

Design And Implementation Of Automatic Content Delivery System For Virtual Classes

Miss. Domakonda Kavya¹, Dr V Mahendra²

PG Scholar¹, Professor²

Department of Electronics and communication Engineering^{1,2}, MLR Institute of Technology, Hyderabad, Telangana^{1,2,3}.

E-mail: kavyabalu59@gmail.com , mahendra.v@mlrinstitutions.ac.in

Abstract: - The time-consuming, traditional approach of teaching with chalk and boards has recently given way to smart classroom instruction that uses digital gadgets such as computers, speakers, projectors, and many others. These tools enhance student-teacher engagement, but most institutions are still hesitant to implement digital technology because of the expense. Additionally, the traditional approach of recording classes necessitates that teachers participate and be available at the computer to browse through the slides, taking time away from the scheduled time slot. The major goal of this work is to develop and deploy a low-cost technology-assisted approach that improves service teaching by making efficient use of instructional time. The suggested system consists of Raspberry Pi, a low-cost device with a single controller. The goal of this study is to create a virtual education system to improve on-campus learning. Even though virtual classrooms and online learning are becoming more and more popular, several issues still need to be resolved, such as the restricted audio and video bandwidth, poor strength and durability, poor collaborative work, and limited evaluation during virtual classroom sessions. The design and execution of a classroom - based system that emphasizes cooperative learning involving students and instructors at distant places are shown in this project. To display the PPT material, the system was created and hosted online using the flask web server.

This PPT uses a Raspberry Pi 4 to play back the information and show it. We loaded every software such as text to speech, Image extraction and also automatic video play option with audio background according to our requirement. The Raspberry Pi loaded Python software to complete this operation. Python is a dynamically semantic, interpreter-based, object-oriented, high-level programming language. Python's straight forward syntax prioritizes readability and makes it simple to learn, which lowers the cost of programmed maintenance.

Newest Slightly elevated Quad-Core 64-bit Qualcomm 2711, Helix A72 processor capable at 1.5GHz, is installed in the Pi 4 B. This is intended to perform 90% better or use 20% cheaper power than the previous model. With brand-new 1GB/2GB as well as 4GB LPDDR4 SDRAM versions, Dual-band optimized features and 5GHz, IEEE 802b/g/n/ac wireless LAN, and PoE functionality through a separate PoE HAT, the Pi4's hardware has been upgraded for quicker performance.

I. INTRODUCTION

Text-to-speech (TTS) software creates a spoken audio version of something like the text, such as that found in a file format or a PPT. TTS can be used to enhance text message understanding or to make it simpler for someone who is blind to understand information on a computer display. A method called text to speech

relies on the automated creation of new

paragraphs or words. This technique enables the creation of a speaking computer. The appropriate and suitable of words is frequently sought for using dictionaries, yet for some terms, the dictionaries are unable to provide the proper pronunciation. The dictionary is thus solely restricted to a specific selection of terms. The suggested system is only capable of converting some of the user's desired audio files since the dictionary can only include a certain set of words,

making it impossible to cover the complete word string.

Speech synthesis is brought up when the study covers text to speech conversion. Artificial or computer-generated human voice is referred to as speech synthesis. Speech synthesizer is the name of a system that is employed for this purpose. For the Malay language used in Malaysia, a rule-based text-to-speech generation system was created [1]. The SMaTTS speech synthesis system was created with an intuitive graphical user interface. (GUI). The SMaTTS system's algorithms were created and evaluated. For a thorough examination, categorical estimation (CE) is used to analyze the system's overall performance.

Another voice synthesizer that is based on concatenating speech fragments was created for both the Macedonian language [2]. Concatenation approaches, in combination with a minimum amount of signal processing, have been able to overcome the difficulty of acquiring extra-linguistic information. However, it is still problematic when the speaker uses different prosodies. Glottal inverse filtering can also be used in voice synthesis to produce natural-sounding speech [3].

One method of doing speech synthesis is Unit Selection Synthesis, which offers greater naturalness and effectiveness depending on the unit utilized, such as a phoneme, diaphone, or syllable [5]. Thai text was turned into speech using a text-to-speech method (TTTS). Thai text, text processing, dictionary lookup, and voice synthesis are the four components that make up TTTS. 5300 Thai words and their sound waves are included in the TTTS dictionary [6]. Text to sound conversion may also be done with neural networks. Many of the idiosyncrasies in Greek pronunciation can be absorbed by the use of neural networks [8]. Chinese language analysis is based on an examination of Tibetan language traits [9]. For this, the SAMPA ST standard pronunciation is utilized. The development of a text-to-speech system language Azerbaijani Turkish illustrates several speech synthesis techniques as well as some crucial aspects of the language. This

system employs corpus-based concatenative speech synthesis. The proper units are chosen and then concatenated [10] based on the text input.

Speech is converted into text using autonomous speech recognition (ASR) [4]. ASR is based on the notion of speech coding and simulates some spectrum analyses carried out by the ear. Voice synthesis is the process of creating a speech waveform automatically, usually from input text. TTS uses a database of information, including text and speech. The database stores previously analyzed speech in tiny pieces so that it may be concatenated in the right order during runtime.

To create the phonetic transcription, TTS systems first process text, including "letter-to-sound" conversion. The quality of the outputs is enhanced when the database contains words or phrases. A synthesizer may also mimic human speech traits to produce a synthetic voice that sounds fairly real. A speech synthesizer is mostly assessed in terms of how closely its output speech resembles that of a human [5]. Future days the text-to-speech (TTS) technology places more emphasis on the emotional resonance of the speech, necessitating a more genuine, realistic voice that may be closely compared to the human voice. Thus, a restriction on the input text is necessary to obtain output of a high caliber. Multimodal messaging and voice-enabled e-mail are two instances of speech synthesis. The user speaks a word into a microphone as the initial stage in speech synthesis, after which the voice is converted from analogue to digital (A/D) and stored in memory as a database.

The computer compares this sound to a voice sample with a known meaning in order to recognize it. Another instance of voice control for any device rather than a keyboard or mouse is now available. Converting text to voice essentially consists of two parts: the front end and the back end. Apart from this text to speech, Image extraction from the running slides and also video automatic playing is implemented. Text ability to differentiate, Text to Speech conversion, and other language-related

processing are done at the front end [2]. Pitch estimation, voice synthesis, and other speech-related signal processing are part of the back-end process. The backend processes vary because of the many synthesizers used depending on the output the user wants, whereas the front end procedures often stay the same. Text tokenization is the process of separating words in a string by spaces, commas, etc. Text normalization is essentially the process of changing letters, numbers, and abbreviations into their equivalent nouns, such as Mr., Mrs., Dr., etc. In addition to this, front-end stages include labeling speech fragments, phrasing, pause insertion, etc. Tone modulator and the creation of speech using an algorithm make up the last phase. The entire text-to-speech conversion process is carried out on a Raspberry Pi, which has an ARM CPU and so produces highly effective and precise results.

II. LITERATURE SURVEY

Many interactive learning systems have flaws, including live broadcasting, the inability to respond to conversations, and the management of assignments. The virtual classroom system described in [11] enhances communication between both students and teachers and is built on an intranet. The system adopts a fresh approach to e-learning, addresses a number of problems with the existing ones, and encourages teacher-student interaction that mimics a conventional classroom environment. The technology permits the submission of coursework, questions, and answers as well as the live transmission of lectures.

The authors in [12] presented a web-based learning system with multiple categories for live broadcast lectures, whiteboard technology, chat rooms, and numerous instructor and student feedback modules. The system is a state-of-the-art synthesis of synchronous learning, streaming media video, and online chat. It is mixed with group tasks, asynchronous internet communication and scenarios that are quite similar to real-world settings. The lack of email plus short message (SMS) announcements for mobile devices is one of the system's biggest flaws.

[13] suggests a virtual picture production-based interactive learning system. The system consists of three classrooms. The first instruction is for theory and comprises video clips with instructions for lecture and practice as well as online live lectures, whereas the second classroom is a library that connects the ch websites with radiographic testing. The final and most important classroom functions as a lab wherever students may observe the results of experiments. All three classrooms were produced as virtual places using computer graphics. Characters that resemble students have also been created in order to promote interaction between them and virtual classrooms.

The authors in [14] presented an architectural proposal for an embedded design for the delivery of educational lectures in a setting akin to a virtual environment. The architectural framework and a specification of the system's component pieces are provided along with methods and ideas for putting the suggested system into practice. In order to support database middleware, the architecture features multi-tiered, scalable, and meant to be adaptable. The whole functionality of the program is offered by online services, communicates using the industry-standard XML protocol, and can only be accessed from a web browser. The software that the virtual interactions with teachers employs has the capacity to provide users with simple and standardized user experiences in accordance with the tiers for online services, knowledge, and from prior to the hypertext. The sensor design method categorizes application objects into three logical groups: those that negotiate with the presentation of the application, some that interact with data and industry best practices, and those that don't just accept but rather comprehend user requests and instruct the functionality to carry them out.

The creators of [15] utilized Real-Time Consistent Reporting (RTMP), a sophisticated Net filter world wide web protocol that enables the efficient exchange of messages, synchronized data, audio, and video. The system enables instructors and students to publish

streaming radio, video, and certain other data signals in order to provide streaming content with navigation or interactivity. The technology enables quick online dialogue between students and teachers, creating the impression that you are in a physical classroom. In order to enable instructors and students to exchange resources outside of the classroom, it is also recommended to use a content system altogether as an add-on portion to the educational system.

The recommended web-based integrated learning system in [16] is based on e-pedagogy and has increased participant collaboration capabilities in a web-based educational system (instructors and students). The teacher, course, student, and teaching students performance made up the key structural components of the system model. Its features, which were all utilized to promote students to participate in experiential learning, were the course outline, dissemination, materials, evaluation poll, chat, eyes, students' lists, lectures, and cooperate.

The authors of [17] outlined the conception and implementation of a learning program created to improve wireless technology for constructing an ad-hoc classroom in order to give a modern and creative learning environment. A wireless solution has been created that enable instructors and students to dynamically establish a classroom irrespective of geographic or time limitations. Information technology was created to support conventional classrooms by providing teachers with instructional resources. The equipment in an open mike classroom includes a blackboard, board rubber, colored chalk, camera, sound and video recorders, than for lectures and discussions. Additionally, the youngster received a "e Schoolbag," an electronic school bag. This bag had electronic copies of a book, a notepad, the phone number of the parents, some papers, a calculator, a pencil case, and other materials.

Outlines the crucial steps for developing and implementing a successful interactive learning idea in [18]. It was clearly stated how to assess, design, create, carry out, and manage a virtual classroom in 10 easy stages, with a warning that failing to do so might eventually

lead to subpar, insufficient, and ineffective solutions as Internet-based distance education. [19] explains how the authors built a streaming media development and learning theories-based online classroom framework. The two primary elements of the system are the Instructional Digital communication Environment (ICE) and the Blended Learning Environment (CLE). Among other things, ICE provides learners with instructional materials, lecture videos, also an interactive setting. CLE encourages active learning by providing the setting with educational tools, materials, and chances for students to participate in pertinent conversation. Synchronous event-based planning strategies and e-learning best practices were used to develop the environments.

III. PROPOSED SYSTEM

This paper has represented the innovative idea as well as low cost technique that are used to hear the contents of the text image without reading them. It combines the concept of optical character recognition (OCR) and Text to speech synthesizer (TTS) as well as image extraction and automatic video playing in raspberry pi. Text extraction from color image is very difficult task for computer the text to speech conversion system is read the English alphabets and numbers that are in the image using the OCR technique and convert it into the voice format. This paper has presented the simple approach for text extraction and its conversion into speech, image extraction from presentation and also it supports automatic video playing, the testing of devices was done on raspberry pi module. Text to speech (TTS) system produces the more natural voice that can be closely matched with human voice. The first step of speech synthesis is for the uses to speak a word in microphones and then that speech is converted into the digital format by using analog to digital converter and stored in memory, image extraction and auto video playing will be done using Flask web server as shown.

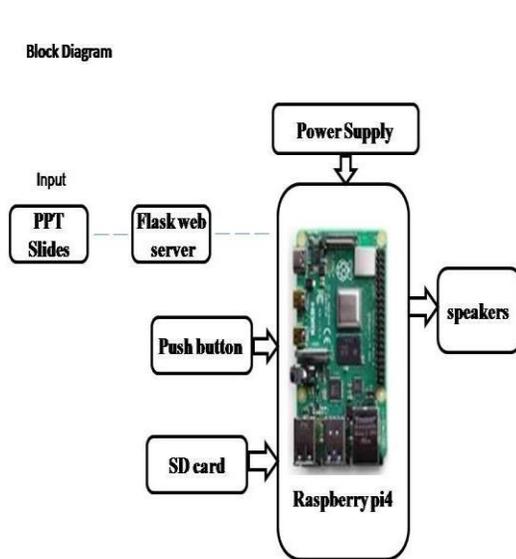


Fig 1.1 Proposed Block Diagram

The system was developed and hosted on the web using flask web server to demonstrate the content of PPT. This PPT running back of raspberry pi4 and demonstrate that content through speaker. To achieve this task Raspberry pi loaded program written in python language. The major components used in this project is

- ✓ Adapter power supply.
- ✓ Raspberry Pi4 Processor.
- ✓ SD Card.
- ✓ Push button.
- ✓ Speakers.

A. Raspberry Pi

The Raspberry Pi is a little computer that lacks a hard drive like a PC but instead uses an SD card with 4GB or more of storage space instead. This little computer has a wide range of interactive applications, including playing 3D games and high-quality audio and video [4]. Figure 1.2 depicts the Raspberry-pi 4 Model B.



Fig 1.2 Raspberry pi

As seen in Fig. The Raspberry Pi 4 Model B has four ports for connecting peripherals, including a mouse, keyboard, webcam, and other devices. It contains a single HDMI connector for connecting from a TV or a monitor and a single Ethernet connection for internet connectivity. It contains one audio port and one switch Mode power port for the Raspberry. There is an RCA video port available for connecting older TV models. In addition, it has GPIO headers for connecting additional hardware components, such as LEDs, motors, and so forth. The project that is being suggested uses GPIO ports.

The GPIO pins on the Raspberry Pi must be initialized since the keyboard is attached to it. The Raspberry Pi has various benefits for text-to-speech conversion, including: It offers a number of voice synthesis solutions, including eSpeak, Festivals, fite, and others. The Raspberry Pi uses only 5-7 Watts of electricity at most. It includes extension options, like GPIO, camera, etc. The HDMI display connector may support resolutions up to 1920x1200, making it useful for more applications in the future.

B. Flask Web

Python's Flask API enables us to create web apps. Armin Ronacher was the person who created it. Because it requires less base code to construct a straightforward web application, Flask's framework is more clear than Django's architecture and is also simpler to understand. A web framework, also known as a web application framework, is a group of modules and packages that enable programmers to create applications without having to write low-level code for things

like thread management and protocols. Flask is built using the Jinja2 template engine and the WSGI (Server Is a server Gateway Interface) tools.

Starting With Flask:

The installation of Flask requires Python 2.6 or above. Starting in any Python IDE, import Flask out from flask package. You may click the installation link provided below to install on any environment.

Check out the code below to see if the operation is operating as intended.

```
# a WSGI application object imported from flask app in Flask is same to Flask(name) #
Constructor for flask
```

```
A decorator is used to inform the application of the URL that corresponds to the @app function.
hello() route('/') def:
send back "HELLO"
```

If `_name == 'main_'`, then `app.run()` will exit. The function `hello()` is connected to the URL. The result of this procedure will be presented in line with the webserver's home page when it is accessed in the browser.

Calling the `run()` method initiates the Flask application.

Any changes to the code need a manual restart of the function. To avoid this, debug support is turned on in order to trace any errors.

Routing: Today's web frameworks include this feature so that users may easily remember URLs. Directly to the web page rather than going through the Home page is helpful. The accompanying `route()` decorator is used to link the URL to just a function.

```
# Route URL decorator: @app.route('/hello')
binding to the route def hello world()
function, which should return "hello world"
When a user accesses the
URL http://localhost:5000/hello, the
browser will display the results of the hello
world() function. It is also possible to connect a
URL with a function, like in the example above,
by using the add url rule() function about an
```

application object.

Define `hello world()` to return the app "Hello World."

"/", "hello," "hello world," add url rule Using Numbers in Flask: By adding

```
your variable components to the rule argument, the variable portions in the flask are utilised to generate a URL dynamically. This portion is marked as changeable. The keyword parameter is given. see the illustration below from import flask app in Flask is same to Flask(name) the decorator function's route
hello name name
@s@app.route('/hello/<name>')
hello name(name) def 'Hello%s!'
return name if "__main__" is
the name_,
then"app.run(debug =
True)"
```

Run the aforementioned example from power shell and save it as `hello.py`. Next, launch a browser and type `http://localhost:5000/hello/GeeksforGeeks` into the address bar.

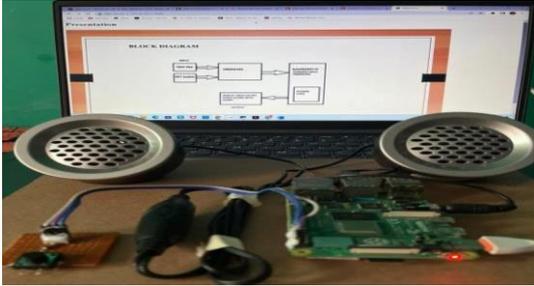
Output: Good day, GeeksforGeeks!

In the aforementioned illustration, the `route()` decorator's parameter takes as an argument a variable that is associated to the URL `"/hello."` The `hello()` method will get the parameter "Geeks for Geeks" if the URL is supplied

as
`http://localhost:5000/hello/GeeksforGeeks.`

Other input parameters like `int`, `decimal`, and `path` (for directories separator channel that may also accept slash) are also utilized addition to the typical string variable component. The routing module of Werkzeug is the foundation for Flask's URL rules. This makes sure that the URLs created are original and follow the rules established by Apache.

Examples: `from import flask app in Flask is same to Flask(name) @app.route('/blog/<postID>') return 'Blog Number%d%' postID from the`



```
function show_blog(postID)
@app.route('/rev/<revNo>')
return "Revision Number%f" from
defrevision(revNo) revNo
App.run if __name__ is "__main__" ()
Say the website address
is http://localhost:5000/blog/555.
```

Output:

Blog Post No. 555

```
# Do you enter this url into your browser?
http://localhost:5000/rev/1.1
Number of Revisions: 1.100000
```

Building a dynamic URL with Flask Theurl for() function is used to construct theHttp for a certain function. The first parameter to the function is the function name, and it also takes one or more adjective arguments. see this illustration from import flask Redirect, Flask, and url for app = Flask(name_)

```
The decorator that route(argument)
function def hello admin() is
@app.route('/admin'): 'Hello Admin' call
returned by #binding to hello admin
@app.route('/guest/<guest>')
```

```
Define Hello Guest as Guest. "Hello%s as
Guest"% guest #binding to hello guest call return
@app.route('/user/<name>')
hello user(name) def
if name equals "admin" URL dynamically bound
to the function return redirect(url for('hello
admin'))
```

If not, url for ('hello guest', guest = name) should produce a redirect.

```
if "__main__" is the_name_, then "app.run(debug =
True)"
```

Save the code above, execute it using the Python

shell, then open a browser and type the following URL to test this:

Please enter:

```
http://localhost:5000/user/admin output:
```

Good day, admin

User: ABC at localhost:5000

Hello, ABC as a guest.

The user(name) function in the code above takes values through URL input. It determines whether the argument received matches the "admin" argument or not. The method hello admin() is executed if it matches; else, hello guest() is invoked.

```
pi@raspberrypi: ~/Documents/rpiWebServer
pi@raspberrypi:~/Documents/rpiWebServer $ ifconfig
eth0: flags=4099<UP, BROADCAST, MULTICAST> mtu 1500
    ether b8:27:eb:10:da:5f txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP, LOOPBACK, RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1 (Local Loopback)
    RX packets 231832 bytes 40189187 (38.3 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 231833 bytes 40189187 (38.3 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP, BROADCAST, RUNNING, MULTICAST> mtu 1500
    inet 10.0.1.27 netmask 255.255.255.0 broadcast 10.0.1.255
    inet6 fe80::8f1a:c0ba:3079:709e prefixlen 64 scopeid 0x20<link>
    ether b8:27:eb:4d:8f:0a txqueuelen 1000 (Ethernet)
    RX packets 843888 bytes 108830029 (103.7 MiB)
    RX errors 0 dropped 467 overruns 0 frame 0
    TX packets 808367 bytes 579127740 (552.2 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

pi@raspberrypi:~/Documents/rpiWebServer $ sudo python3 helloWorld.py
* Running on http://0.0.0.0:80/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger pin code: 355-015-716
10.0.1.27 - - [04/Jan/2018 15:10:48] "GET / HTTP/1.1" 200 -
10.0.1.10 - - [04/Jan/2018 16:14:12] "GET / HTTP/1.1" 200 -
```

Fig 1.3 Web Server activation

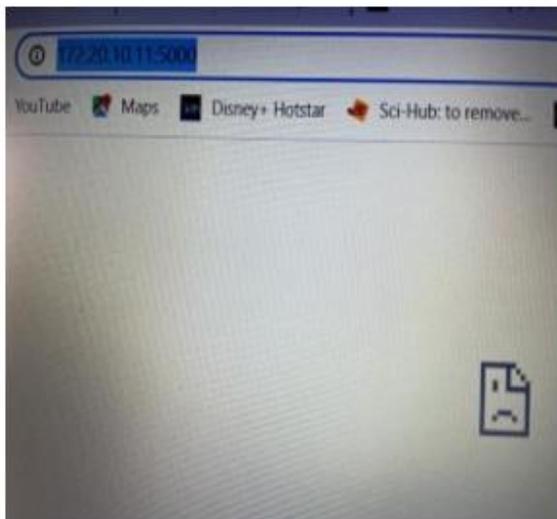
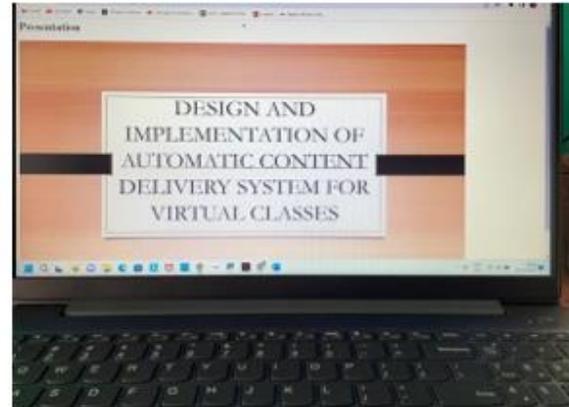
The bottom of the image above has twolines, as you can see. The lines demonstrate that our server responded with HTTP statuscode 200, or "OK," after receiving requests from two separate web browsers for the rootURL. On both my Mac and the RPi itself (10.1.0.27), I supplied our RPi Serveraddress (10.1.0.10). As long as theprogramme is running, a new line will show up on the terminal for each new request.

IV. RESULTS AND DISCUSSION

An automatic content delivery system using raspberry pi and flask web server is implemented successfully. A content/topic can be presented in virtual classes using our voice over video in theabsence of Lecturer/speaker as shown in figure 2.1.

Fig 2.1 Voice over video using raspberry pi

Instead of connecting to laptops or display screen with HDMI cable, An IOT technology is being used to connect to laptop or screen. Connect laptop to the mobile hotspot, which detects the raspberry pi IP address through network analyzer app as shown in figure 2.2. In any browser, ping the IP address of raspberry pi which is shown in figure 2.3 and after entering the IP address, a voice-over-video starts to play as shown in figure 2.4. A Reset button to connect to raspberry pi to restart the presentation so that we can replayit anytime

**Fig 2.2 Network Analyzer****Fig 2.3 IP Address of Raspberry Pi****Fig 2.4 Voice over presentation**

Statistical Analysis

A thorough analysis of the project has been done using different inputs and have collected the outputs for the respective data. The time taken to play the voice over video with different inputs have been measured in this analysis.

Different size text files (1kb, 2kb, 4kb) and PPT files (2.709mb, 2.718mb, 2.736mb) have been given as an input and the output is measured in time. The statistical graph is shown in table 1.1.

Text file size	PPT file Size	No of slides	Time Taken
1KB	2.709MB	4	1.09MIN
1.5KB	2.711MB	10	2.72MIN
2KB	2.718MB	16	4.36MIN
2.5KB	2.723MB	20	5.45MIN
3KB	2.736MB	32	8.75MIN

Table 1.1 Estimation Time of Voice over video generation

As per the above table, time taken by the voice over video is directly proportional to size of the files. As the size and no of slides increases, the time taken for audio-video playing increases.

V.CONCLUSION

The intended purpose of the proposed Text to Speaking (TTS) system, Image extraction and auto video Playing was to create an analogous acoustic signal that synchronized with the text inputted into the Raspberry-Pi system. Presence of every module has been reasoned out and placed carefully, thus contributing to the best working of the unit. Secondly, using highly advanced IC's with the help of growing technology, the project has been successfully implemented. The database connectivity, the audio amplifiers, and the GPIO lines were all part of the programming plan. This experimental validation may also be used to more complicate conjoined words and sounds. Interms of voice synthesis and feature extraction, the Raspberry-potential Pi's is essentially limitless. Integrating features of all the hardware components used have been developed in it and a statistical analysis of the voice over video generation is provided. According to the table given in Results and Discussion section, the analysis has been done in terms of various input sizes vs time consumed by the video and outcome of the analysis shows that the time taken for the audio video playing is directly proportional to file size given. Thus the project has been successfully designed and tested.

References

- [1] Rebecca F. Bruce, J. Dean Brock, Susan L. Reiser. Make Space for the Pi. Proceedings of the IEEE SoutheastCon, 2015.
- [2] Barbadekar Ashwini, Iadhav Shivaji Digambarrao and Prof.(Dr) S.P Patil. Performance Analysis of Finger Print Sensors. 2nd International Conference on Mechanical and Electronics Engineering (ICMEE), 2010.
- [3] Dhaval Chheda et al. Smart Projectors using Remote Controlled Raspberry Pi. International Journal of Computer Applications, November 2013.
- [4] L.Arunkumar and A.Arun raja. Biometrics Authentication Using Raspberry Pi. International Journal For Trends In Engineering Technology, May 2015.
- [5] Vishal D Gadhav, S. N. Kore. Portable Attendance System Integrated with Learning Management System like Moodle. IEEE International Conference On Recent Trends In Electronics Information Communication Technology, 2016.
- [6] Hari Baabu V, Senthil kumar G, Meru Prabhatb and Suhail Sayeed Bukhari. Smart Classroom Attendance System. International Journal of Control Theory and Applications, 2016.
- [7] Dhaval Chheda, Divyesh Darde, Shraddha Chitalia. Smart Projectors using Remote Controlled Raspberry Pi. International Journal of Computer Applications, November 2013.
- [8] Arief Budiman, Joko Triono. Smart and Simple Classroom Presentation Tools. International Journal of Advanced Research in Computer Science, Nov-Dec 2016
- [9] Ansari Imran, Bhatkar Firoz, Chaudhary Faisal, Khan Shahrukh, Prof. Mubashir Khan. Smart Lecture Delivery Using Raspberry Pi. International Journal of Advanced Research in Computer Engineering Technology (IJARCET), March 2015.
- [10] Harshit Yadav, Kapil Chandra, Ritika Davis, and Sandeep Rasgotra (2012), Institute Of engineering And technology, Hamirpur, Virtual Classroom. Significant Project Synopsis presented to the Department of Computer Engineering and Science Institute Of engineering And technology, Hamirpur, cse.nith.ac.in/Document/btech_mjrprojects/report/6.pdf, accessed on June 11, 2013.
- [11] Mahesh Maurya, Chitvan Oza, Alok Shah, and Sadiya Patel (2011), Virtual Classroom Process, www.nmims.edu/1809177872/virtual%20Classroom%20System%20NCTET2011.pdf, accessed on 10/06/2013
- [12] Lertkulvanich Suriyong, Nithi Buranajant, and Suwant to Sombunsukho (2010), The Development of Web - based Classroom instruction by
- [13]. Development of a Communication Is the transfer Classroom System, Obasa A.I. (2010). University Of nigeriA, Akure, MTech thesis.
- [14] In their 2011 paper, Enhancing Characteristic by Using Virtual Collaborative

- Teachers and Web-based Collaborative Work, Wichian Premchaiswadi, Anucha Tungkasthan, and Nipat Jongsawat Access as of 11/06/2013: ww.ieec.uned.es/Investigacion/Educon2010/SearchTool/EDUCON2010/papers/2010S09A03.pdf
- [15] . Web-Based Fully interactive School System Model for Senior Secondary schools, Ibam Michael Onwuka (2012) Documents from the 2012 Summit of something like the African Library And information Network, held May 18–20 at 1 Northeastern University and the DoubleTree Hilton, Florence Brownstone University Hotel
- [16] . Jang-Ping Sheu and Chih-Yung Chang (2002), Ad Hoc Teaching and eSchoolbag System Design and Implementation for Universal Learning IEEE [18] 0-7695-1706-4/02 Proceedings of such IEEE International Symposium on Generation of Wireless Technologies in Education (WMTE-02). How to Design a Virtual Classroom: 10 Easy Steps to Follow, by Hsu Sam, Oge, Marques, M. Khalid Hamza, and Bassem Alhalabi (2011), www.pps.teithe.gr/491995645/HowToDesignAVirtualClassroom.pdf, accessed 10/06/2013
- [17]. Investigation and development of Content Virtual Online Classroom, Zongkai Yang and Qingtang Liu, 2007. Computers & Pedagogy Volume 48, 171–184
- [18] ScienDirect. Elluminate Live! Sessions, released in 2007. Canada's Elluminate Inc. Moodle, Moodle for Teachers, Trainers, and Administrators, 2005 [21]. the Moodle website, retrieved.