it simple for the user to convert the audio to text.

Keywords: Video to text, Audio to text, Python, Tkinter.

I. INTRODUCTION:

Now a days the online mode or work in online has the major part in the educational departments, jobs, and much more. As the pandemic occurred few year where the world has used to sit in their respective indoor with the personal computers, mobile devices, laptops, etc.If we interview participant for various research projects. Often grant funding for such projects will cover transcription costs. Human transcribers remain the work and usually do an excellent job. You could of course transcribe your own interviews, works but this can be a very time consuming and laborious task.Some qualitative researchers also advocate transcribing your own work as away of becoming more familiar with the data. We will start by converting the input video and audio file which we can convert it into text.

Objective:

By using this application we can ease the documentation problem and get the notes from the audio and video files. We are using openCV with the given input video as a source to get the frame rate and pytesseract module to extract the text present in that frame and then adding it to the output textbox. Speech recognition for the audio file.

II. Literature Survey:

TITLE	AUTHOR	FINDINGS
From Speech-to-Speech Translation to Automatic Dubbing	Marcello Federico, Robert Enyedi, Roberto Barra-Chicote, Ritwik Giri, Umut Isik, Arvindh Krishnaswamy, Hassan Sawaf	In this related work, the main goal is to evaluate the naturalness of automatic speech dubbing after enhancing a baseline speech-to-speech translation system with the possibility to control the verbosity of the translation output, to segment and synchronize the target words with the speech-pause structure of the source utterances, and to enrich TTS speech with ambient noise and reverberation extracted from the original audio.

Personalized Speech Translation using Google Speech API and Microsoft Translation API	Sagar Nimbalkar, Tekendra Baghele, Shaifullah Quraishi, Sayali Mahalle, Monali Junghare	In this work, the author is translating speech from one language to another in an efficient manner. This process is carried out in three steps with the help of two APIs. Those APIs are Google speech API and Microsoft Translation API. The Google speech API converts the speech into text format which is feed to Microsoft Translation API that translates text into the desired language.
Synchronized Audio-Visual Frames with Fractional Positional Encoding for Transformers in Video-to-Text Translation	Philipp Harzig, Moritz Einfalt, Rainer Lienhart	In this work, the authors presented a Transformer-based Video-to-Text architecture aimed to generate descriptions for short videos and able to gradually improve a vanilla Transformer designed for Machine Translation into a architecture that generates appropriate and matching captions for video clips.
Textless Speech-to-Speech Translation on Real Data	Ann Lee, Hongyu Gong, Paul-Ambroise Duquenne, Holger Schwenk, Peng-Jen Chen,SravyaPopuri,Juan Pino ChanghanWang, , Jiatao Gu, Wei-Ning Hsu	In this work, the authors used reduces variations in the target speech while retaining the lexical content and take advantage of self-superviseddiscrete representations of a reference speaker speech and perform CTC fine-tuning with a pre-trained speech encoder.

Problem statement:

There are many systems like which just converts the audio neither the video into the text. For many transcribing work there are many live transcribing applications where we see, use and exists in the day to day work and work profiles. As there is the problem in this existing system some time it may goes the pronunciation which is wrong and is done with only the audio transcribing.The accuracy comes the problem when the video to text converting part, where we try to get the accuracy for the audio and its done with the thing to get the output .Whereas the video goes frame by frame in which it takes the time to get the output slow.

III. Proposed System:

Here we are proposing a new video and audio to text converter that work with the accuracy by the tkinter's GUI. The uses in this proposed system are: Simple execution, Interactive GUI, Working amount of accuracy etc…

Modules Description:

In this system, we have developed few modules such as

- Video to audio conversion
- Speech recognition
- Image recognition
- Gui output display

Video to audio conversion:

In this we are using the moviepy package in python in order to extract the audio of the uploaded video file and save it to a audio.wav file. We are using "vidfile.audio.write_audiofile("audio.wav")" to write the audio of the given video file into a file named audio.wav and saving it the parent directory for further use.

Speech recognition:

We are using the SpeechRecognition module in python with the extracted audio.wav file as the source and using google speech recognition to extract the text from the audio file and return it as a stringGoogle speech recognition

Image recognition:

We are using opencv with the given input video as a source and using "capture.get(cv2.CAP_PROP_FPS)" t0 get the frame rate of the video and dividing the current number of frame with the fps to get a single frame per second in the video and passing that frame to the pytesseract module to extract the text present in that frame and then adding it to the output textbox.

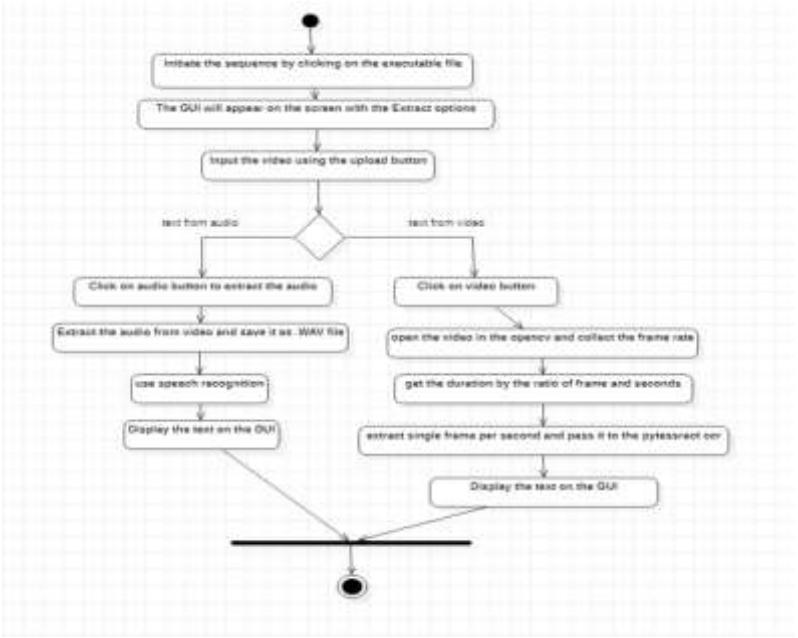
GUI output display: The gui 3 main elements the left frame, textbox , and the control buttonsthe left frame is a container to display the name of the project along with the control buttons.The text box displays the output text from either the audio or video. The control buttons are used to upload.

IV. Methodology Used

- We are using tkinter to display the gui and upload,audio and video buttons for controlling the guiWe are using first uploading the video file using fd.askopenfilename() in tkinter to get the file location.
- Then if the user presses the audio button the audio of the video file is extracted and is used as a source for google speech recognition using Speech Recognition module and the output is displayed in the text box.If the video button is pressed the text is extracted once per every second in the video and the output is added to the gui text box.

METHOD	**DESCRIPTION**
outins()	This method is used to remove the current text in the textbox and insert the given string
upload_button()	This method is used to ask the user for the video file and saves the file location to the global variable “filename”
audio_to_text()	This method is used to extract the audio from the video and use speech recognition to retrieve the text from the audio
video_to_text()	This method is used to get a frame from every second of the given video and use optical character recognition to extract the text from the frame.

Methods of video and audio toText Converter



Process:

1. The system starts by opening a tkinter window which contains a text box and a three buttons.
2. The user can start by selecting the upload button and selecting the required file in the open file window.
3. When Audio button is clicked the audio is extracted from the given video file using the moviepy module.The extracted audio is then sent to google speech recognition and the output is added to the textbox in the gui.
4. When the video button is selected it the uploaded video is used as a capture source in the opencv modules VideoCapture() method.
5. Then a while True loop is used to iterate through all the frames and break is used to exit the loop when the iterating through the frames is completed.
6. While iterating through the frames it check the if takes a frame per every second in the video and sends it to the tesseract ocr the output from the ocr is added to a list.
7. After exiting the loop the list is then formatted into a string with a tabular appearance. The string is then inserted into the textbox.

V. Result:

The result of our proposed work is shown is in fig 1 to 3

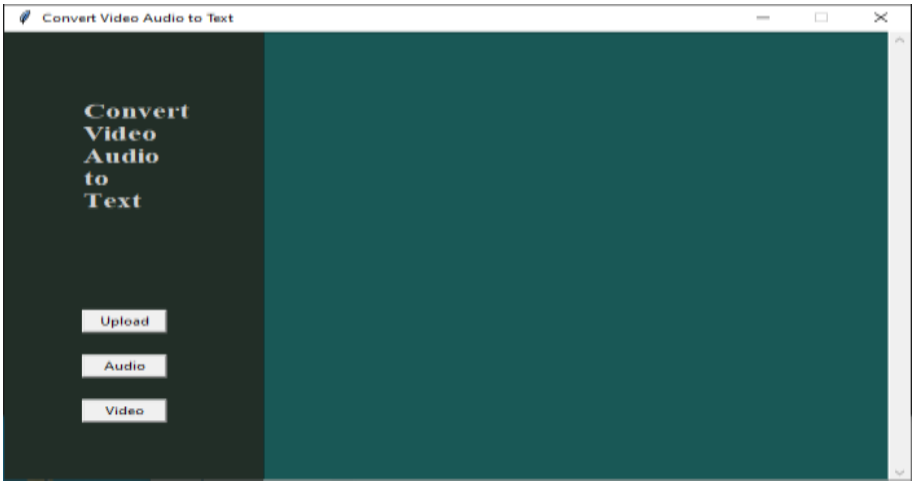


Fig 1:Appearing of the gui of the project:

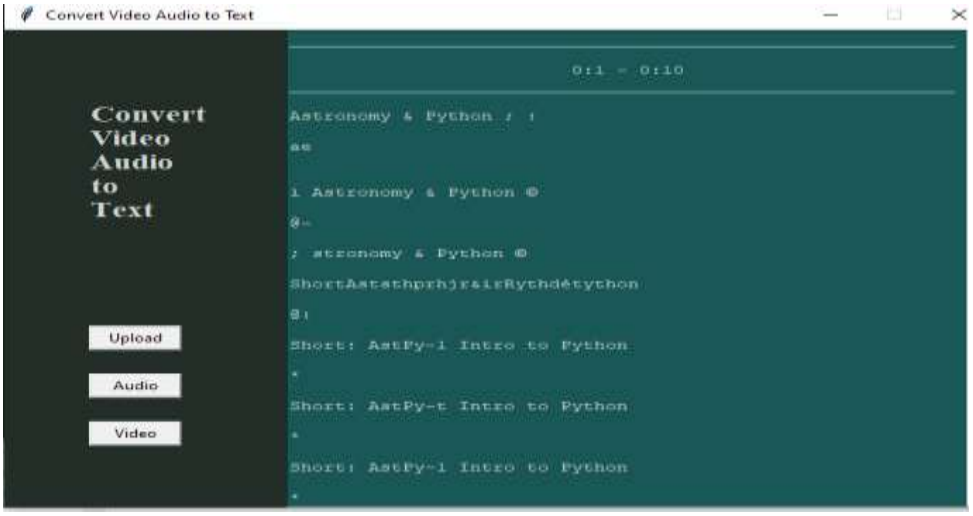


Fig 2:Result of the video to text:

Fig 3: Result of the audio to text:

Conclusion:

This program offers a very simple, barely functioning sample. This might be improved in a number of ways, including by giving users the option to load either an mp4 or a wav file, by letting them select from a variety of speech recognizers, and by showing more details like file size and length. The user experience of simple applications can be enhanced with graphical user interfaces, which are simple to add in Python using tools like PyQt and Tkinter Designer. Performance issues can also be resolved using threading and running computationally intensive tasks in the background to minimise their impact on the user experience.

References:

[1]. Marcello Federico, Robert Enyedi, Roberto Barra-Chicote, Ritwik Giri, Umut Isik, Arvindh Krishnaswamy, Hassan Sawaf" From Speech-to-Speech Translation to Automatic Dubbing" Proceedings of the 17th International Conference on Spoken Language Translation July 2020

[2]. GOPA - International Energy Consultant INTEC & Hamm-Lippstadt University of Applied Sciences 2022

[3]. Published in: IEEE Journal of Selected Topics in Signal Processing (Volume: 16, Issue: 6, October 2022)

[4]. Published in: IEEE Transactions on Software Engineering (Volume: 48, Issue: 1, 01 January 2022)

[5]. J. Pradeep, E. Srinivasan, S. Himavathi, Neural network based handwritten character recognition system without feature extraction, in 2011 International Conference on Computer, Communication and Electrical Technology (ICCCET), pp. 40–44 (2011).

[6]. R. Mittal, A. Garg, Text extraction using OCR: A systematic review, in 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), pp. 357–362 (2020).