CHAPTER 1

INTRODUCTION

1.1 GENERAL

In today's world, communication and information sharing play a critical role in both personal and professional contexts. However, not everyone can read and write in the same language, which can create barriers to effective communication and access to information. In this project, we aim to address this issue by developing a text-to-speech system that can capture an image containing text, extract the text using optical character recognition (OCR), translate the text to a desired language, and generate speech from the translated text. This project involves the use of a Raspberry Pi 4 along with a Pi Camera module 3 to capture the input image, which is then processed using Google Cloud Vision API for OCR. The extracted text is then translated to the desired language using the Google Translate API. Finally, the translated text is converted to speech using the Google Cloud Text-to-Speech API. This project is designed to be both user-friendly and efficient, enabling users to capture images containing text in any language and receive a spoken translation in their desired language. This project has many potential applications, including education, language learning, and accessibility for individuals with visual impairments. Overall, this project represents a step towards breaking down language barriers and facilitating communication and access to information for individuals worldwide.

1.2 OPTICAL CHARACTER RECOGNITION

Optical character recognition (OCR) is a technology that enables computers to recognize printed or handwritten text in digital images or scanned documents.

OCR is used in a wide range of applications, including document scanning and digitization, automated data entry, and text recognition in images. OCR works by analysing an image of a printed or handwritten text and using algorithms to identify individual characters and words. The process involves several steps, including image pre-processing, character segmentation, feature extraction, and classification. During the image pre-processing stage, the image is filtered and enhanced to improve its clarity and contrast. Next, character segmentation involves identifying and separating individual characters in the image. Feature extraction is the process of identifying the unique characteristics of each character, such as its shape, size, and orientation. Finally, the characters are classified using machine learning algorithms that match the extracted features to known characters in a database. OCR has become an essential tool in the digital age, enabling large volumes of text to be converted quickly and accurately into digital format. It has also opened up new possibilities for text recognition in images, enabling computers to "read" and process information from a wide range of sources, including handwritten notes, receipts, and other documents. Despite the advances in OCR technology, it still faces challenges in recognizing handwritten text accurately and dealing with variations in text layout, font, and style. Nevertheless, OCR remains a critical technology for many applications and continues to improve with advances in machine learning and computer vision.

1.3 GOOGLE CLOUD VISION API

The Google Cloud Vision API is a powerful image recognition tool that can automatically analyse and classify images using advanced machine learning algorithms. It can detect objects, faces, logos, text, and other visual elements within an image, providing insights and actionable data for a wide range of applications. Developed by Google, the Cloud Vision API provides developers with easy-to-use tools for image analysis and recognition. With the Cloud Vision API, developers can quickly and accurately classify images, detect and extract text from images using OCR, and even identify inappropriate content in images. The API can also recognize and identify landmarks, detect and track objects in motion, and generate descriptive captions for images. The Cloud Vision API uses deep learning models to analyse images, allowing it to accurately recognize and classify images with high precision. The API is designed to be scalable and flexible, making it suitable for a wide range of applications, from mobile apps to large-scale enterprise systems. Some common use cases for the Cloud Vision API include image analysis and recognition in e-commerce, image search and retrieval in media and entertainment, and object and face detection in security and surveillance systems. In summary, the Google Cloud Vision API is a powerful image recognition tool that enables developers to easily analyse and classify images using advanced machine learning algorithms. With its flexible and scalable design, the Cloud Vision API can be used for a wide range of applications, providing valuable insights and actionable data for businesses and organizations around the world.

1.4 GOOGLE TRANSLATE

Google Translate is a free online translation service developed by Google. It provides machine translation of text, web pages, and documents between over 100 languages. The service was first introduced in 2006 and has since become one of the most widely used translation services in the world. Google Translate uses machine learning algorithms to provide accurate translations. The system is based on statistical machine translation, which involves analysing large amounts of language data to identify patterns and generate translations. In recent years, Google has also integrated neural machine translation (NMT) into its system, which uses deep learning techniques to improve translation accuracy and fluency. Google Translate offers a range of features, including text translation, website translation, and document translation. The service can also detect the language of the input text automatically, making it easy to use for individuals who may not be familiar with the language they are translating. Google Translate is available as a web application and as a mobile app for both Android and iOS devices. The service is free to use, and there are no limits on the number of translations that can be performed. In summary, Google Translate is a powerful and widely used translation service that enables individuals to communicate and access information across languages. Its machine learning algorithms and neural machine translation technology have made it one of the most accurate translation services available today.

1.5 GOOGLE CLOUD TEXT TO SPEECH API

The Google Cloud Text-to-Speech API is a cloud-based service that enables developers to convert text into natural-sounding speech. The API uses deep learning techniques to generate human-like speech with lifelike intonation, rhythm, and pacing. The API offers a wide range of voices, each with its own unique style and characteristics. These voices are available in multiple languages, allowing developers to generate speech in different languages with high accuracy and quality. The Google Cloud Text-to-Speech API is designed to be highly customizable, offering developers control over various aspects of speech synthesis, including pitch, speed, and volume. The API also supports the use of SSML (Speech Synthesis Markup Language), which allows developers to add pauses, emphasis, and other expressive elements to their speech output. In addition to its high-quality speech output, the Google Cloud Text-to-Speech API offers fast and efficient processing, with low latency and high reliability. The API is scalable, allowing developers to generate speech for large volumes of text efficiently. Overall, the Google Cloud Text-to-Speech API represents a significant advancement in speech synthesis technology, offering developers a powerful and flexible tool for creating natural-sounding speech for a wide range of applications, including voice assistants, audiobooks, and accessibility solutions for individuals with visual impairments.

1.6 NEED FOR THIS PROJECT

- With the increasing globalization and interconnectivity of the world, it is becoming increasingly essential to be able to read, understand, and communicate in different languages.
- However, language barriers can be a significant obstacle for many people, including those who are visually impaired or have reading difficulty.
- This project aims to overcome these barriers by providing a tool that can recognize and translate text from images, scanned documents, or handwritten notes in various languages.
- Additionally, the text can be converted to speech, allowing users to listen to the content and improve their language skills.
- This project has numerous practical applications in various fields such as education, business, and healthcare, where multilingual communication is essential.

1.7 OBJECTIVES

The overall objective of this project is to develop an automated text to speech device for the system is built considering the following design factors:

- **Accuracy:** Using Google Cloud Vision API is to leverage its powerful OCR technology to accurately recognize printed characters in a variety of settings, even when the text is of poor quality, has unusual fonts or is handwritten.

- **Language Translation:** Using Google Translator is to accurately translate text from one language to another, while taking into consideration the nuances of the language, including grammar, syntax, and cultural context.

- **Text-to-Speech Quality:** Using Google Cloud Text-to-Speech is to ensure high-quality speech output that sounds natural and is easy to understand, even in languages with complex phonetic and tonal systems.

- **Integration of different technologies:** Using Raspberry Pi processor to overcome the challenges associated with technology integration, ultimately producing a robust and reliable system that can process OCR, translation, and text-to-speech tasks with minimal user intervention.

- **User Interface:** The Device should be lightweight, easy to use, and provide high-quality text extraction, translations and speech output.

1.8 ORGANISATION OF CHAPTERS

- Chapter 2 deals with review of literature on the topics with summarization
- Chapter 3 describes methodology and defines the extraction of text from image using OCR, Text translation using Translator, and Text-to-Speech Conversion.
- Chapter 4 deals with the simulation result & discussion of our result.

CHAPTER 2

LITERATURE SURVEY

2.1 GENERAL

In this chapter, a brief discussion about the techniques for text extraction using OCR, translator process, and TTS conversion is made and implementing into hardware.

2.2 REVIEW OF LITERATURE SURVEY

Dr. S. A. Ubale el al (2022) explained the Internet is a bone to mankind. The main field revolutionised by the internet is communication. A Text-to-speech synthesiser is used to convert text into speech (voice) by analysing and processing the text using Natural Language Processing and then using Digital Signal Processing technology to convert this processed text into a synthesised speech representation of the text. Through this paper, we aim to study the different methodologies for Speech-To-Text and Text-To-Speech conversion that will be used in a voice-based email system. Developed a useful Text-to- Speech synthesiser in the form of a simple application that converts inputted text into synthesised speech and reads it out to the user, which can then be saved as an mp3. file. The development of a text-to-speech synthesiser will be of great help to people with visual blindness and make reading through large volumes of text easier.

Prof. Mrunalinee Patole et al (2021) discussed Text to Speech (TTS) is a form of speech synthesis where in the text is converted right into a spoken human-like voice output. The state-of-the-art strategies for TTS employ a neural network based totally method. These paintings pursuits to take a look at a number of the problems and barriers gift inside the contemporary works, especially Tacotron-2, and attempts to in addition enhance its performance by means of editing its structure. till now many papers were published on these topics that display various exceptional TTS structures by means of developing new TTS products. The aim is to have a look at different textual content-to-Speech structures. in comparison to different text-to-Speech systems, Tacotron2 has multiple blessings. In opportunity algorithms like CNN, speedy-CNN the algorithmic program may not investigate the photo fully however in YOLO the algorithmic application checks out the picture absolutely by predicting the bounding boxes through using convolutional network and possibilities for those packing containers and detects the image faster in comparison to alternative algorithms.

Paras Doshi et al (2018) explained visually impaired people confront a number of visual challenges every day – from reading the label on a frozen dinner to figuring out if they're at the right bus stop. Probable solutions include Braille wherein tactile information is converted into meaningful patterns. Other visual aids include liquid level indicators, coin sorters and large button telephones for daily living; electronic magnifiers, audio books, text to voice technology as a technological aid. Aim through this paper is to propose a system that facilitates reading for a blind person. With the help of our system, we extract text from images using google cloud vision API. Approach is capable of recognizing text in various challenging conditions where traditional OCR systems fail; in the presence of blur, low resolution, low contrast, high image noise, and distortions. The output text is converted into audio output in the form of synthetic speech. Thus, proposed system will be very helpful to visually impaired person.

Vaishnavi R. Ambaskar et al (2022) reviewed speech is one amongst the oldest and most natural means that of data exchange between human. Over the years, tries are created to develop vocally interactive computers to understand voice/speech synthesis. Clearly such AN interface would yield nice advantages. During this case a pc will synthesize text and provides out a speech. Text-To-Speech Synthesis may be a Technology that gives a way of changing written communication from a descriptive type to a speech communication that's simply comprehensible by the top user (Basically in English Language). It runs on JAVA platform, and also the methodology used was Object orientating Analysis and Development Methodology; whereas knowledgeable System was incorporated for the interior operations of the program. This style is going to be double-geared towards providing a unidirectional communication interface whereby the pc communicates with the user by reading out matter document for the aim of fast assimilation and reading development. These days, communication is that the key part to progress. Passing on information, to the right person, and inside the correct manner is implausibly very important, not merely on a corporation level, but to boot on a personal level Thus, on serve the aim of effective communication between two parties whereas not hindrances, many applications have come to image, that acts as a negotiator and facilitate in effectively carrying messages fashionable of text, or speech signals over miles of networks. Most of these applications understand the use of functions like pronunciation and acoustic-based speech recognition, conversion from speech signals to text, and from text to artificial speech signals, language translation amongst varied others.

Shruti Mankar et al (2023) discussed all of us are aware of how important knowledge is and it is also true that mostly the data or knowledge is in the form of books or online articles, or various pdfs i.e., in text format. But not all of us are privileged to read. Some are illiterate, some are blind, and some have reading difficulties. Hence, a form of adaptive technology or procedure that reads digital text aloud, which is called text-to-speech (TTS) was developed. It is occasionally referred to as "read-aloud" technology. Words on a computer or other digital device can be converted into audio using TTS. TTS is particularly beneficial for all those people who have reading difficulties or are illiterate. Also, a significant amount of research has been done and is currently being done on text-to-speech technology. Various technologies, methodologies, and algorithms are used in the various proposed approaches and solutions for TTS. This research presents a systematic review of all those methods which have been proposed and implemented by different active researchers in this field.

Nisha P et al (2021) proposed there is an image everywhere around us and we see the image and read the text in our day-to-day life. Like bus names, bus numbers, hotel names, newspapers, etc. But the question is how Visually Impaired or blind people can recognize this text. Surely, they need some assistance to read the text. In this research, the images are converted into text and the text is converted into audio output. It is mainly used for low visual persons or blind peoples to recognize the text. The field of research in Character recognition, Speech recognition and computer vision. In this research, as the recognition process is done using OCR, Raspberry Pi, MAT lab and OpenCV library. It recognizes characters using API, the e-Speak algorithm, PYTHON, and JAVA programming. This paper explains the purpose, implementation, and test results of the device. This project consists of capturing the image, text localization, text to audio conversion.

Sharvari S et al (2020) explained the present situation, communication plays a vital role in the world. Transferring on information, to the correct person, and in the proper manner is very important on a personal and professional level. The world is moving towards digitization, so are the means of communication are Phone calls, emails, text messages etc. have become a major part of message conveyance in this digital world. In order to serve the purpose of effective communication between two parties without any delay, many applications have come to existence, which acts as a mediator and help in effectively carrying messages in from text to the speech signals over miles of networks. The main purpose of this project is to overcome the problems facing by the blind people and illiterates. Because the blind people and illiterates can be easily manipulated, this leads to misuse. To overcome this problem, we are proposing a device which helps in conversion of hard copy of text which is inserted into the device will be converted to speech. Most of these applications find the use of functions such as articulators, conversion from text to synthetic speech signals, language translation amongst various others. In this project, we'll be executing different techniques and algorithms that are applied to achieve the concept of Text to Speech (TTS).

Ashanti Widyana et al (2022) discussed about the technology has improved temporarily from time to time. There are a lot of technologies that had been developed to enhance language learning, and one of them is text-to-speech technology. Text-to-speech technology is a form of system that can convert phoneme to audio. It has provided an impact in language learning since it was developed. This article presents how the application of text-to-speech technology is used in language learning, including the negative and positive side of text-to-speech technology in language learning. It reports on the results of a systematic review of articles that specifically examine the use of text-to-speech technology in language learning. The articles were then reviewed and selected using the Preferred Reporting Items for Systematic Reviews and Meta-Analysis (PRISMA) approach. The analysis results of 20 selected articles revealed that the use of text-to-speech assisted the process of knowledge transfer. Text-to-speech technology has also played a practical role in language learning, especially in improving students' language skills. The review also revealed that text-to-speech technology lacks in intonation, eye-contact, and real-time class interaction. But overall, despite that it has a slight negative impact, text-to-speech technology can be a breakthrough to support language learning.

Chooi Shir Ley et al (2021) proposed the handwriting character recognition is one of the interesting research realms in artificial intelligence. Handwriting recognition is assumed to obtain and identify characters in handwritten documents, images, touch-screen devices, and other sources and convert them into digital form. A handwriting recognition system is a tool for identifying the alphabet and number from an image. It converts the image into a machine-encoded form by applying the technique of machine learning. Machine learning is the ability to learn one of the distinctive attributes of intelligent behaviour. Machine learning is a data analytics technique that teaches computers to do what comes naturally to humans. Machine learning algorithms use computational methods to “learn” information spontaneously from data without relying on a predetermined equation. The algorithms adaptively improve their achievement as the number of samples available for learning increases.

Karun Somasunder M et al (2021) implemented an assistive device that's capable of capturing a picture from a camera and extracting the text from the captured image and further to convert the text to speech as voice-based output to assist the people. The captured image is analysed using Google Cloud Vision API Optical Character recognition (OCR). So as to extract text, we use image pre-processing methods to obviate any noise or blur within the captured image so that the accuracy is often increased. Further, it includes software-based text to speech to convert the text to speech as voice output. The Google Cloud Speech API integrates with Google Cloud Storage for data storage.

2.3 SUMMARY OF REVIEW

From the brief survey presented above, it is observed that various changes have been made in an OCR, Translator and TTS. However, there are some limitations to this project. The accuracy of OCR can be affected by factors such as lighting, image quality, and font type, which can lead to errors in text recognition. The OCR engine used in this project is Google Cloud Vision, which is a powerful and accurate OCR engine that can recognize text from a wide variety of sources. The text-to-speech engine used in this project is Google Cloud Text-to-Speech, which can produce high-quality audio output with natural-sounding voices. Overall, the project is a useful demonstration of the capabilities of OCR, machine translation, and text-to-speech technologies. It has practical applications in fields such as accessibility, language learning, and translation services. The Raspberry Pi 4 is an excellent platform for this project, as it is affordable and has sufficient processing power to run the necessary software.

CHAPTER 3

METHODOLOGY

3.1 GENERAL

The current system in use is that words of any language are often typed manually and translated to any language as needed. With the use of existing systems, it is impossible to convert the image containing text in a different language to speech in a common standard like English. To overcome the problems that we have identified, we are using Google Cloud Vision API for text recognition and in real time. It was released in the year 2015. It enables developers to analyse the content of images. It uses powerful machine learning tools to extract the required data from images. It can perform different functions like label detection, face detection, Logo detection, Optical Character Recognition (OCR). Finally, the UTF-8 Unicode text is fed to a Text to Speech engine such as Google Cloud TTS which converts it into speech. The converted speech is then sent back to the device from the server where the user can access it. We can use Google translate to translate from one language to another, thereby supporting Multi Language translation.

3.2 PRINCIPLE OF THE SYSTEM BUILT:

Capture Image: The first step is the one in which the document is placed in front of the camera and the camera captures an image of the placed document. The quality of the image captured will be high so as to have fast and clear recognition due to the high-resolution camera.

Pre-Process Image: The captured image will be pre-processed using OpenCV. This will involve converting the image to grayscale, applying thresholding to enhance the text, and removing any noise from the image.

Extract Text: The pre-processed image will be passed to the Google Cloud Vision to extract the text.

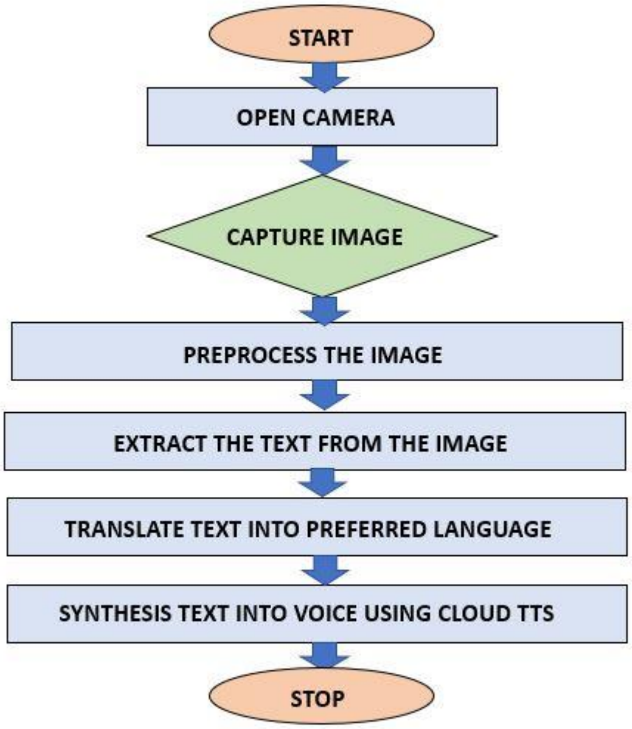

START
OPEN CAMERA
CAPTURE IMAGE
PREPROCESS THE IMAGE
EXTRACT THE TEXT FROM THE IMAGE
TRANSLATE TEXT INTO PREFERRED LANGUAGE
SYNTHESIS TEXT INTO VOICE USING CLOUD TTS
STOP

Figure 3.1: Flow chart

Post-Processing: Post-processing techniques can help to correct or improve the OCR output by analyzing the recognized text and making corrections based on rules or machine learning models. Some common post-processing techniques used in Cloud Vision OCR include: spell checking, language identification, text normalization, post-correction.

Translate Text: The extracted text will be passed to the Google Translate API to translate it into the desired language.

Convert Text to Speech: The translated text will be passed to Google Cloud Text-to-Speech API to convert it into speech.

Output: The extracted text and translated text will be displayed on the screen, and the converted text into speech output(audio) is listened either by connecting headsets via 3.5mm audio jack or by connecting speakers via Bluetooth.

3.3 GOOGLE CLOUD VISION API

The Google Cloud Vision API is a machine learning service that allows you to integrate image analysis into your applications. It can recognize and extract information from images, such as text, logos, and objects, and convert it into machine-readable data. One of the features of the Google Cloud Vision API is the ability to perform Optical Character Recognition (OCR) on images, which means it can extract text from images and convert it into machine-readable text. This can be useful for a variety of applications, including text-to-speech conversion. Text-to-speech conversion is the process of converting written text into spoken words. By using the Google Cloud Vision API to extract text from images, you can then use a text-to-speech engine to convert that text into speech. This can be useful in scenarios where the text is embedded in an image, such as a sign or a poster, and needs to be read out loud for people who cannot see the image.

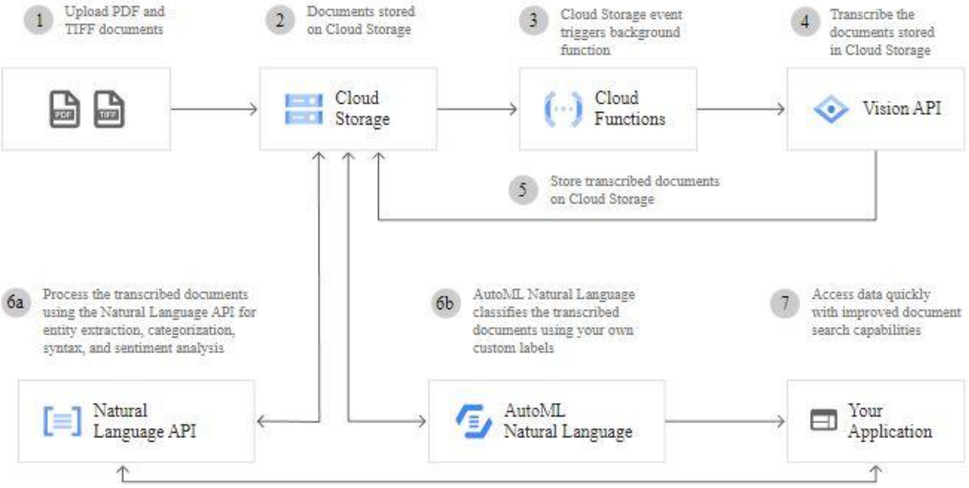

1
Upload PDF and TIFF documents
PDF
TIFF
2
Documents stored on Cloud Storage
Cloud Storage
3
Cloud Storage event triggers background function
Cloud Functions
4
Transcribe the documents stored in Cloud Storage
Vision API
5
Store transcribed documents on Cloud Storage
6a
Process the transcribed documents using the Natural Language API for entity extraction, categorization, syntax, and sentiment analysis
Natural Language API
6b
AutoML Natural Language classifies the transcribed documents using your own custom labels
AutoML Natural Language
7
Access data quickly with improved document search capabilities
Your Application

Figure 3.2: Flow diagram of Vision API.

Overall, the Google Cloud Vision API can be a powerful tool for text-to-speech conversion, as it allows you to extract text from images with high accuracy and reliability, making it easier to provide accessibility features for those with visual impairments or other disabilities.

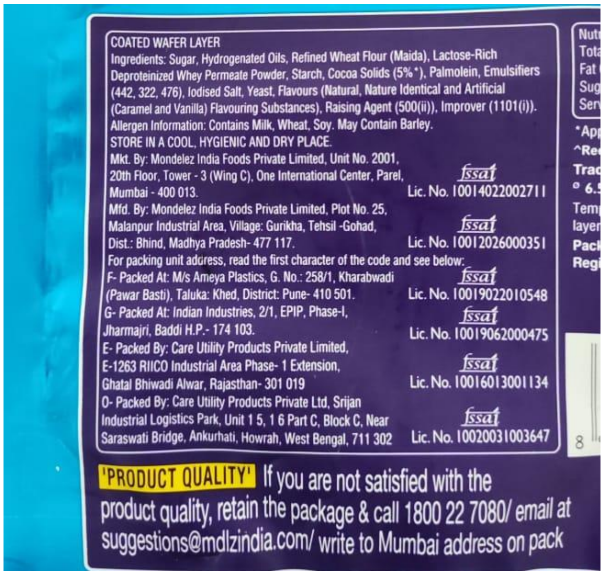
3.3.1 PRE-PROCESSING

Google Cloud Vision API provides a variety of pre-processing options to optimize an image for analysis. Here are some of the key pre-processing features of the Cloud Vision API.

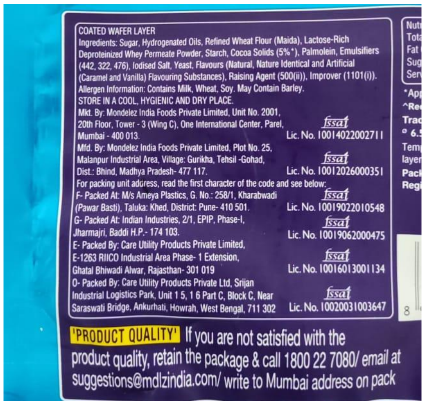
Image resizing: It is the process of changing the size of an image to a new resolution or aspect ratio. Resizing an image can be useful for a variety of reasons, such as reducing the file size, preparing an image for display on a website or mobile device, or optimizing an image for analysis by an AI or machine learning model, such as the Google Cloud Vision API.

- **Scaling:** An image involves changing its size by a certain percentage. For example, you can scale an image up by 50% to make it 1.5 times its original size, or scale it down by 25% to make it 0.75 times its original size.
- **Aspect ratio preservation:** When resizing an image, it's important to preserve the aspect ratio to avoid distorting the image. The Cloud Vision API can automatically adjust the size of an image while maintaining its aspect ratio.
- **Padding:** It is the process while resizing an image to a different aspect ratio, the Cloud Vision API can add padding to fill the empty space around the image. This can help maintain the image's original proportions and prevent distortion.

- **Thumbnail generation:** The Cloud Vision API can generate a thumbnail version of an image that is smaller in size and resolution, making it ideal for use in mobile apps, websites, and other digital applications.


COATED WAFER LAYER
Ingredients: Sugar, Hydrogenated Oils, Refined Wheat Flour (Maida), Lactose-Rich Deproteinized Whey Permeate Powder, Starch, Cocoa Solids (5%*), Palmolein, Emulsifiers (442, 322, 476), Iodised Salt, Yeast, Flavours (Natural, Nature Identical and Artificial (Caramel and Vanilla) Flavouring Substances), Raising Agent (500(ii)), Improver (1101(i)).
Allergen Information: Contains Milk, Wheat, Soy. May Contain Barley.
STORE IN A COOL, HYGIENIC AND DRY PLACE.
Mkt. By: Mondelez India Foods Private Limited, Unit No. 2001, 20th Floor, Tower - 3 (Wing C), One International Center, Parel, Mumbai - 400 013. Lic. No. 10014022002711
Mfd. By: Mondelez India Foods Private Limited, Plot No. 25, Malanpur Industrial Area, Village: Gurikha, Tehsil -Gohad, Dist.: Bhind, Madhya Pradesh- 477 117. Lic. No. 10012026000351
For packing unit address, read the first character of the code and see below:
F- Packed At: M/s Ameya Plastics, G. No.: 258/1, Kharabwadi (Pawar Basti), Taluka: Khed, District: Pune- 410 501. Lic. No. 10019022010548
G- Packed At: Indian Industries, 2/1, EPIP, Phase-I, Jharmajri, Baddi H.P.- 174 103. Lic. No. 10019062000475
E- Packed By: Care Utility Products Private Limited, E-1263 RIICO Industrial Area Phase- 1 Extension, Ghatal Bhiwadi Alwar, Rajasthan- 301 019 Lic. No. 10016013001134
O- Packed By: Care Utility Products Private Ltd, Srijan Industrial Logistics Park, Unit 1 5, 1 6 Part C, Block C, Near Saraswati Bridge, Ankurhati, Howrah, West Bengal, 711 302 Lic. No. 10020031003647
fssai
'PRODUCT QUALITY' If you are not satisfied with the product quality, retain the package & call 1800 22 7080/ email at suggestions@mdlzindia.com/ write to Mumbai address on pack
Nutr
Tota
Fat
Sug
Serv
*Ap
^Re
Trac
º 6.5
Temp
layer
Pack
Regi
8

Before


COATED WAFER LAYER
Ingredients: Sugar, Hydrogenated Oils, Refined Wheat Flour (Maida), Lactose-Rich Deproteinized Whey Permeate Powder, Starch, Cocoa Solids (5%*), Palmolein, Emulsifiers (442, 322, 476), Iodised Salt, Yeast, Flavours (Natural, Nature Identical and Artificial (Caramel and Vanilla) Flavouring Substances), Raising Agent (500(ii)), Improver (1101(i)).
Allergen Information: Contains Milk, Wheat, Soy. May Contain Barley.
STORE IN A COOL, HYGIENIC AND DRY PLACE.
Mkt. By: Mondelez India Foods Private Limited, Unit No. 2001, 20th Floor, Tower - 3 (Wing C), One International Center, Parel, Mumbai - 400 013. Lic. No. 10014022002711
Mfd. By: Mondelez India Foods Private Limited, Plot No. 25, Malanpur Industrial Area, Village: Gurikha, Tehsil -Gohad, Dist.: Bhind, Madhya Pradesh- 477 117. Lic. No. 10012026000351
For packing unit address, read the first character of the code and see below:
F- Packed At: M/s Ameya Plastics, G. No.: 258/1, Kharabwadi (Pawar Basti), Taluka: Khed, District: Pune- 410 501. Lic. No. 10019022010548
G- Packed At: Indian Industries, 2/1, EPIP, Phase-I, Jharmajri, Baddi H.P.- 174 103. Lic. No. 10019062000475
E- Packed By: Care Utility Products Private Limited, E-1263 RIICO Industrial Area Phase- 1 Extension, Ghatal Bhiwadi Alwar, Rajasthan- 301 019 Lic. No. 10016013001134
O- Packed By: Care Utility Products Private Ltd, Srijan Industrial Logistics Park, Unit 1 5, 1 6 Part C, Block C, Near Saraswati Bridge, Ankurhati, Howrah, West Bengal, 711 302 Lic. No. 10020031003647
fssai
'PRODUCT QUALITY' If you are not satisfied with the product quality, retain the package & call 1800 22 7080/ email at suggestions@mdlzindia.com/ write to Mumbai address on pack

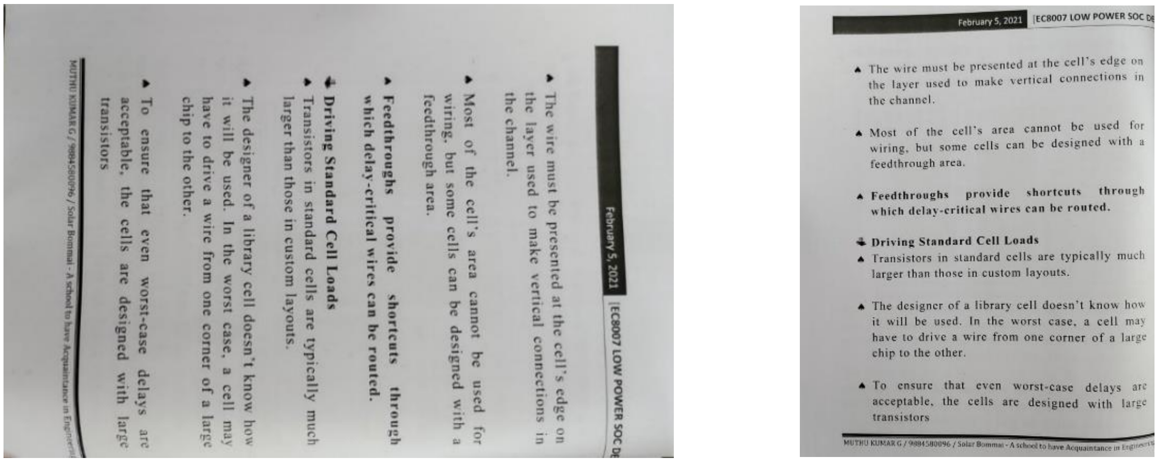
After

Figure 3.3: Image resizing.

Image rotation: It is a common pre-processing step in image analysis that involves changing the orientation of an image. This can be useful for aligning an image correctly or for analysing images that are taken in different orientations. The Google Cloud Vision API provides several options for rotating an image.

- **Automatic orientation detection:** The Cloud Vision API can automatically detect the correct orientation of an image and adjust it accordingly. This is particularly useful when dealing with images that have been taken at different angles or orientations.

- **Explicit rotation:** You can specify a rotation angle in degrees to rotate an image explicitly. This can be useful when you need to rotate an image to a specific angle, such as 90 degrees, 180 degrees, or 270 degrees.
- **Orientation correction:** The Cloud Vision API can correct the orientation of an image that has been incorrectly rotated or mirrored. This can be particularly useful when working with scanned documents or images that have been processed by optical character recognition (OCR) systems.
- **Custom rotation:** The Cloud Vision API can also rotate an image by a custom angle that you specify. This can be useful when working with images that have been taken at an angle that is not a multiple of 90 degrees.



February 5, 2021 [EC8007 LOW POWER SOC DE

▲ The wire must be presented at the cell's edge on the layer used to make vertical connections in the channel.

▲ Most of the cell's area cannot be used for wiring, but some cells can be designed with a feedthrough area.

▲ **Feedthroughs provide shortcuts through which delay-critical wires can be routed.**

Driving Standard Cell Loads

▲ Transistors in standard cells are typically much larger than those in custom layouts.

▲ The designer of a library cell doesn't know how it will be used. In the worst case, a cell may have to drive a wire from one corner of a large chip to the other.

▲ To ensure that even worst-case delays are acceptable, the cells are designed with large transistors

MUTHU KUMAR G / 9884580096 / Solar Bommai - A school to have Acquaintance in Engineeri

February 5, 2021 [EC8007 LOW POWER SOC DE

▲ The wire must be presented at the cell's edge on the layer used to make vertical connections in the channel.

▲ Most of the cell's area cannot be used for wiring, but some cells can be designed with a feedthrough area.

▲ **Feedthroughs provide shortcuts through which delay-critical wires can be routed.**

Driving Standard Cell Loads

▲ Transistors in standard cells are typically much larger than those in custom layouts.

▲ The designer of a library cell doesn't know how it will be used. In the worst case, a cell may have to drive a wire from one corner of a large chip to the other.

▲ To ensure that even worst-case delays are acceptable, the cells are designed with large transistors

MUTHU KUMAR G / 9884580096 / Solar Bommai - A school to have Acquaintance in Engineeri

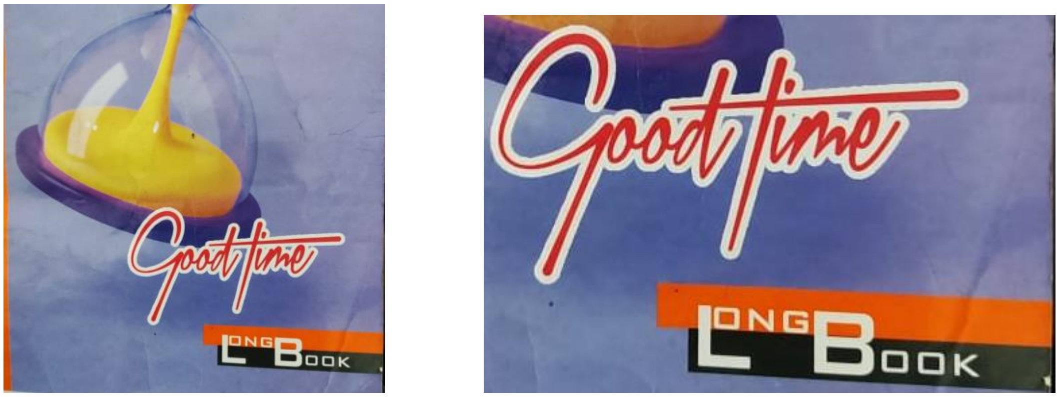
Before

After

Figure 3.4: Image rotation.

Image cropping: It is the process of selecting and extracting a portion of an image to create a new image that contains only the selected portion. This can be useful for focusing on a specific area of an image or for removing unwanted parts of an image. The Google Cloud Vision API provides several options for cropping an image.

- **Automatic object detection:** The Cloud Vision API can automatically detect objects within an image and crop the image to focus on the detected objects. This can be particularly useful when working with images that contain multiple objects, such as a group photo or a landscape with multiple elements.
- **Fixed crop region:** You can specify a fixed crop region in the form of a rectangle with a specific width and height. The Cloud Vision API will extract the region specified by the rectangle and create a new image with only the selected portion.
- **Smart crop:** The Cloud Vision API can perform a smart crop that automatically selects the most important part of an image and crops it accordingly. This can be useful when working with images that have a lot of background or irrelevant content.
- **Aspect ratio preservation:** When cropping an image, important to preserve the aspect ratio to avoid distorting the image. Cloud Vision API can automatically adjust size of the crop region while maintaining its aspect ratio.

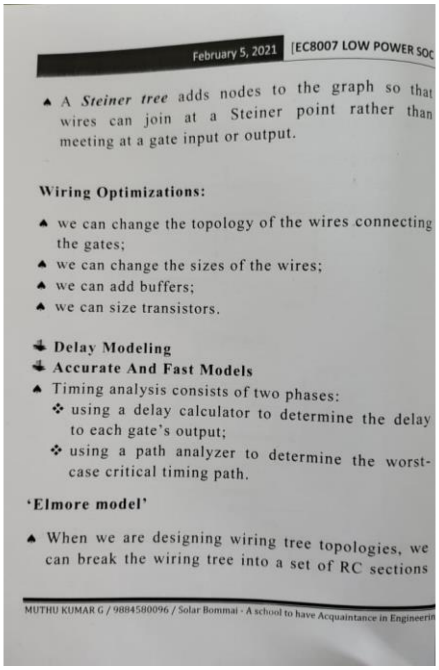


Before After

Figure 3.5: Image cropping.

Image normalization: It is a pre-processing step that aims to adjust the pixel values of an image so that they fall within a certain range or distribution. The goal of image normalization is to improve the consistency and quality of an image by removing any variations in brightness, contrast, or color that may affect the accuracy of image analysis. The Google Cloud Vision API provides several options for image normalization.

- **Automatic image adjustment:** The Cloud Vision API can automatically adjust the brightness, contrast, and color balance of an image to improve its overall quality. This can be particularly useful when working with images that have uneven lighting conditions or color variations.
- **Histogram equalization:** Histogram equalization is a technique for adjusting the pixel values of an image so that they are distributed more evenly across the range of possible values. This can be useful for improving the contrast of an image and enhancing its details.
- **Color space conversion:** The Cloud Vision API can convert an image from one color space to another, such as from RGB to grayscale or from sRGB to Adobe RGB. This can be useful for adjusting the color balance of an image or for simplifying the image for analysis.
- **Scaling and normalization:** The Cloud Vision API can scale an image so that its pixel values fall within a specific range, such as [0, 1] or [-1, 1]. This can be useful for normalizing an image for analysis by machine learning models.



February 5, 2021 [EC8007 LOW POWER SOC

- A *Steiner tree* adds nodes to the graph so that wires can join at a Steiner point rather than meeting at a gate input or output.

Wiring Optimizations:

- we can change the topology of the wires connecting the gates;
- we can change the sizes of the wires;
- we can add buffers;
- we can size transistors.

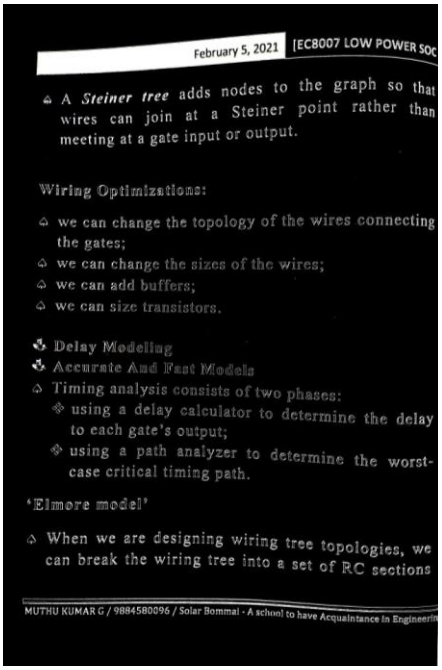
- **Delay Modeling**
- **Accurate And Fast Models**
- Timing analysis consists of two phases:
  - using a delay calculator to determine the delay to each gate's output;
  - using a path analyzer to determine the worst-case critical timing path.

'Elmore model'

- When we are designing wiring tree topologies, we can break the wiring tree into a set of RC sections

MUTHU KUMAR G / 9884580096 / Solar Bommai - A school to have Acquaintance in Engineerin

Before



February 5, 2021 [EC8007 LOW POWER SOC

- A *Steiner tree* adds nodes to the graph so that wires can join at a Steiner point rather than meeting at a gate input or output.

Wiring Optimizations:

- we can change the topology of the wires connecting the gates;
- we can change the sizes of the wires;
- we can add buffers;
- we can size transistors.

- **Delay Modeling**
- **Accurate And Fast Models**
- Timing analysis consists of two phases:
  - using a delay calculator to determine the delay to each gate's output;
  - using a path analyzer to determine the worst-case critical timing path.

'Elmore model'

- When we are designing wiring tree topologies, we can break the wiring tree into a set of RC sections

MUTHU KUMAR G / 9884580096 / Solar Bommai - A school to have Acquaintance in Engineerin

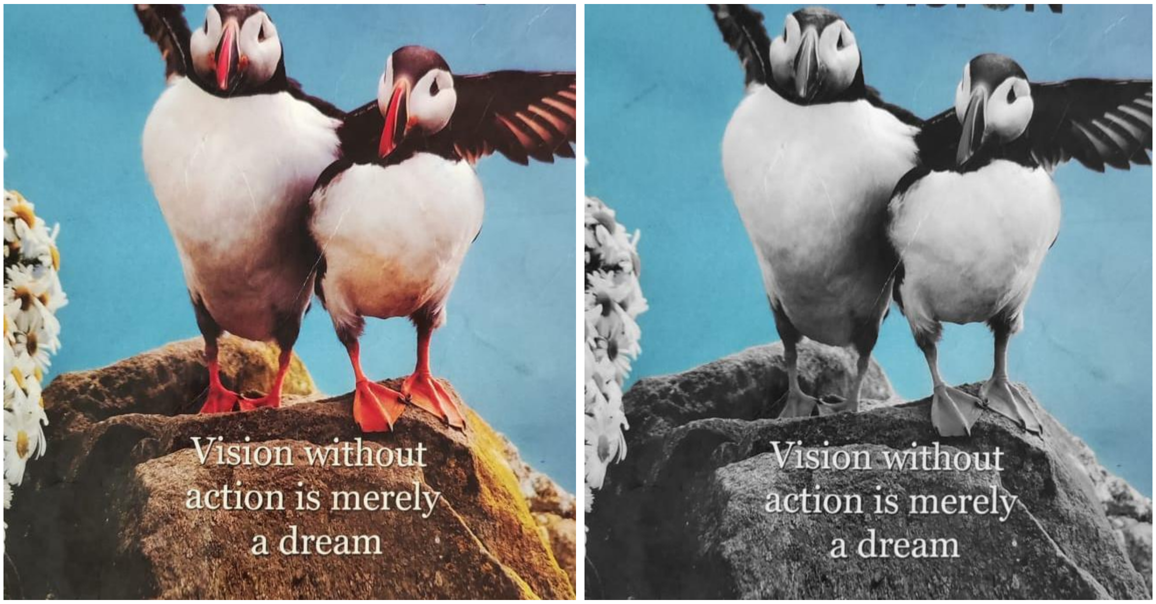
After

Figure 3.6: Image normalization.

Color filtering: It is a pre-processing technique that involves selecting a subset of colors from an image based on their hue, saturation, and brightness values. Color filtering can be used to isolate specific objects or regions within an image based on their color, and can be useful for a variety of applications, such as object recognition, image segmentation, and color-based tracking. The Google Cloud Vision API provides several options for color filtering.

- **Dominant colors detection:** The Cloud Vision API can detect the dominant colors within an image, which can be useful for identifying the most prominent colors within an image.
- **Color matching:** The Cloud Vision API can match specific colors within an image, which can be useful for identifying objects that are a certain color or for detecting specific patterns or shapes within an image.

- **Color extraction:** The Cloud Vision API can extract specific colors from an image and create a new image that contains only the selected colors. This can be useful for isolating specific regions or objects within an image based on their color.
- **Color quantization:** The Cloud Vision API can reduce the number of colors in an image to a specific palette, which can be useful for reducing the size of an image or for simplifying the image for analysis.


Vision without action is merely a dream
Vision without action is merely a dream

Before

After

Figure 3.7: Color filtering.

Image compression: It is a technique used to reduce the size of an image by removing redundant or unnecessary information while preserving the important features of the image. Image compression can be useful for reducing the amount of storage space required for images, improving their transfer over networks with limited bandwidth, and speeding up the processing of images. The Google Cloud Vision API provides several options for image compression.

- **Lossy compression:** The Cloud Vision API can compress an image using a lossy compression algorithm, which removes some of the image data in order to reduce its size. This can result in some loss of image quality, but can be useful for reducing the size of large images or for images that do not need to be stored at high quality.
- **Lossless compression:** The Cloud Vision API can also compress an image using a lossless compression algorithm, which preserves all of the image data but reduces its size through more efficient encoding. This can be useful for preserving the highest possible image quality while still reducing the size of the image.
- **Image format conversion:** The Cloud Vision API can convert an image from one format to another, such as from JPEG to PNG or from BMP to GIF. Different image formats have different compression algorithms and can result in different levels of image quality and file size.

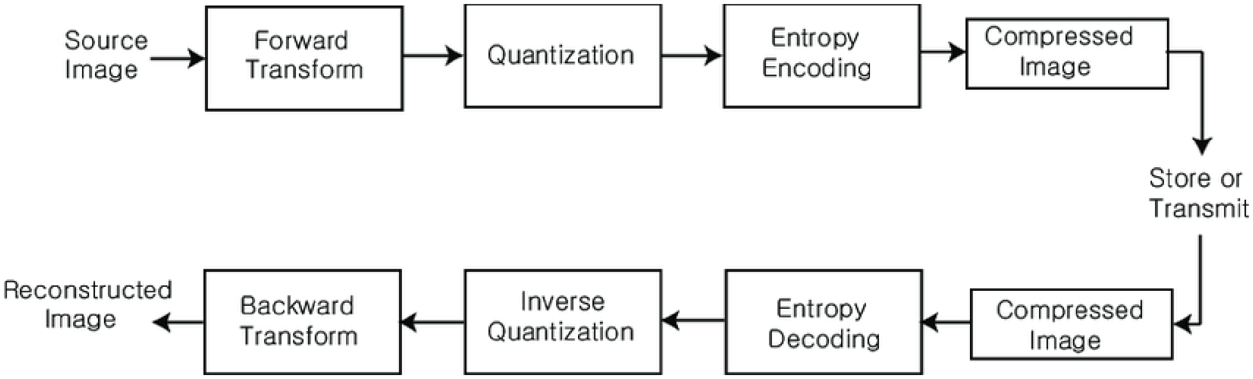


Figure 3.8: Image compression block diagram.

3.3.2 TEXT DETECTION

Text detection is the process of identifying and localizing text regions within an image or a video frame. Text detection is a key feature of the Google Cloud Vision API, which is a cloud-based machine learning platform that provides a range of computer vision capabilities. The text detection feature of the Cloud Vision API enables developers to easily identify and extract text from images, scanned documents, and videos.

Text Recognition: The Cloud Vision API can recognize and extract text from images or video frames, including handwritten or printed text. It can also detect the language of the text and provide the recognized text as a string.

Optical Character Recognition (OCR): The Cloud Vision API can perform OCR on scanned documents or images, which involves recognizing the characters in the image and converting them into machine-readable text. This can be useful for digitizing old documents or for extracting text from printed materials.

OCR is process of classifying optical patterns contained in a digital image. The character recognition is achieved through segmentation, feature extraction and classification. OCR is the recognition of printed or written text characters by a computer. This involves photoscanning of the text character-by-character, analysis of the scanned-in image, and then translation of the character image into character codes, such as ASCII, commonly used in data processing.

In OCR processing, the scanned-in image or bitmap is analyzed for light and dark areas in order to identify each alphabetic letter or numeric digit. When a character is recognized, it is converted into an ASCII code. Special circuit boards and computer chips designed expressly for OCR are used to speed up the recognition process.

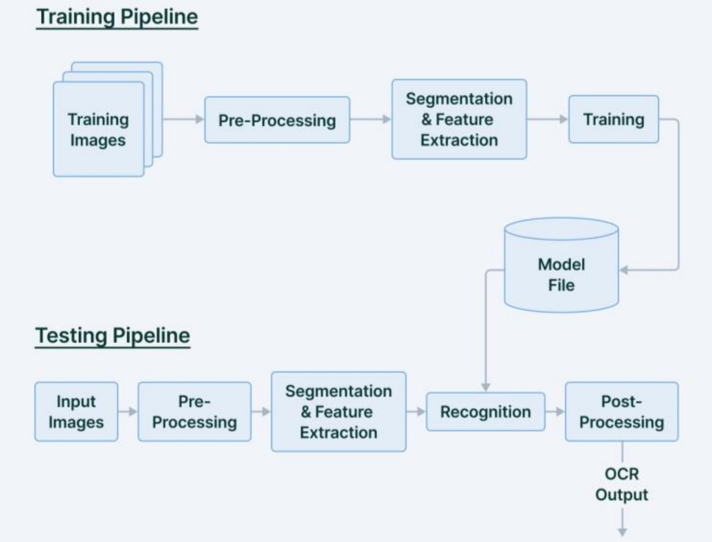
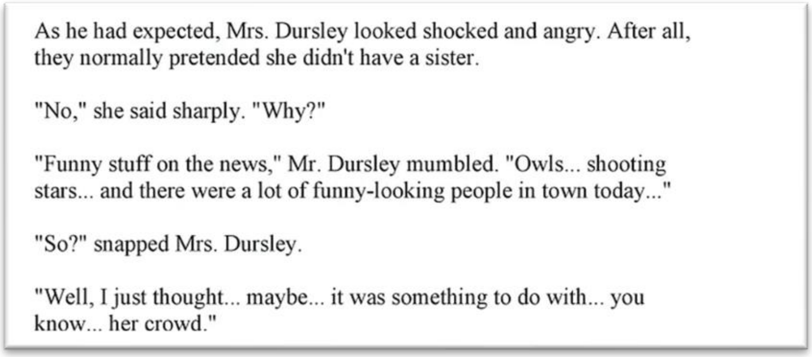


Figure 3.9: General OCR model.

Steps involved in OCR:

1. Optical Scanning ✂ from Image:

Select any document or letter of having text information



As he had expected, Mrs. Dursley looked shocked and angry. After all, they normally pretended she didn't have a sister.

"No," she said sharply. "Why?"

"Funny stuff on the news," Mr. Dursley mumbled. "Owls... shooting stars... and there were a lot of funny-looking people in town today..."

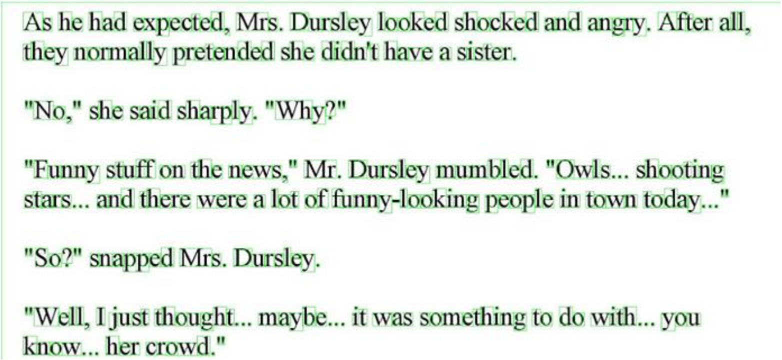
"So?" snapped Mrs. Dursley.

"Well, I just thought... maybe... it was something to do with... you know... her crowd."

Figure 3.10: Image having text information.

Extract character boundaries: Contours can be explained simply as a curve joining all the continuous points (along the boundary). The contours are a useful tool for shape analysis and object detection and recognition. Here Contours explained in differentiating each individual character in an image with using contour dilation technique. Create a boundary to each character in an image with using OpenCV Contours method. Character recognition with the use of OpenCV contours method.

OpenCV code implementation in differentiating the words with the use of contours



As he had expected, Mrs. Dursley looked shocked and angry. After all, they normally pretended she didn't have a sister.

"No," she said sharply. "Why?"

"Funny stuff on the news," Mr. Dursley mumbled. "Owls... shooting stars... and there were a lot of funny-looking people in town today..."

"So?" snapped Mrs. Dursley.

"Well, I just thought... maybe... it was something to do with... you know... her crowd."

Figure 3.11: Extracting character boundaries.

Naming Convention followed (Labelling): The extracted text characters should be labelled with the original character name associated with it. Naming convention followed here is, last letter of file name should be the name associated with the character for pre-processing the images data.

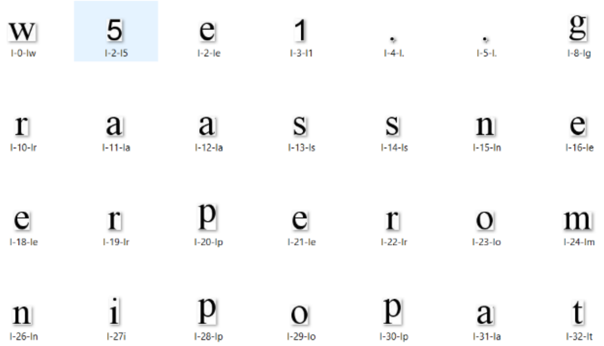


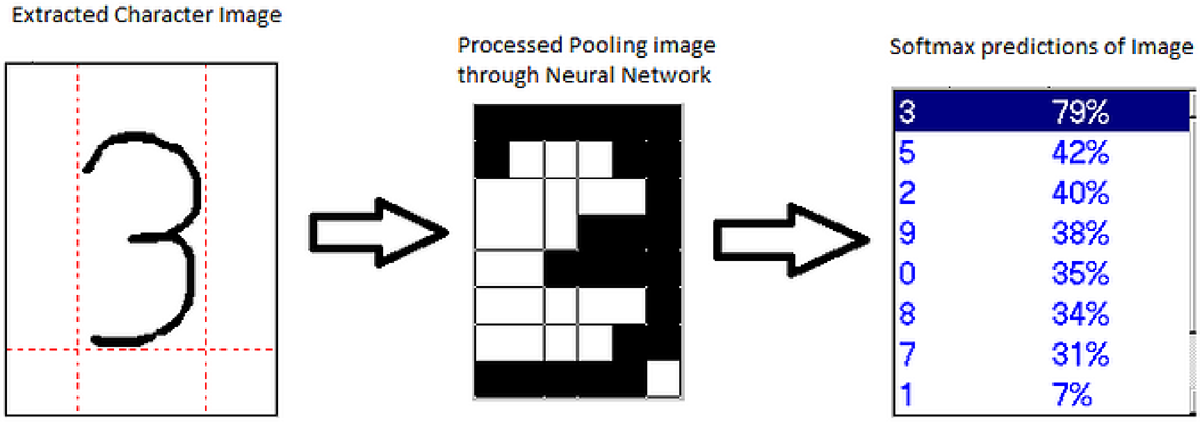
Figure 3.12: Naming Convention followed (Labelling).

Pre-processing: The raw data depending on the data acquisition type is subjected to a number of preliminary processing steps to make it usable in the descriptive stages of character analysis. The image resulting from scanning process may contain certain amount of noise. Smoothing implies both filling and thinning. Filling eliminates small breaks, gaps and holes in digitized characters while thinning reduces width of line.

(a) noise reduction (b) normalization of the data and (c) compression in the amount of information to be retained.

2. Build a ConvNet Model ✂(Character Recognition Model):

Convolution Network of 8 layers with 2*4 layers residual feedbacks used in remembering the Patterns ✂ of the Individual Character Images.


Extracted Character Image
Processed Pooling image through Neural Network
Softmax predictions of Image

3	79%
5	42%
2	40%
9	38%
0	35%
8	34%
7	31%
1	7%

Figure 3.13: ConvNet Model.

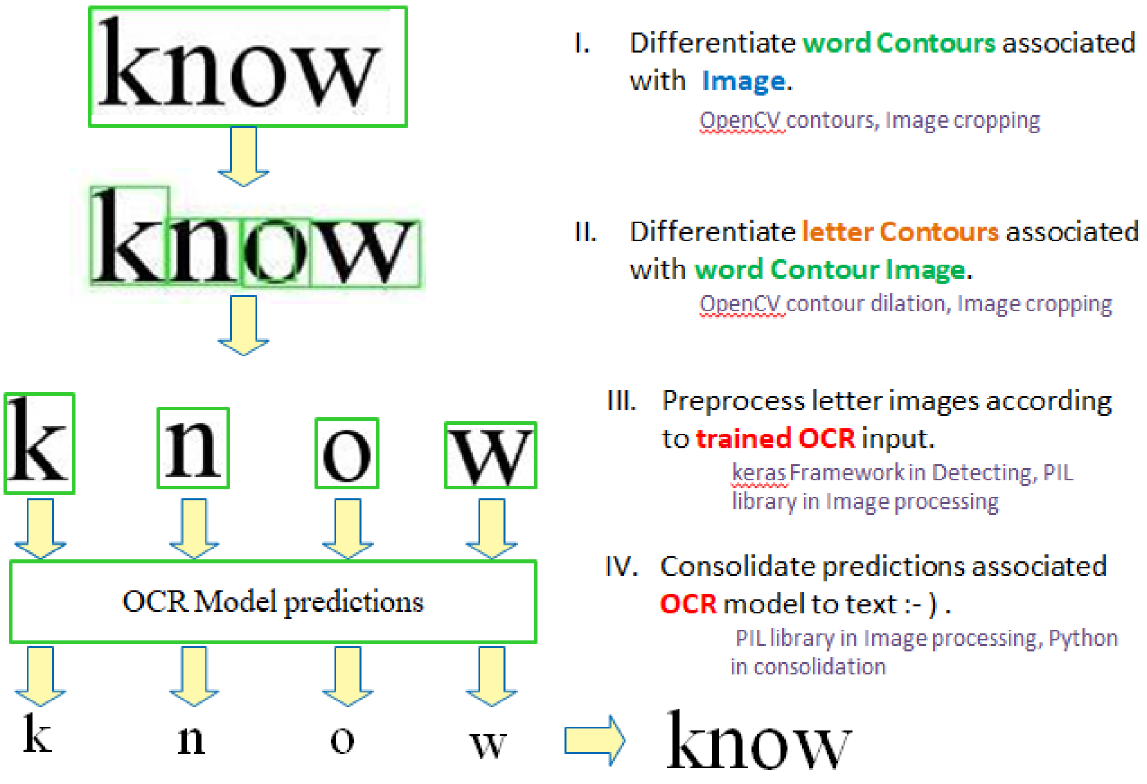
First the model will train on the Individual Character Images with direct Classification to predict the Images with softmax Classification of Character Categories.

Second the model is same model with last before layer as predictor which will Calculate a Embedding of specified Flatten Neurons (The Predicted flatten Values will have Feature Information of Receipt Images).

3. Load Trained ConvNet OCR model:

Optical Character recognition last step involves pre-processing of image into specific word related contours and letter contours, followed by prediction and consolidating according to letter and word related contours in an image.

once after training the model, we can save and load the pre-trained Optical character recognition model.



I. Differentiate word Contours associated with Image.
OpenCV contours, Image cropping

II. Differentiate letter Contours associated with word Contour Image.
OpenCV contour dilation, Image cropping

III. Preprocess letter images according to trained OCR input.
keras Framework in Detecting, PIL library in Image processing

IV. Consolidate predictions associated OCR model to text :-) .
PIL library in Image processing, Python in consolidation

Figure 3.14: OCR flow diagram.

4) Test and Consolidate Predictions of OCR:

Consolidate predictions involves, assigning specific ID to each word related contour with the line associated with the word in image, consolidating all predictions in a sorted series of specific word related contour and letters associated word.

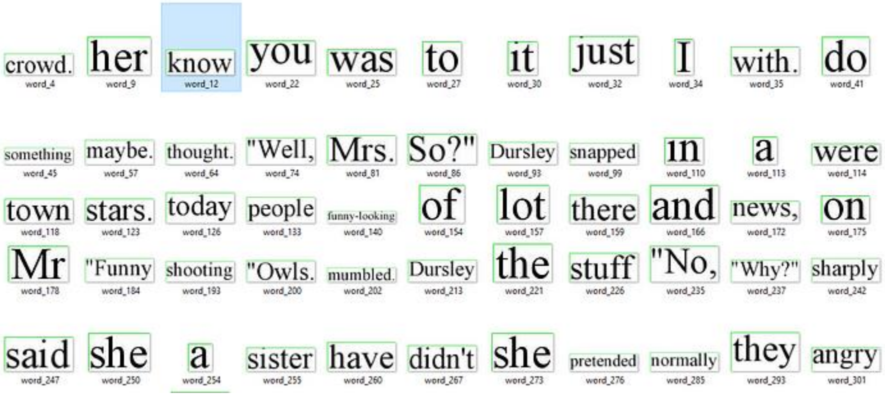


Figure 3.15: Obtaining contour for given dataset.

Automatic Language Identification: The Cloud Vision API can automatically detect the language of the text, which can be useful for multilingual applications or for processing documents in different languages.

Text Annotation: This feature provides information about the text detected in an image, including its location, size, and orientation. It can also recognize different types of text, such as headings, captions, and body text.

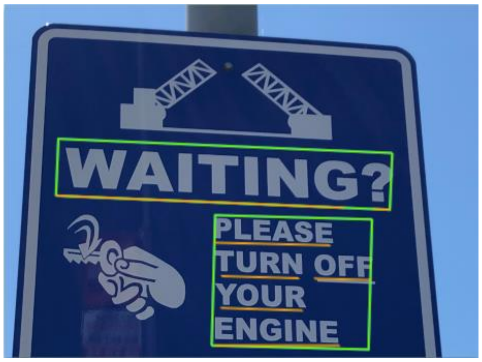


Figure 3.16: Text detection using Vision API.

3.4 GOOGLE TRANSLATE

Google Translate is a free online language translation service provided by Google. It uses artificial intelligence and machine learning techniques to automatically translate text, speech, images, and web pages from one language to another. The service supports over 100 languages, including English, Tamil, Spanish, French, Chinese, Japanese, and many more.

Google Translate works by analyzing patterns in documents that have already been translated by humans and identifying similar patterns in new texts. The system then uses these patterns to create translations that are as accurate as possible. Users can enter text or upload documents to be translated, or use the microphone on their device to translate spoken language in real-time.

Steps and algorithms involved in the Google Translate process:

- **Text Analysis:** Google Translate first analyzes the input text using various algorithms to identify the language and the context of the text. This involves parsing the text, identifying key phrases and grammar structures, and using statistical models to identify the language.
- **Language Model:** Once the language of the input text is identified, Google Translate uses a language model to identify the most probable translations of the input text into the desired language. This language model is based on statistical techniques and large amounts of training data, including previously translated text and bilingual text corpora.
- **Machine Translation:** After identifying the most probable translations, Google Translate uses machine translation algorithms to create a preliminary translation. These algorithms use statistical models and neural networks to

learn from vast amounts of training data and improve the quality of translations.

- **Neural Machine Translation:** In recent years, Google Translate has shifted to a neural machine translation approach, which uses deep learning algorithms to produce more accurate translations. This approach involves training a neural network on large amounts of bilingual text and then using this network to generate translations.
- **Post-Processing:** After the initial translation is generated, Google Translate uses a series of post-processing algorithms to improve the quality of the translation. These algorithms include error correction, grammar checking, and phrase reordering to ensure that the translated text is accurate and readable.
- **Evaluation:** Google Translate uses a range of evaluation techniques to assess the quality of translations, including human evaluations, automated metrics, and feedback from users. This feedback is used to improve the quality of the translation system and make it more accurate over time.

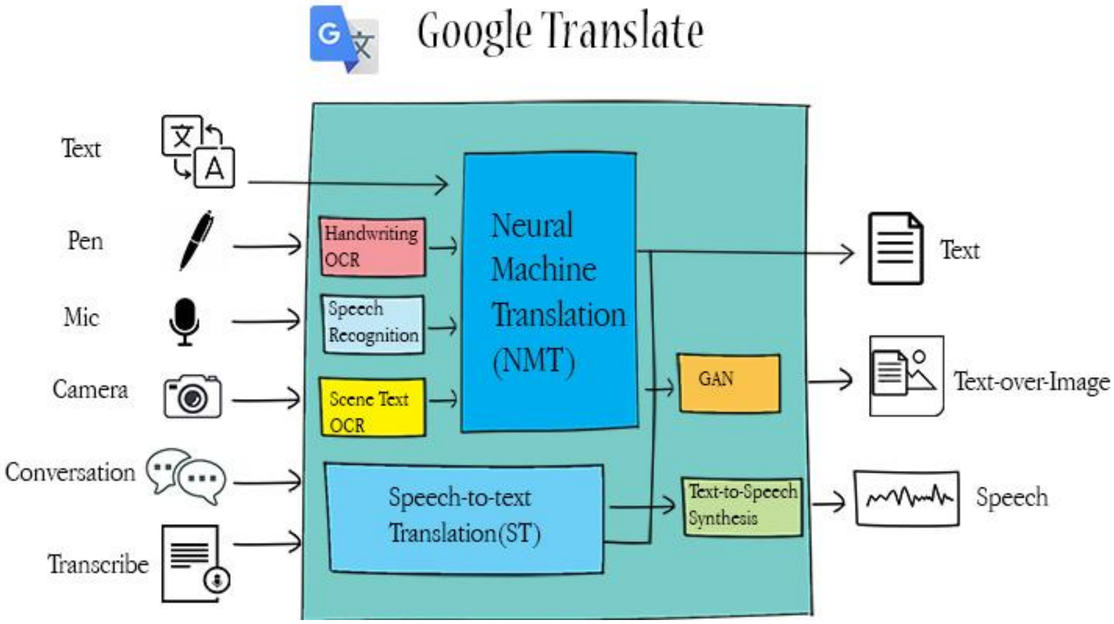


Figure 3.17: Google translate block diagram

3.5 GOOGLE CLOUD TEXT TO SPEECH

Text to speech (TTS) is a technology that allows computers to convert written text into spoken words. The Google Cloud Text-to-Speech API is a cloud-based service that enables developers to add TTS functionality to their applications. The API uses advanced machine learning algorithms to provide high-quality, natural-sounding synthesized speech in over 30 different languages. The Text-to-Speech API supports a wide range of text input formats, including plain text, SSML (Speech Synthesis Markup Language), and Audio Config. The API can also handle various speaking styles and voice preferences, allowing developers to customize the speech output according to their needs.

The Cloud Text-to-Speech API uses deep neural network models to generate synthesized speech. These models are trained on large datasets of speech and text data to produce high-quality, natural-sounding speech that is indistinguishable from human speech. The API can also generate speech in real-time, making it ideal for applications that require fast and responsive speech output.

In addition to its TTS capabilities, the Cloud Text-to-Speech API also includes several other features, such as support for multiple languages, voices, and speaking styles, as well as the ability to control the volume, speed, and pitch of the synthesized speech. The API also provides extensive documentation, sample code, and support resources, making it easy for developers to integrate TTS functionality into their applications.

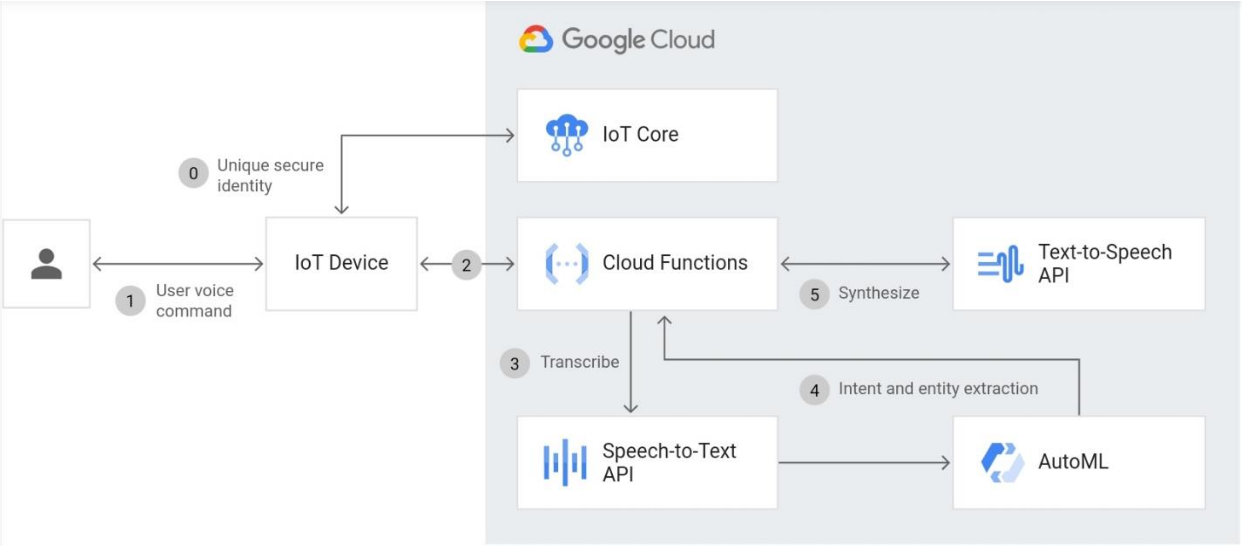


Figure 3.18: Google Cloud TTS block diagram

Steps and algorithms involved in the Google Cloud TTS:

- **Text Analysis:** The input text is first analyzed to identify the language, punctuation, and other features that will affect the pronunciation of the text. This involves processing the text to identify words, phrases, and sentences, and analyzing the context of the text to ensure that the pronunciation is accurate.
- **Speech Synthesis:** After analyzing the text, Google Cloud Text-to-Speech uses advanced speech synthesis algorithms to convert the text into audio. These algorithms are based on neural network models that are trained on large amounts of data, including recordings of human speech and text transcripts.
- **Waveform Generation:** Once the audio is synthesized, Google Cloud Text-to-Speech uses waveform generation algorithms to create a high-quality audio output. These algorithms ensure that the audio has a natural-sounding tone and pitch, and that it is free from artifacts such as noise or distortion.

- **Customization:** Google Cloud Text-to-Speech also provides a range of customization options that allow developers to adjust the speed, pitch, and volume of the audio output. Developers can also use SSML (Speech Synthesis Markup Language) to add pauses, emphasis, and other effects to the audio.
- **API Integration:** Google Cloud Text-to-Speech provides an API that allows developers to easily integrate speech synthesis functionality into their applications. The API supports a range of programming languages, such as Python, Java, and Ruby, and allows developers to send text to the service and receive audio output in a range of formats.
- **Evaluation:** Google Cloud Text-to-Speech uses a range of evaluation techniques to assess the quality of the audio output, including human evaluations, automated metrics, and feedback from users. This feedback is used to improve the quality of the speech synthesis algorithms over time.

3.5.1 FEATURES

Table 3.1: Features of Google Cloud TTS

Custom Voice (beta)	Train a custom speech synthesis model using your own audio recordings to create a unique and more natural sounding voice for your organization. You can define and choose the voice profile that suits your organization and quickly adjust to changes in voice needs without needing to record new phrases
Voice and language selection	Choose from an extensive selection of 220 voices across 40-languages and variants, with more to come soon.

WaveNet voices	Take advantage of 90+ Wavelet voices built based on DeepMind's ground breaking research to generate speech that significantly closes the gap with human performance.
Speaking rate tuning	Adjust your speaking rate to be 4x faster or slower than the normal rate.
Pitch tuning	Personalize the pitch of your selected voice, up to 20 semitones more or less than the default.
Text and SSML support	Customize your speech with SSML tags that allow you to add pauses, numbers, date and time formatting, and other pronunciation instructions.
Volume gain control	increase the volume of the output by up to 16dbor decrease the volume up to -96db.
Integrated REST and gRPC APIs	Easily integrate with any application or device that can send a REST or gRPC request including phones, PCS, tablets, and loT devices (e.g, cars, TVs, speakers).
Audio format flexibility	Convert text to MP3, Linear16, OGG Opus, and a number of other audio formats.
Audio profiles	Optimize for the type of speaker from which your speech is intended to play, such as headphones or phone lines.

3.7 DEVICE IMPLEMENTATION:

3.7.1 SOFTWARE:

The operating system under which the proposed project is executed in Raspberry Pi OS (64bit) which is derived from the Debian operating system. The program is written using the python language. The functions in algorithm are called from the OpenCV Library. OpenCV is an opensource computer vision library, which is written under C and C++ and runs under Linux, Windows and Mac OS X. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. OpenCV is written in optimized C and can take advantage of multi-core processors.

Installation:

- Download the latest OS for the Raspberry Pi using Pi imager and write the file into the sdcard.
    - ❖ Raspberry Pi OS (64-bit)
    - ❖ System: 64-bit
    - ❖ Kernel version: 6.1
    - ❖ Debian version: 11 (bullseye)
- By default, Python 3.9 is installed on your OS and install VS Code (Python IDE) on the raspberry pi 4.
- Install the required libraries for this project,
    - ✓ pip install opencv-python
    - ✓ pip install google-cloud-vision
    - ✓ pip install googletrans-py
    - ✓ pip install google-cloud-texttospeech
    - ✓ pip install pygame

Steps and Programming explanation:

- Set up the necessary software and libraries required for the project. This includes installing Python, the picamera module, Google Cloud Vision API, Google Cloud Text-to-Speech API, googletrans, and pygame.
- Import the necessary libraries in your Python script:

```
import picamera

from google.cloud import vision

from google.cloud import translate_v2 as translate

from google.cloud import texttospeech_v1 as tts

import pygame
```

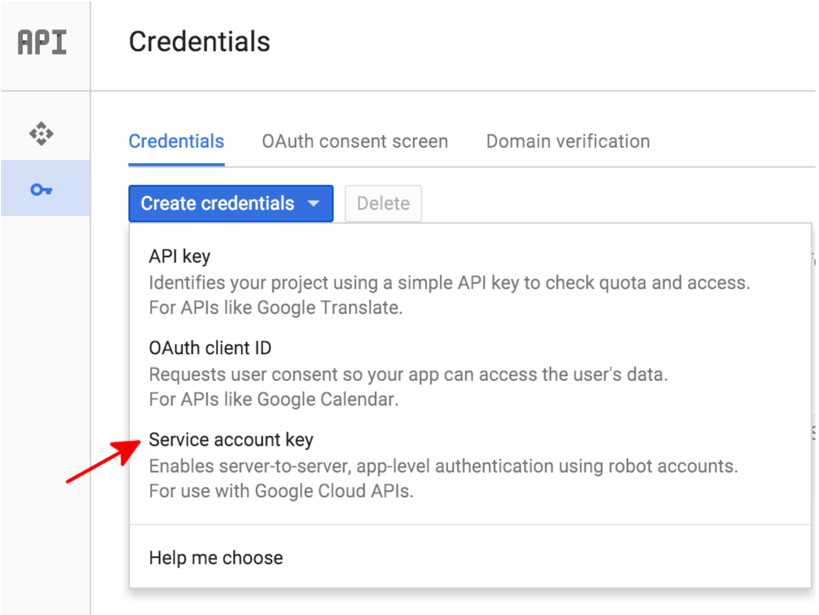
These libraries are required for the various functionalities of the project:

- **picamera** is used to capture images with the Raspberry Pi camera module.
- **google.cloud.vision** is used to perform Optical Character Recognition (OCR) on the captured image to extract text.
- **google.translate** is used to translate the extracted text into different languages.
- **google.cloud.texttospeech_v1** is used to convert the translated text to speech.
- **pygame** is used to play the audio file generated by the text-to-speech conversion.

- Set up the API credentials for the Google Cloud Vision API, Google Translate API, and Google Cloud Text-to-Speech API. You can store these credentials in a separate configuration file and load them in your Python script using os and json libraries:

  ```
  import os
  import json
  os.environ['GOOGLE_APPLICATION_CREDENTIALS'] = 'path/to/credentials.json'
  with open('path/to/credentials.json', 'r') as f:
  creds = json.load(f)
  ```

This sets the GOOGLE_APPLICATION_CREDENTIALS environment variable to the path of the JSON file containing the authentication credentials for the Google Cloud project. This is necessary for the other parts of the project to access the Google Cloud APIs. This opens the JSON file containing the authentication credentials in read mode, and loads its contents into a Python dictionary called creds. This dictionary contains the credentials needed to authenticate with the Google Cloud APIs, such as the project ID and the private key. The path/to/credentials.json should be replaced with the actual path to the JSON file containing the credentials.



API key
Identifies your project using a simple API key to check quota and access.
For APIs like Google Translate.

OAuth client ID
Requests user consent so your app can access the user's data.
For APIs like Google Calendar.

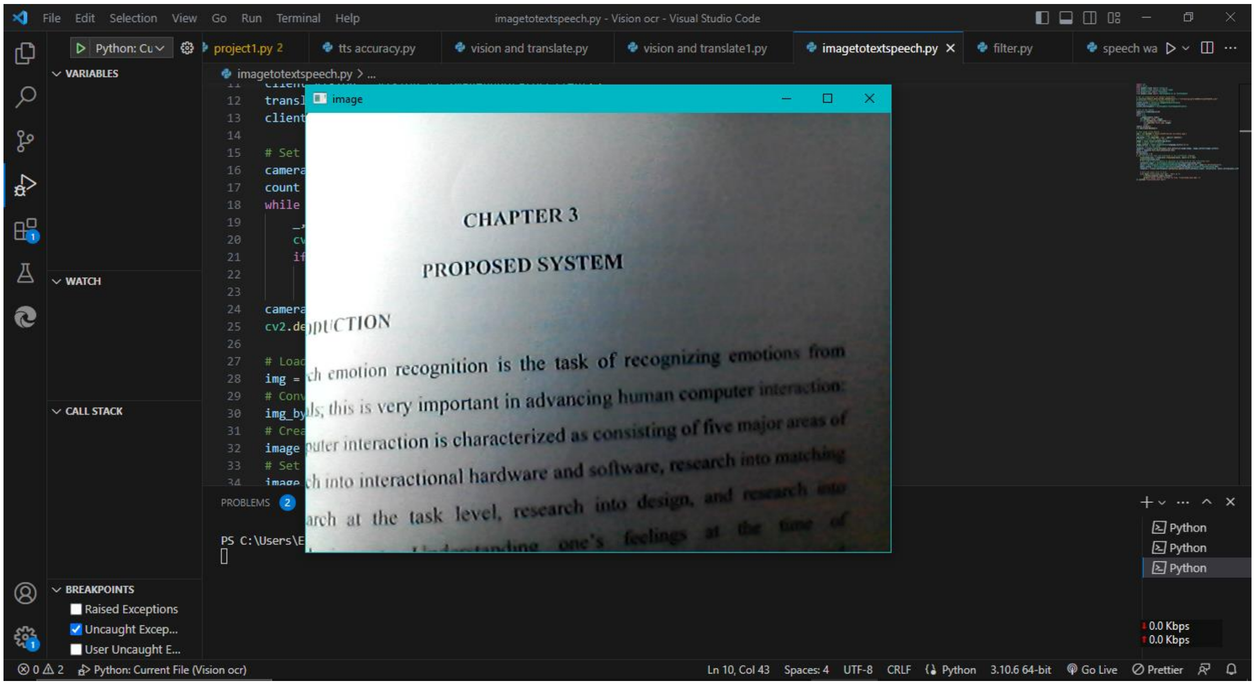
Service account key
Enables server-to-server, app-level authentication using robot accounts.
For use with Google Cloud APIs.

Figure 3.19: Activating service account key

- Set up the Pi Camera module and capture an image. You can use the picamera library to do this:

```
with picamera.PiCamera() as camera:
     camera.start_preview()
    camera.capture('path/to/image.jpg')
    camera.stop_preview()
```

This code uses the ‘picamera’ module in Python to capture an image using the Pi Camera module. The with statement ensures that the camera is properly released after use. The resolution parameter sets the resolution of the captured image, and ‘start_preview()’ starts the camera preview on the Raspberry Pi. Finally, camera.capture('image.jpg') captures the image and saves it to a file named image.jpg.



```
transl
client

# Set
camera
count
while
    _,
    cv
    if


camera
cv2.de

# Load
img =
# Conv
img_by
# Crea
image
# Set
image
```

```
PS C:\Users\E
```

CHAPTER 3

PROPOSED SYSTEM

)DUCTION

ch emotion recognition is the task of recognizing emotions from
ls; this is very important in advancing human computer interaction:
puter interaction is characterized as consisting of five major areas of
ch into interactional hardware and software, research into matching
arch at the task level, research into design, and research into
Understanding one's feelings at the time of

Figure 3.20: Camera capturing the image

- Process the image using OCR to extract the text from the image. You can use the google-cloud-vision library to do this:

```
client = vision.ImageAnnotatorClient(credentials=creds)

with open('path/to/image.jpg', 'rb') as image_file:

    content = image_file.read()
```

```
image = vision.types.Image(content=content)

response = client.text_detection(image=image)

texts = response.text_annotations
```

This code uses the google-cloud-vision library to interact with the Google Cloud Vision API and extract text from the captured image using OCR. The with statement opens the image.jpg file in binary mode and reads its content into a variable named content. The types.Image class creates an image object from the content variable, which is then passed as a parameter to the text_detection() method of the ImageAnnotatorClient class. The response variable contains the OCR results, and the text_annotations[0].description statement extracts the text from the first result.

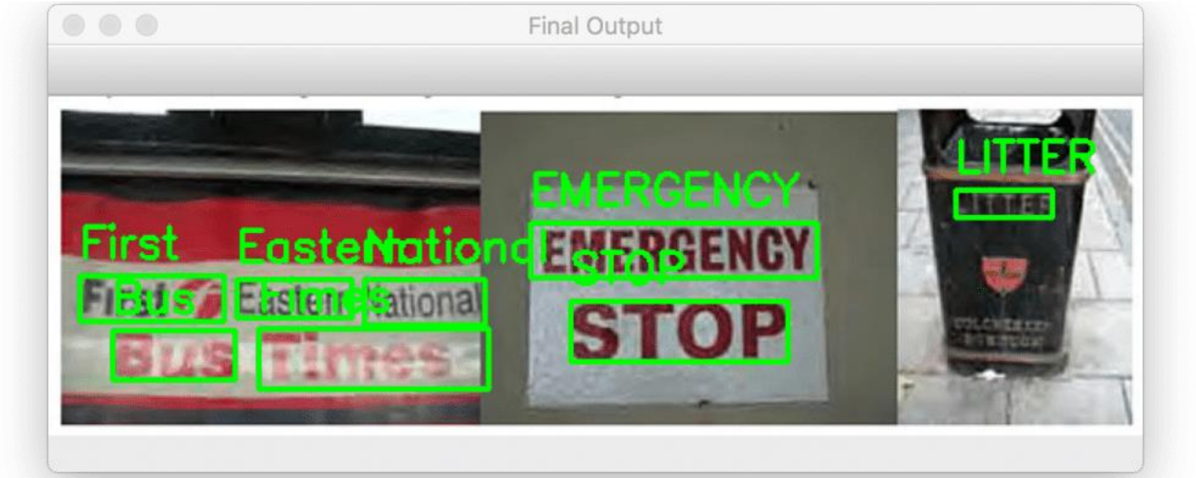


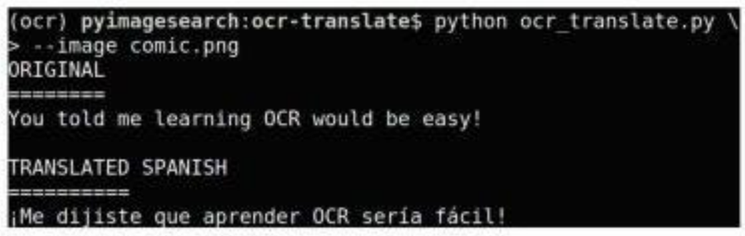
Figure 3.21: Extracted text output

- Translate the extracted text to your desired language. You can use the googletrans library to do this:

```
translator = translate.Client(credentials=creds)

text = texts[0].description
```

```
translation = translator.translate(text, target_language='en')
```

This code uses the googletrans library to interact with the Google Translate API and translate the extracted text to the desired language (in this case, English). The Translator() class creates a translator object, which is then used to call the translate() method with the text variable and the dest parameter set to 'en' (for English). The text property of the resulting translation object contains the translated text.



```
(ocr) pyimagesearch:ocr-translate$ python ocr_translate.py \
> --image comic.png
ORIGINAL
========
You told me learning OCR would be easy!

TRANSLATED SPANISH
==========
¡Me dijiste que aprender OCR sería fácil!
```

Figure 3.22: Translated text output

- Send the translated text to the Google Cloud Text-to-Speech API to generate speech. You can use the google-cloud-texttospeech library to do this:

```
tts_client = tts.TextToSpeechClient(credentials=creds)

synthesis_input = tts.types.SynthesisInput(text=translation['translatedText'])

voice = tts.types.VoiceSelectionParams (language_code ='en-US', ssml_gender = tts.enums.SsmlVoiceGender.NEUTRAL)

audio_config = tts.types.AudioConfig (audio_encoding = tts.enums.AudioEncoding.MP3)

response = tts_client.synthesize_speech(synthesis_input, voice, audio_config)

with open('path/to/output.mp3', 'wb') as out:

    out.write(response.audio_content)
```

This code uses the google cloud textospeech library to interact with the Google Cloud Text-to-Speech API. The ‘TextToSpeechClient’ class, which provides an interface to the Google Cloud Text-to-Speech API. The ‘SynthesisInput’ object, which specifies the input text to be synthesized. The text parameter is set to translated_text, which contains the translated text. The ‘VoiceSelectionParams’ object, which specifies the voice to be used for the synthesized audio. In this case, the language_code parameter is set to 'en-US', which specifies English as the language, and the ‘ssml_gender’ parameter is set to ‘texttospeech. SsmlVoiceGender.NEUTRAL’, which specifies a neutral voice. The ‘AudioConfig’ object, which specifies the audio format to be used for the synthesized audio. In this case, the ‘audio_encoding’ parameter is set to texttospeech.AudioEncoding.MP3, which specifies the MP3 audio format. This calls the ‘synthesize_speech()’ method of the ‘TextToSpeechClient’ object to synthesize the speech. The input, voice, and ‘audio_config’ parameters specify the input text, voice, and audio format, respectively. The resulting response object contains the synthesized audio. This writes the synthesized audio to an MP3 file named output.mp3 using a with statement. The ‘audio_content’ property of the response object contains the actual audio data, which is written to the file.

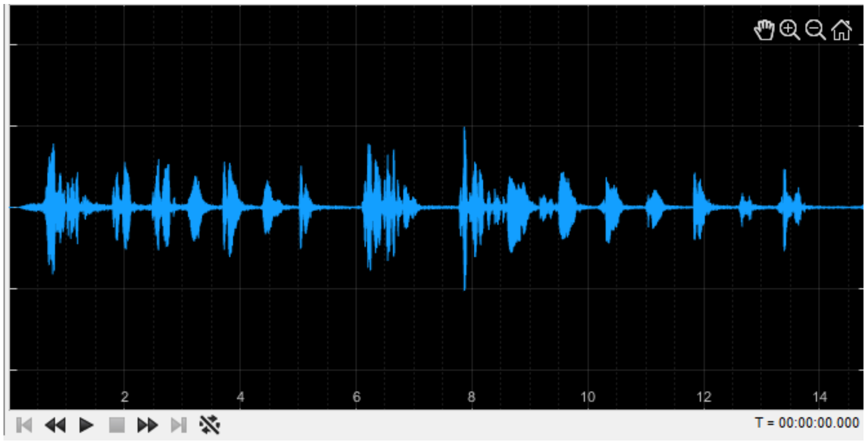


Figure 3.23: Generated speech output

- Play the generated speech through a speaker. You can use the pygame library to do this:

```
pygame.mixer.init()

pygame.mixer.music.load('path/to/output.mp3')

pygame.mixer.music.play()

while pygame.mixer.music.get_busy():

    continue
```

The ‘pygame.mixer’ module, which is used for handling audio playback in ‘Pygame’. Then loads an audio file into memory for playback. The path to the audio file should be replaced with the actual path to the generated output file and plays the loaded audio file. Then waits for the audio file to finish playing before continuing with the rest of the program. The ‘pygame.mixer.music.get_busy()’ function returns True if the mixer is still playing audio, so the while loop continues running until the audio finishes playing. The continue keyword just means to skip to the next iteration of the loop.

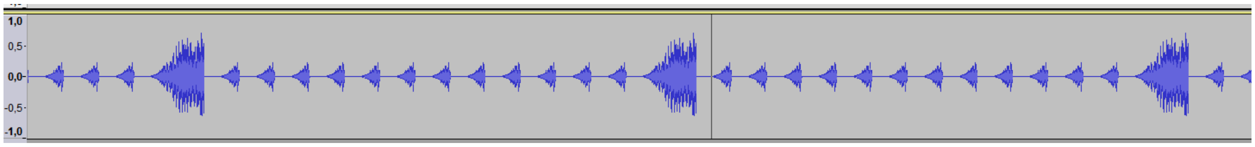


Figure 3.24: Playing speech output

- You may also want to add some error handling and user interface to the project. For example, you can use a button to trigger the image capture and text-to-speech generation.

- Finally, run the Python script and test the project. When you press the button, the Pi Camera module will capture an image, process it using OCR, translate the text, and generate speech that will be played through the speaker.

3.7.2 Hardware:

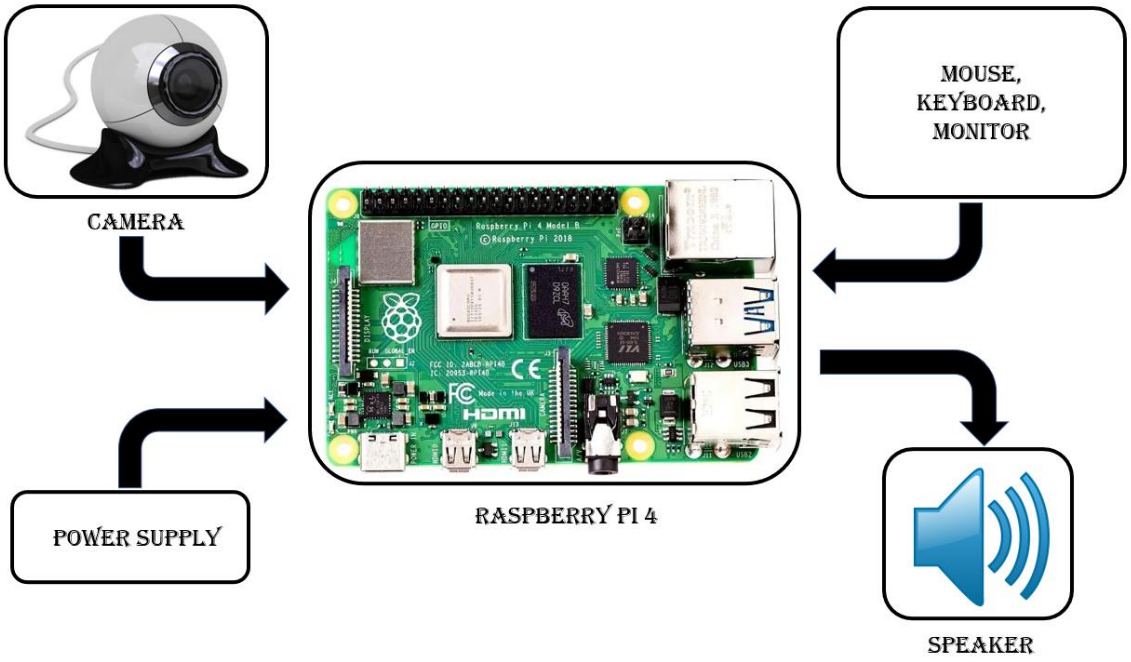


Figure 3.25: Block diagram of the device

Raspberry Pi 4:

The Raspberry Pi 4 Model B is a powerful single-board computer that was released by the Raspberry Pi Foundation in 2019. It is an upgrade over its predecessor, the Raspberry Pi 3 Model B+, and comes with a number of significant improvements and features. One of the most notable improvements in the Raspberry Pi 4 Model B is its processor. It is powered by a Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit system-on-chip (SoC) running at 1.5 GHz. This makes it significantly faster and more capable than its predecessor, making it suitable for a wider range of applications. In terms of memory, the Raspberry Pi 4 Model B comes with up to 4GB LPDDR4-3200 SDRAM, which is a significant upgrade over the 1GB of RAM that was available in the Raspberry Pi 3 Model B+. This makes it more capable of handling complex and memory-intensive applications. The Raspberry Pi 4 Model B also comes with faster networking capabilities, with a Gigabit Ethernet port and support for dual-band 802.11ac wireless networking. It also has Bluetooth 5.0 and BLE (Bluetooth Low Energy) support. Another significant upgrade in the Raspberry Pi 4 Model B is its video and display capabilities. It comes with two micro-HDMI ports, both of which can support 4Kp60 video output. This makes it suitable for use as a media centre or for other applications that require high-quality video output. Overall, the Raspberry Pi 4 Model B is a versatile and powerful single-board computer that is suitable for a wide range of applications. Its improved processing power, memory, networking, and video capabilities make it a significant upgrade over its predecessor, and it remains a popular choice for hobbyists, students, and professionals alike.

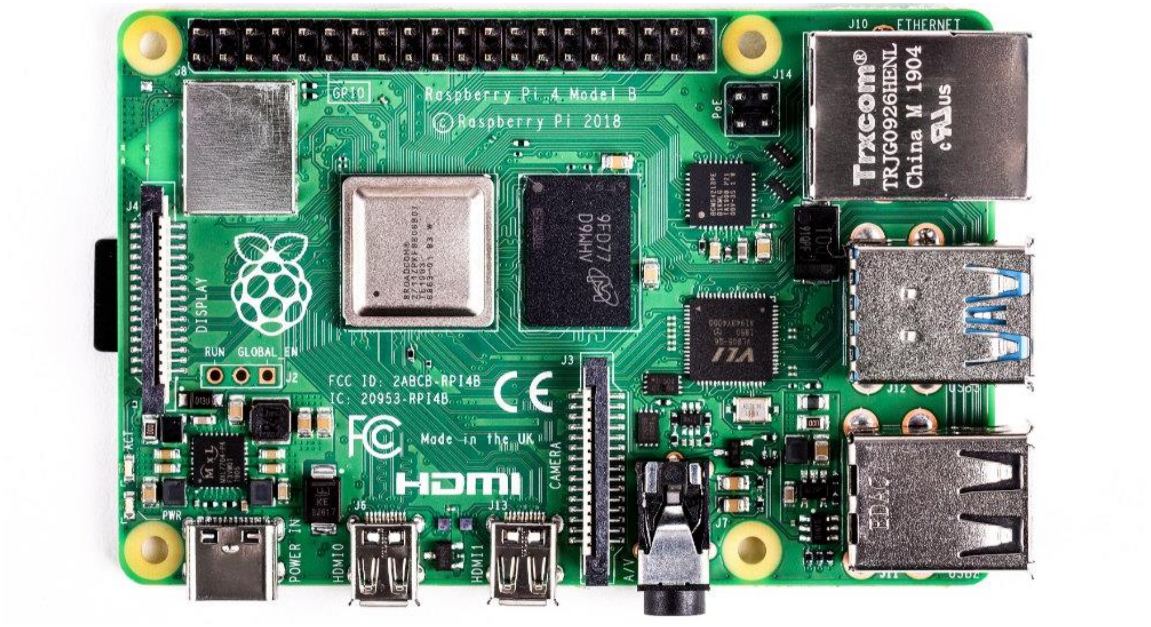


Figure 3.26: Raspberry Pi 4 Model B

Raspberry Pi Camera:

The Raspberry Pi Camera Module 3 is a small and lightweight camera designed specifically for use with the Raspberry Pi single-board computer. It offers a range of features and capabilities that make it suitable for a wide range of applications, including robotics, home automation, and security systems. The Raspberry Pi Camera is easy to connect to a Raspberry Pi computer. It connects to the Raspberry Pi's CSI (Camera Serial Interface) port, which is a ribbon cable connector located near the HDMI port on the Raspberry Pi board. The camera module itself is a small board that measures just 25mm x 20mm x 9mm, making it easy to integrate into a wide range of projects.

Here are some key features and components of the Raspberry Pi Camera Module 3:

Image Sensor: The camera module features a 5-megapixel OmniVision image sensor that is capable of capturing high-quality images and video.

Lens: The camera module comes with a fixed-focus lens that provides a wide-angle field of view. It is also possible to purchase additional lenses that offer different focal lengths and fields of view.

Image Processing: The camera module features a dedicated image processing chip that performs advanced image processing and correction. This chip helps to improve the quality of the images and video captured by the camera module.

Connectivity: The camera module connects to the Raspberry Pi via a ribbon cable that provides power and data connectivity. It is also possible to connect multiple camera modules to a single Raspberry Pi, allowing for the creation of stereo or multi-camera setups.

Software: The camera module is supported by a range of software tools and libraries, including the Raspberry Pi Camera Module Python library. This library allows developers to easily access the camera module from their Python code, and provides a range of features and capabilities for capturing images and video.

Applications: The Raspberry Pi Camera Module 3 is suitable for a wide range of applications, including home security systems, robotics projects, and time-lapse photography. Its compact size and low power consumption make it ideal for use in embedded systems and other applications where space and power are limited.

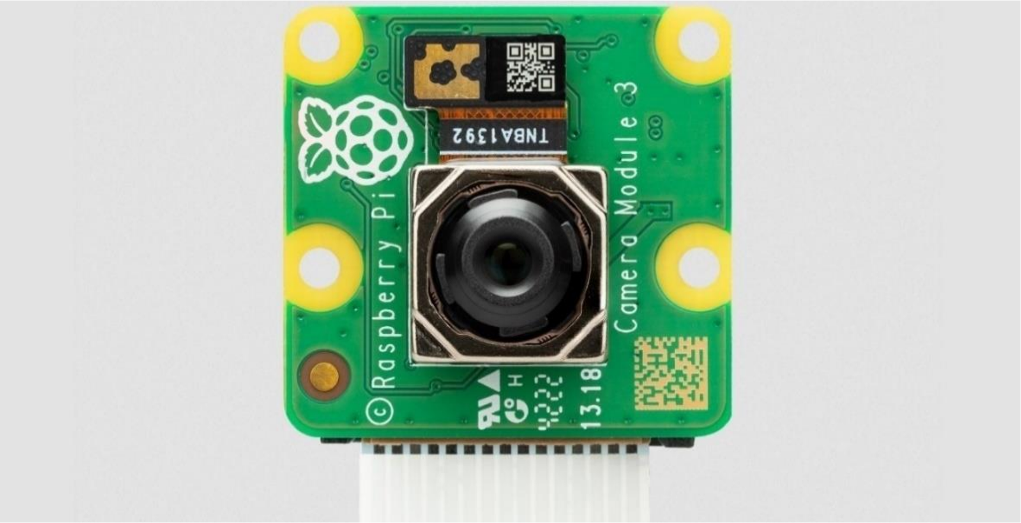


Figure 3.27: Pi camera module V3

Hardware components:

- Raspberry Pi 4
- PiCamera module V3
- Power Adaptor for Raspberry Pi 4
- USB or Bluetooth speaker
- HDMI monitor (optional)
- Keyboard and Mouse (optional)

Connections:

- Connect the Power supply adaptor to the Raspberry Pi 4 via Type-C port.
- Connect the PiCamera module to the Raspberry Pi 4 via the camera connector.
- Connect the USB speaker to the Raspberry Pi 4 via the USB port.
- Connect the Monitor to the Raspberry Pi 4 via HDMI cable. (optional)
- Connect the Keyboard and Mouse to the Raspberry Pi 4 via USB port.

Steps:

- Set up the Raspberry Pi 4 with an operating system such as Raspbian or Ubuntu. Ensure that the Raspberry Pi 4 is updated with the latest software updates and packages. Install Python on the Raspberry Pi 4.
- Connect the PiCamera module to the Raspberry Pi 4 using the camera connector. Ensure that the camera is enabled in the Raspberry Pi configuration. Test the camera by running the "libcamera -hello" command to take a picture.
- Install the necessary Python libraries for the project. This includes the "picamera" module, "google-cloud-vision" library, "googletrans" library, and "google-cloud-texttospeech" library.
- Write a Python script to capture the image from the camera. Use the "picamera" module to take a picture and save it to a file.
- Use the "google-cloud-vision" library to process the image and extract text from it using OCR (optical character recognition). The library provides a function to detect text in an image, and you can use it to get the text in the image.
- Translate the extracted text to your desired language using the "googletrans" library. The library provides a function to translate text to different languages.

- Send the translated text to the "google-cloud-texttospeech" library to generate speech. The library provides a function to generate speech from text.
- Play the generated speech using the USB speaker connected to the Raspberry Pi 4. You can use the "pygame" library to play audio on the Raspberry Pi.
- You can also display the captured image, extracted text, and translated text on an HDMI monitor connected to the Raspberry Pi 4.
- Using VNC viewer to display the monitor in headless mode which means does not have wire connection between monitor, keyboard, and mouse.

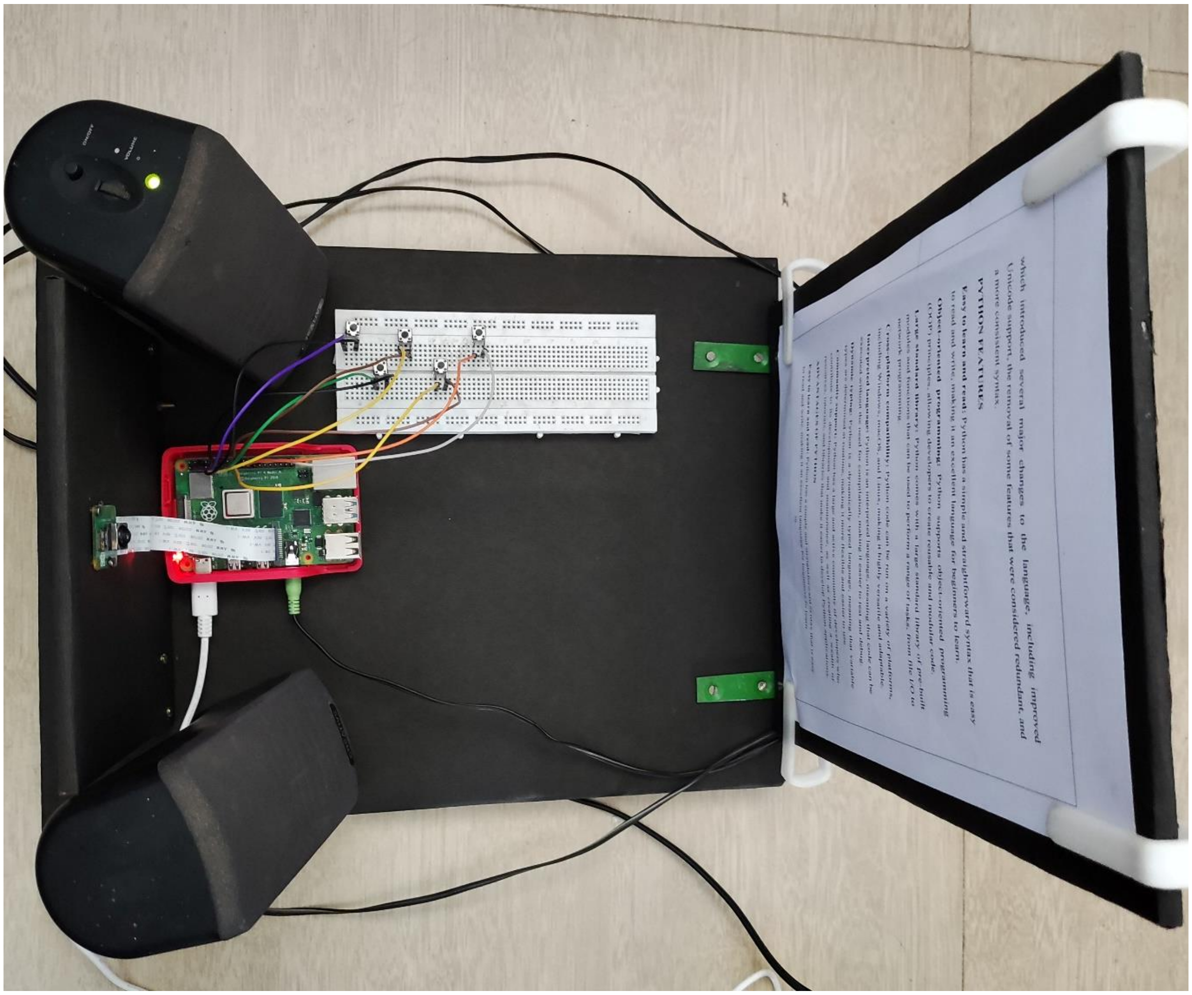


Figure 3.28: Hardware implementation

CHAPTER 4

RESULTS AND DISCUSSION

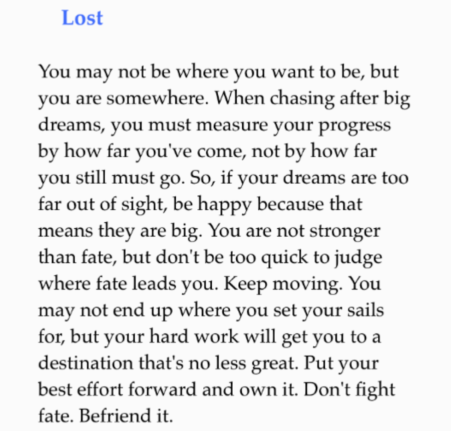
4.1 GENERAL

Our project is a text-to-speech device that takes an input image, performs OCR to extract the text, translates the text to a specified language, and converts the translated text to speech using Google Cloud APIs. The project is implemented using Python on a Raspberry Pi 4 with a Picamera module. Overall, the project is a good demonstration of the capabilities of machine learning and cloud computing technologies. The use of OCR and translation APIs allows the system to be easily adapted to work with a variety of languages, making it useful for a wide range of applications. Limitation of the system is not correctly recognizing all the symbols, which can affect the accuracy of OCR. However, using Google Cloud APIs helps to mitigate this limitation, as they are constantly updated and improved with new features and enhancements. Another limitation is the cost of using the Google Cloud APIs, as they require a subscription and may incur usage fees depending on the amount of usage. However, the cost can be minimized by optimizing the use of the APIs and leveraging free usage quotas that come with a Google Cloud account. In summary, our project is a practical application of machine learning and cloud computing technologies that demonstrates the power of these technologies to solve real-world problems. With further improvements and optimizations, this system can be used in a variety of contexts, such as language learning, accessibility, and automation. We have analysed the accuracy of Google Cloud Vision API, Google Translate, Google Cloud Text-to-Speech was shown in Tabulation section 4.1 & 4.2. Results of our project for text to speech was shown in Simulation Result section.

4.2 SIMULATION OUTPUT

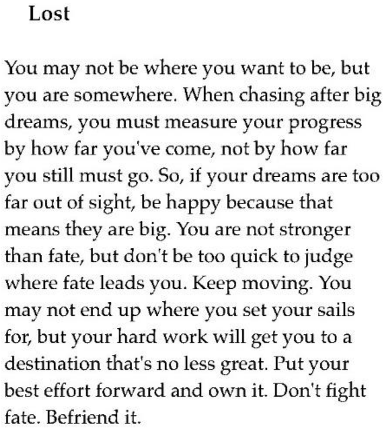
Test Case 1: Input image contain printed English text

Input Image: **Pre-processing:**



Lost

You may not be where you want to be, but you are somewhere. When chasing after big dreams, you must measure your progress by how far you've come, not by how far you still must go. So, if your dreams are too far out of sight, be happy because that means they are big. You are not stronger than fate, but don't be too quick to judge where fate leads you. Keep moving. You may not end up where you set your sails for, but your hard work will get you to a destination that's no less great. Put your best effort forward and own it. Don't fight fate. Befriend it.



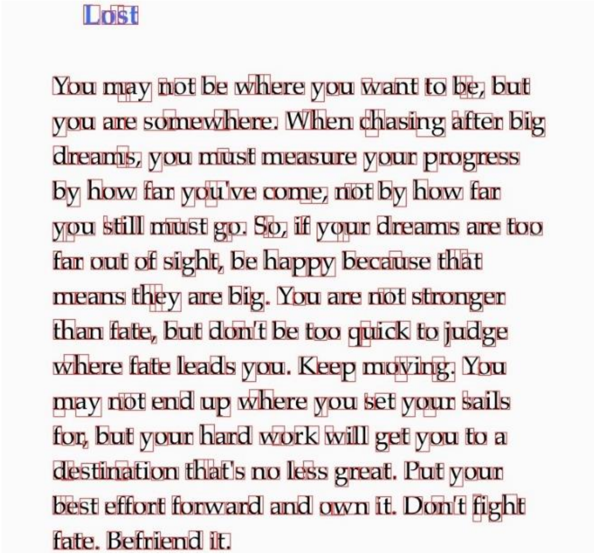
Lost

You may not be where you want to be, but you are somewhere. When chasing after big dreams, you must measure your progress by how far you've come, not by how far you still must go. So, if your dreams are too far out of sight, be happy because that means they are big. You are not stronger than fate, but don't be too quick to judge where fate leads you. Keep moving. You may not end up where you set your sails for, but your hard work will get you to a destination that's no less great. Put your best effort forward and own it. Don't fight fate. Befriend it.

Figure 4.1: Input image1 **Figure 4.2: Pre-processed image1**

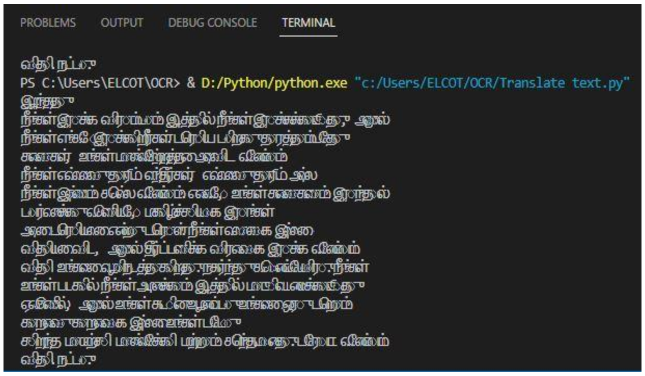
Camera captures the input image contain printed English text and then the captured image is Pre-processed by using open cv. Pre-processed image sends to vision OCR for Boundary detection and extracted text translate into Tamil text.

Boundary Detection: **Translation:**



Lost

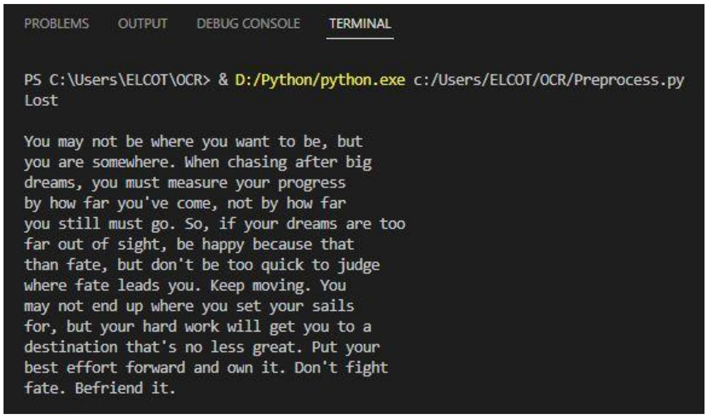
You may not be where you want to be, but you are somewhere. When chasing after big dreams, you must measure your progress by how far you've come, not by how far you still must go. So, if your dreams are too far out of sight, be happy because that means they are big. You are not stronger than fate, but don't be too quick to judge where fate leads you. Keep moving. You may not end up where you set your sails for, but your hard work will get you to a destination that's no less great. Put your best effort forward and own it. Don't fight fate. Befriend it.



```
விதி நட்பு
PS C:\Users\ELCOT\OCR> & D:/Python/python.exe "c:/Users/ELCOT/OCR/Translate text.py"
இழந்தது
நீங்கள் இருக்க விரும்பும் இடத்தில் நீங்கள் இருக்கக்கூடாது, ஆனால்
நீங்கள் எங்கோ இருக்கிறீர்கள். பெரிய கனவுகளைத் துரத்தும்போது
கனவுகள், உங்கள் முன்னேற்றத்தை அளவிட வேண்டும்
நீங்கள் எவ்வளவு தூரம் வந்தீர்கள், எவ்வளவு தூரம் அல்ல
நீங்கள் இன்னும் செல்ல வேண்டும். எனவே, உங்கள் கனவுகள் இருந்தால்
பார்வைக்கு வெளியே, மகிழ்ச்சியாக இருங்கள்
அவை பெரியவை என்று பொருள். நீங்கள் வலுவாக இல்லை
விதியை விட, ஆனால் தீர்ப்பளிக்க விரைவாக இருக்க வேண்டாம்
விதி உங்களை எங்கு அழைத்துச் செல்கிறது. தொடர்ந்து செல்லுங்கள். நீங்கள்
உங்கள் படகில் நீங்கள் அமைக்கும் இடத்தில் முடிவடையக்கூடாது
ஏனெனில், ஆனால் உங்கள் கடின உழைப்பு உங்களைப் பெறும்
குறைவான குறைவாக இல்லை உங்கள் இடமோ
சிறந்த முயற்சி முன்னோக்கி மற்றும் சொந்தமாக்குங்கள். போராட வேண்டாம்
விதி நட்பு
```

Figure 4.3: Boundary Detected for image1 **Figure 4.4: Translated Output for image1**

Output Text:



```
PS C:\Users\ELCOT\OCR> & D:/Python/python.exe c:/Users/ELCOT/OCR/Preprocess.py
Lost

You may not be where you want to be, but
you are somewhere. When chasing after big
dreams, you must measure your progress
by how far you've come, not by how far
you still must go. So, if your dreams are too
far out of sight, be happy because that
than fate, but don't be too quick to judge
where fate leads you. Keep moving. You
may not end up where you set your sails
for, but your hard work will get you to a
destination that's no less great. Put your
best effort forward and own it. Don't fight
fate. Befriend it.
```

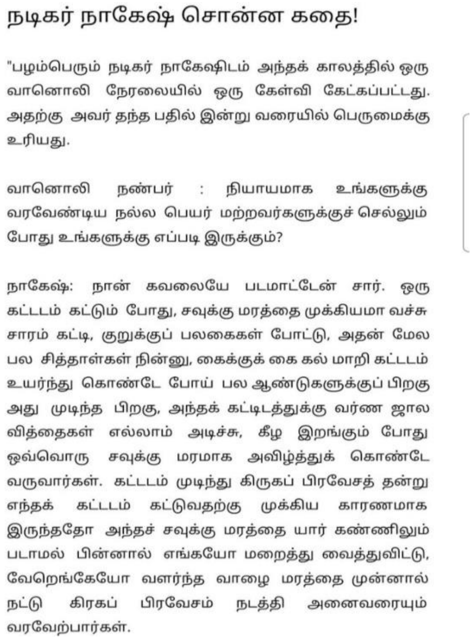
Speech generated using TTS

Figure 4.5: Output text for image1

The Output text contain extracted text from the input image1 using vision OCR.

Test Case 2: Input image contain printed Tamil text

Input Image:



நடிகர் நாகேஷ் சொன்ன கதை!

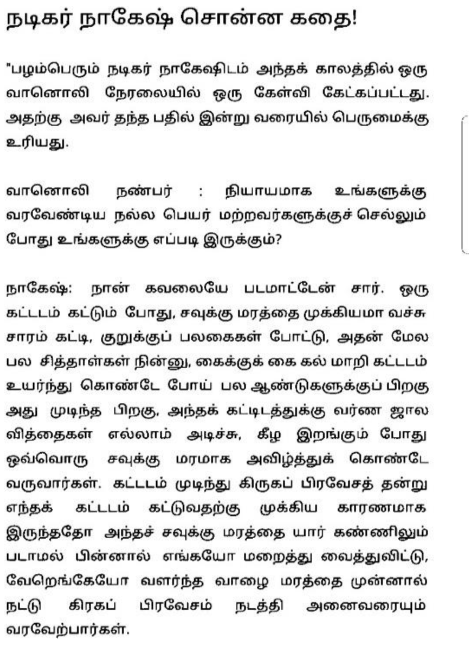
"பழம்பெரும் நடிகர் நாகேஷிடம் அந்தக் காலத்தில் ஒரு வானொலி நேரலையில் ஒரு கேள்வி கேட்கப்பட்டது. அதற்கு அவர் தந்த பதில் இன்று வரையில் பெருமைக்கு உரியது.

வானொலி நண்பர் : நியாயமாக உங்களுக்கு வரவேண்டிய நல்ல பெயர் மற்றவர்களுக்குச் செல்லும் போது உங்களுக்கு எப்படி இருக்கும்?

நாகேஷ்: நான் கவலையே படமாட்டேன் சார். ஒரு கட்டடம் கட்டும் போது, சவுக்கு மரத்தை முக்கியமா வச்சு சாரம் கட்டி, குறுக்குப் பலகைகள் போட்டு, அதன் மேல பல சித்தாள்கள் நின்னு, கைக்குக் கை கல் மாறி கட்டடம் உயர்ந்து கொண்டே போய் பல ஆண்டுகளுக்குப் பிறகு அது முடிந்த பிறகு, அந்தக் கட்டிடத்துக்கு வர்ண ஜால வித்தைகள் எல்லாம் அடிச்சு, கீழ இறங்கும் போது ஒவ்வொரு சவுக்கு மரமாக அவிழ்த்துக் கொண்டே வருவார்கள். கட்டடம் முடிந்து கிருகப் பிரவேசத் தன்று எந்தக் கட்டடம் கட்டுவதற்கு முக்கிய காரணமாக இருந்ததோ அந்தச் சவுக்கு மரத்தை யார் கண்ணிலும் படாமல் பின்னால் எங்கயோ மறைத்து வைத்துவிட்டு, வேறெங்கேயோ வளர்ந்த வாழை மரத்தை முன்னால் நட்டு கிரகப் பிரவேசம் நடத்தி அனைவரையும் வரவேற்பார்கள்.

Figure 4.6: Input image2

Pre-processing:



நடிகர் நாகேஷ் சொன்ன கதை!

"பழம்பெரும் நடிகர் நாகேஷிடம் அந்தக் காலத்தில் ஒரு வானொலி நேரலையில் ஒரு கேள்வி கேட்கப்பட்டது. அதற்கு அவர் தந்த பதில் இன்று வரையில் பெருமைக்கு உரியது.

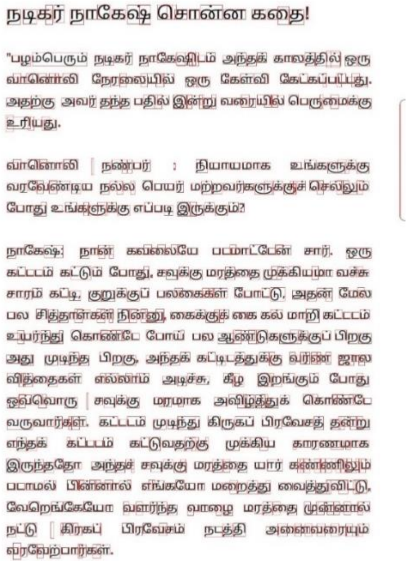
வானொலி நண்பர் : நியாயமாக உங்களுக்கு வரவேண்டிய நல்ல பெயர் மற்றவர்களுக்குச் செல்லும் போது உங்களுக்கு எப்படி இருக்கும்?

நாகேஷ்: நான் கவலையே படமாட்டேன் சார். ஒரு கட்டடம் கட்டும் போது, சவுக்கு மரத்தை முக்கியமா வச்சு சாரம் கட்டி, குறுக்குப் பலகைகள் போட்டு, அதன் மேல பல சித்தாள்கள் நின்னு, கைக்குக் கை கல் மாறி கட்டடம் உயர்ந்து கொண்டே போய் பல ஆண்டுகளுக்குப் பிறகு அது முடிந்த பிறகு, அந்தக் கட்டிடத்துக்கு வர்ண ஜால வித்தைகள் எல்லாம் அடிச்சு, கீழ இறங்கும் போது ஒவ்வொரு சவுக்கு மரமாக அவிழ்த்துக் கொண்டே வருவார்கள். கட்டடம் முடிந்து கிருகப் பிரவேசத் தன்று எந்தக் கட்டடம் கட்டுவதற்கு முக்கிய காரணமாக இருந்ததோ அந்தச் சவுக்கு மரத்தை யார் கண்ணிலும் படாமல் பின்னால் எங்கயோ மறைத்து வைத்துவிட்டு, வேறெங்கேயோ வளர்ந்த வாழை மரத்தை முன்னால் நட்டு கிரகப் பிரவேசம் நடத்தி அனைவரையும் வரவேற்பார்கள்.

Figure 4.7: Pre-processed image2

Camera captures the input image contain printed Tamil text and then the captured image is Pre-processed by using open cv.

Boundary Detection: **Translation:**

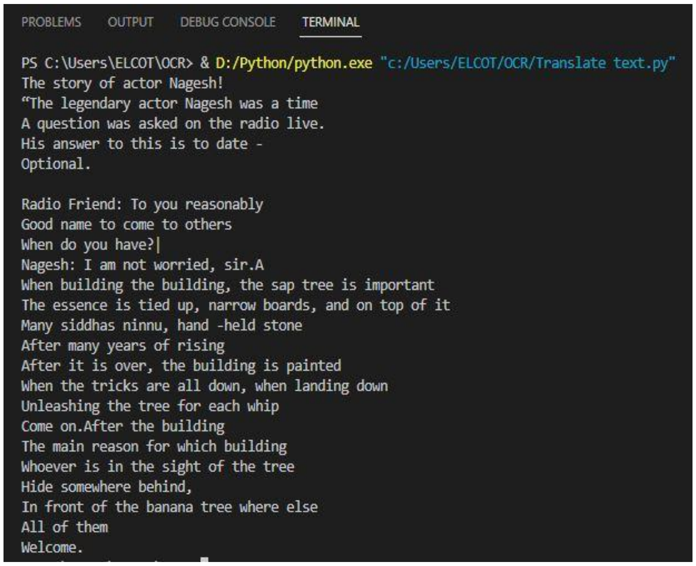


நடிகர் நாகேஷ் சொன்ன கதை!

"பழம்பெரும் நடிகர் நாகேஷிடம் அந்தக் காலத்தில் ஒரு வானொலி நேரலையில் ஒரு கேள்வி கேட்கப்பட்டது. அதற்கு அவர் தந்த பதில் இன்று வரையில் பெருமைக்கு உரியது.

வானொலி நண்பர் : நியாயமாக உங்களுக்கு வரவேண்டிய நல்ல பெயர் மற்றவர்களுக்குச் செல்லும் போது உங்களுக்கு எப்படி இருக்கும்?

நாகேஷ்: நான் கவலையே படமாட்டேன் சார். ஒரு கட்டடம் கட்டும் போது, சவுக்கு மரத்தை முக்கியமா வச்சு சாரம் கட்டி, குறுக்குப் பலகைகள் போட்டு, அதன் மேல் பல சித்தாள்கள் நின்னு, கைக்குக் கை கல் மாறி கட்டடம் உயர்ந்து கொண்டே போய் பல ஆண்டுகளுக்குப் பிறகு அது முடிந்த பிறகு, அந்தக் கட்டிடத்துக்கு வர்ண ஜால வித்தைகள் எல்லாம் அடிச்சு, கீழ இறங்கும் போது ஒவ்வொரு சவுக்கு மரமாக அவிழ்த்துக் கொண்டே வருவார்கள். கட்டடம் முடிந்து கிருகப் பிரவேசத் தன்று எந்தக் கட்டடம் கட்டுவதற்கு முக்கிய காரணமாக இருந்ததோ அந்தச் சவுக்கு மரத்தை யார் கண்ணிலும் படாமல் பின்னால் எங்கேயோ மறைத்து வைத்துவிட்டு, வேறெங்கேயோ வளர்ந்த வாழை மரத்தை முன்னால் நட்டு கிரகப் பிரவேசம் நடத்தி அனைவரையும் வரவேற்பார்கள்.



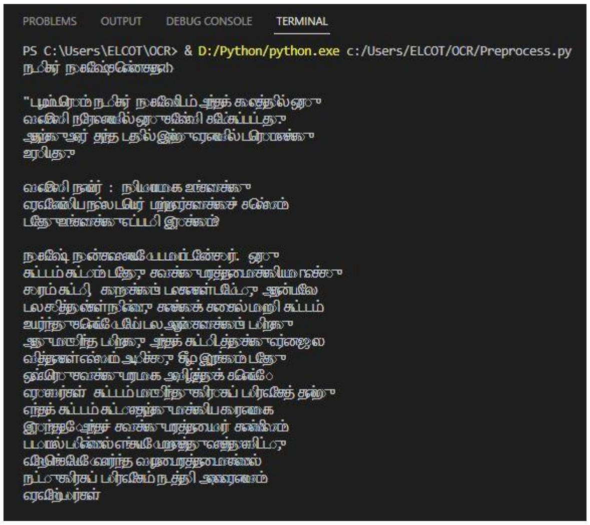
```
PS C:\Users\ELCOT\OCR> & D:/Python/python.exe "c:/Users/ELCOT/OCR/Translate text.py"
The story of actor Nagesh!
“The legendary actor Nagesh was a time
A question was asked on the radio live.
His answer to this is to date -
Optional.

Radio Friend: To you reasonably
Good name to come to others
When do you have?|
Nagesh: I am not worried, sir.A
When building the building, the sap tree is important
The essence is tied up, narrow boards, and on top of it
Many siddhas ninnu, hand -held stone
After many years of rising
After it is over, the building is painted
When the tricks are all down, when landing down
Unleashing the tree for each whip
Come on.After the building
The main reason for which building
Whoever is in the sight of the tree
Hide somewhere behind,
In front of the banana tree where else
All of them
Welcome.
```

Figure 4.8: Boundary Detected Figure 4.9: Translated Output for image2

for image2

Pre-processed image sends to vision OCR for Boundary detection and extracted text translate into English text.

Output Text:



```
PS C:\Users\ELCOT\OCR> & D:/Python/python.exe c:/Users/ELCOT/OCR/Preprocess.py
[illegible]

[illegible]
[illegible]
[illegible]
[illegible]

[illegible]
[illegible]
[illegible]

[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
```

Speech generated using TTS

Fig.4.10: Output Text for image2

The Output text contain extracted text from the input image2 using vision OCR.

Test Case 3: Input image contain handwritten English text

Input Image: **Pre-processing Image:**

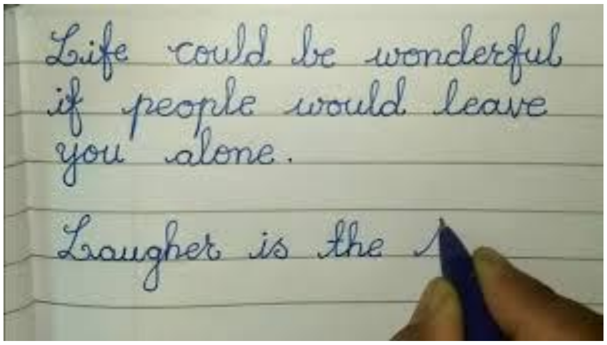


Fig.4.11: Input image3

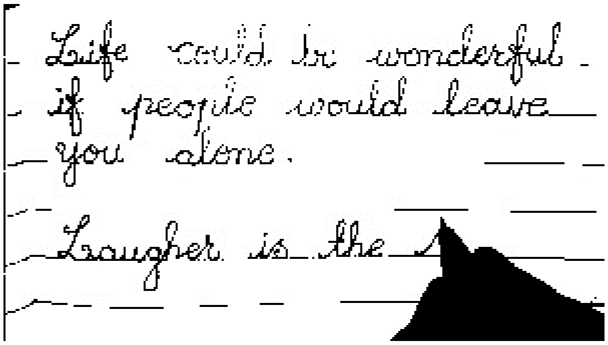


Fig.4.12: Pre-processed image3

Camera captures the input image contain handwritten English text and then the captured image is Pre-processed by using open cv.

Boundary Detection: **Translation:**

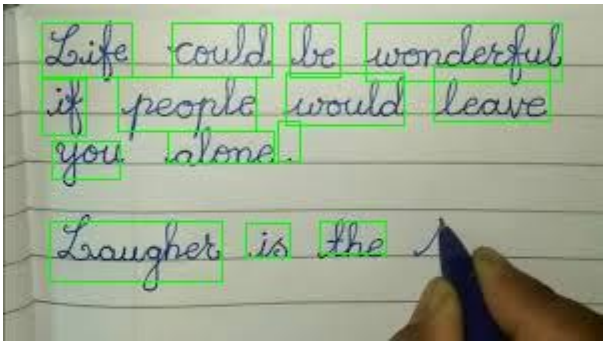
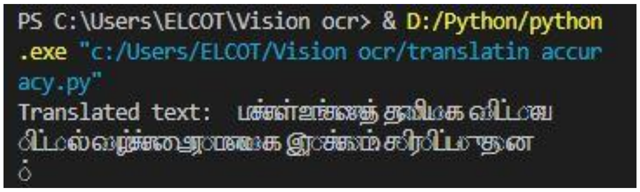


Figure 4.13: Boundary Detected for image3

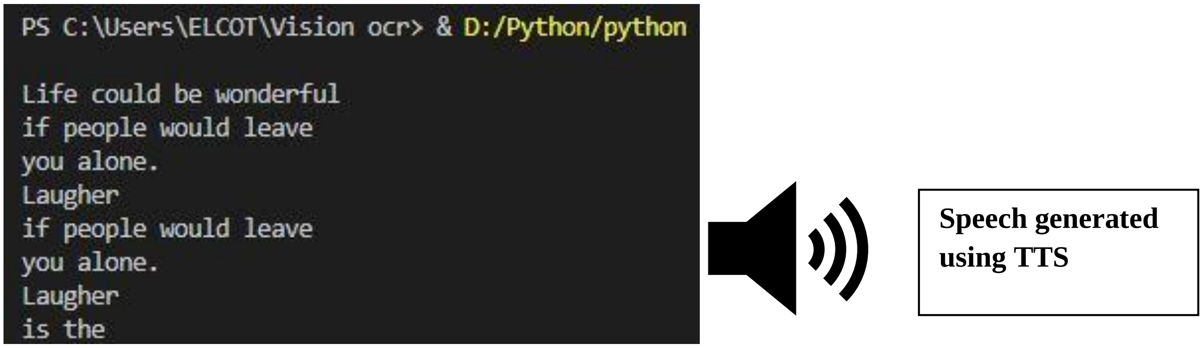


```
PS C:\Users\ELCOT\Vision ocr> & D:/Python/python
.exe "c:/Users/ELCOT/Vision ocr/translatin accur
acy.py"
Translated text:  மக்கள்உங்களைத் தனிமக விட்டவ
ிட்டல் வாழ்க்கைஅற்புதமாக இருக்கும் சிரிப்பு தான
்
```

Figure 4.14: Translated Output for image3

Pre-processed image sends to vision OCR for Boundary detection and extracted text translate into Tamil text.

Output Text:



```
PS C:\Users\ELCOT\Vision ocr> & D:/Python/python

Life could be wonderful
if people would leave
you alone.
Laugher
if people would leave
you alone.
Laugher
is the
```

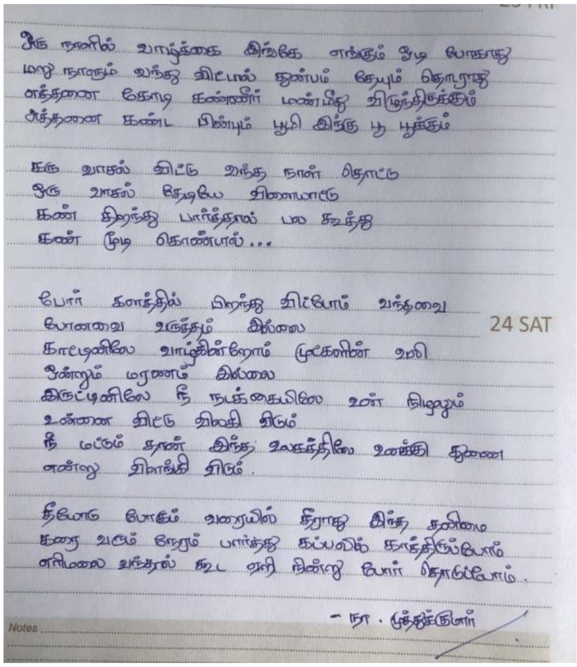
Speech generated using TTS

Figure 4.15: Output Text for image3

The Output text contain extracted text from the input image3 using vision OCR.

Test Case 4: Input image contain handwritten Tamil text

Input Image: **Pre-processing Image:**



ஒரு நாளில் வாழ்க்கை இங்கே எங்கும் ஓடி போகாது
மறு நாளும் வந்து விட்டால் துன்பம் தேயும் தொடராது
எத்தனை கோடி கண்ணீர் மண்மீது விழுந்திருக்கும்
அத்தனை கண்ட பின்பும் பூமி இங்கு பூ பூக்கும்

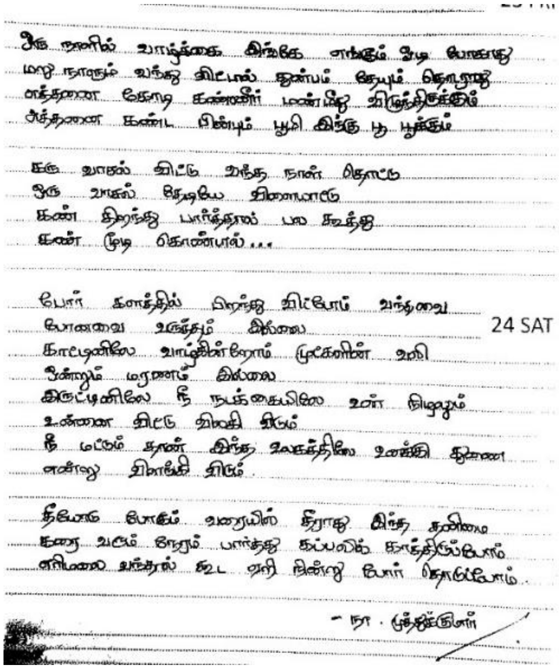
கரு வாசல் விட்டு வந்த நாள் தொட்டு
ஒரு வாசல் தேடியே விளையாட்டு
கண் திறந்து பார்த்தால் பல கூத்து
கண் மூடி கொண்டால்...

போர் களத்தில் பிறந்து விட்டோம் வந்தவை
போனவை வருத்தம் இல்லை
காட்டினிலே வாழ்கின்றோம் முட்களின் வலி
ஒன்றும் மரணம் இல்லை
இருட்டினிலே நீ நடக்கையிலே உன் நிழலும்
உன்னை விட்டு விலகி விடும்
நீ மட்டும் தான் இந்த உலகத்திலே உனக்கு துணை
என்று விளங்கி விடும்.

தீயோடு போகும் வரையில் தீராது இந்த தனிமை
கரை வரும் நேரம் பார்த்து கப்பலில் காத்திருப்போம்
எரிமலை வந்தால் கூட ஏறி நின்று போர் தொடுப்போம்.

- நா. முத்துக்குமார்

24 SAT
Notes



ஒரு நாளில் வாழ்க்கை இங்கே எங்கும் ஓடி போகாது
மறு நாளும் வந்து விட்டால் துன்பம் தேயும் தொடராது
எத்தனை கோடி கண்ணீர் மண்மீது விழுந்திருக்கும்
அத்தனை கண்ட பின்பும் பூமி இங்கு பூ பூக்கும்

கரு வாசல் விட்டு வந்த நாள் தொட்டு
ஒரு வாசல் தேடியே விளையாட்டு
கண் திறந்து பார்த்தால் பல கூத்து
கண் மூடி கொண்டால்...

போர் களத்தில் பிறந்து விட்டோம் வந்தவை
போனவை வருத்தம் இல்லை
காட்டினிலே வாழ்கின்றோம் முட்களின் வலி
ஒன்றும் மரணம் இல்லை
இருட்டினிலே நீ நடக்கையிலே உன் நிழலும்
உன்னை விட்டு விலகி விடும்
நீ மட்டும் தான் இந்த உலகத்திலே உனக்கு துணை
என்று விளங்கி விடும்.

தீயோடு போகும் வரையில் தீராது இந்த தனிமை
கரை வரும் நேரம் பார்த்து கப்பலில் காத்திருப்போம்
எரிமலை வந்தால் கூட ஏறி நின்று போர் தொடுப்போம்.

- நா. முத்துக்குமார்

Figure 4.16: Input image4 **Figure 4.17: Pre-processed image4**

Camera captures the input image contain handwritten English text and then the captured image is Pre-processed by using open cv.

Boundary Detection:

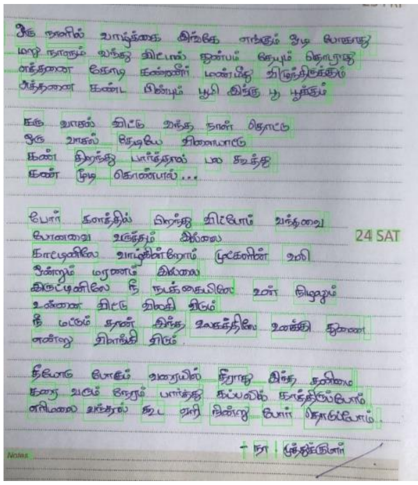
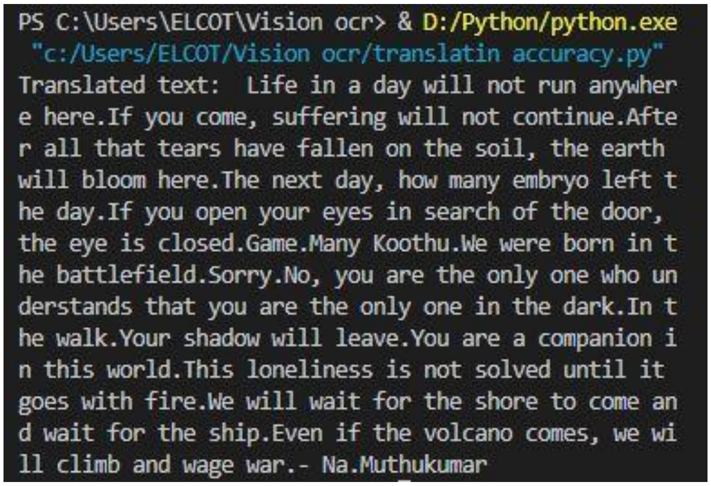

24 SAT
Notes

Figure 4.18: Boundary Detected for image4

Translation:

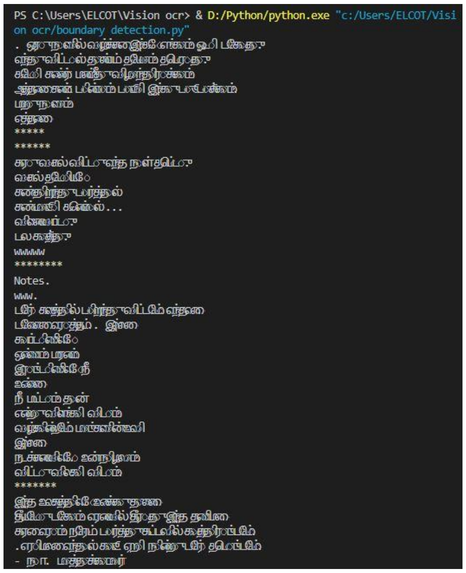


```
PS C:\Users\ELCOT\Vision ocr> & D:/Python/python.exe
 "c:/Users/ELCOT/Vision ocr/translatin accuracy.py"
Translated text:  Life in a day will not run anywher
e here.If you come, suffering will not continue.Afte
r all that tears have fallen on the soil, the earth
will bloom here.The next day, how many embryo left t
he day.If you open your eyes in search of the door,
the eye is closed.Game.Many Koothu.We were born in t
he battlefield.Sorry.No, you are the only one who un
derstands that you are the only one in the dark.In t
he walk.Your shadow will leave.You are a companion i
n this world.This loneliness is not solved until it
goes with fire.We will wait for the shore to come an
d wait for the ship.Even if the volcano comes, we wi
ll climb and wage war.- Na.Muthukumar
```

Figure 4.19: Translated Output for image4

Pre-processed image sends to vision OCR for Boundary detection and extracted text translate into English text.

Output Text:



```
PS C:\Users\ELCOT\Vision ocr> & D:/Python/python.exe "c:/Users/ELCOT/Visi
on ocr/boundary detection.py"
. ஒரு நாளில் வாழ்க்கை இங்கே எங்கும் ஓடி போகாது
வந்துவிட்டால் துன்பம் தீண்டும் தொடராது
எமீி கண்ணீர் மண்மீதுவிழுந்திருக்கும்
அத்தனைகண் பின்பும் பூமி இங்கு பூபூக்கும்
மறுநாளும்
எத்தனை
*****
******
கருவகல் விட்டுவந்த நாள்தொட்டு
வசல்தேடியே
கண்திறந்து பார்த்தல்
கண்மூடி கொண்டல்...
விளையாட்டு
பலகூத்து
wwwww
********
Notes.
www.
போர் களத்தில் பிறந்துவிட்டோம் வந்தவை
போனவையுத்தம் .  இல்லை
காட்டினிலே
ஒன்றும் புரண்
இருட்டினிலே நீ
உன்னை
நீ மட்டும் தான்
என்றுவிலகி விடும்
வழிகின்றும் மட்டுமின்றி
இவ்வை
நடக்கையிலே உன்நிழலும்
விட்டுவிலகி விடும்
*******
இந்த உலகத்திலே உன்னோடு தனிமை
தீயோடு போகும் வரையில் தீராது இந்த தனிமை
கரைவரும் நேரம் பார்த்து கப்பலில் காத்திருப்போம்
. எரிமலை வந்தல் கூட ஏறி நின்று போர் தொடுப்போம்
- நா. முத்துக்குமார்
```

Speech generated using TTS

Figure 4.20: Output Text for image4

The Output text contain extracted text from the input image4 using vision OCR.

Table 4.1: Accuracy for Google Cloud Vision API

TEST CASES	CONTENT	ACCURACY
1	Input image contain printed English text	99%
2	Input image contain printed Tamil text	98%
3	Input image contain handwritten English text	91%
4	Input image contain handwritten Tamil text	90%

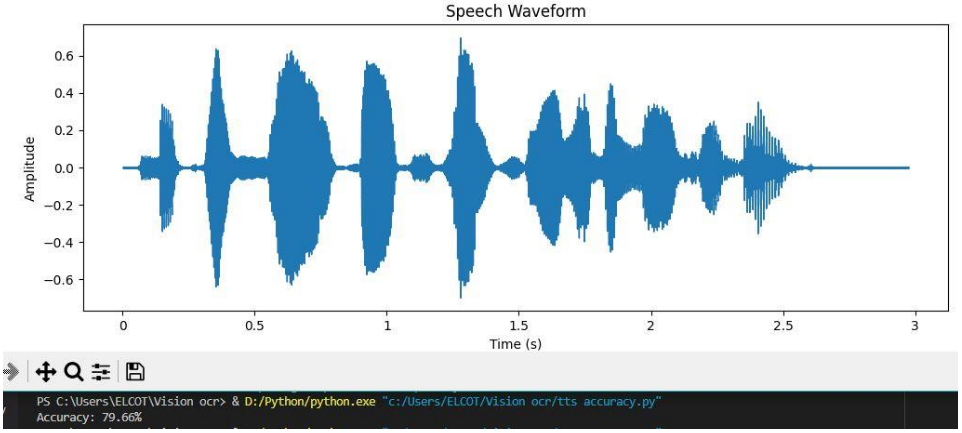
Average: 94.5%

Table 4.2: Accuracy for Google Translate

OBTAINED LANGUAGE	PREFERRED LANGUAGE	ACCURACY
English	Tamil	94%
Tamil	English	93%

Average: 93.5%

The accuracy of Vision OCR & Google Translate is analyzed by comparing the extracted text with expected text & calculating the accuracy in percentage.



```
PS C:\Users\ELCOT\Vision ocr> & D:/Python/python.exe "c:/Users/ELCOT/Vision ocr/tts accuracy.py"
Accuracy: 79.66%
```

Figure 4.21: Accuracy for Text to Speech Output

The accuracy of Google Cloud TTS is analyzed by comparing the generated speech output with expected speech output & calculating the accuracy in percentage.

CHAPTER 5

CONCLUSION

5.1 CONCLUSION

In this project, a model has been built with Raspberry Pi 4 – based Text to Speech device using Google Cloud Vision API, Translator, Google Cloud Text to Speech. The Device successfully recognizes text of various fonts from different input images and transforms the OCR output into an audio output, which is extremely accurate. We have implemented Google Cloud Vision Engine because, it is the most powerful and open-source software, and also it requires license/investment. It will assist visually impaired people so that they can read the document using this aiding methodology. Experiments have been performed by evaluating visual comparison of OCR-Test Cases and good results have been achieved by using the built model. The model has been a far better methodology in OCRing any input document or image and transcribing it into audio output. The proposed system eases the digital experience of people having learning disabilities, reduced vision or those with literary difficulties.

Our system has overcome the existing drawback of converting the text image with multi-lingual script & converting the speech to their preferred language. Although our system has some drawback that Vision OCR does not recognize all symbols and punctuation marks and more spacing between words can affect the OCR accuracy. Because of this, we have faced delay in speech output generating & the naturalness of speech output will be affected.

The overall accuracy of the Google Cloud Vision API that we obtained for printed text is 98.5%, for handwritten text is 90.5%. For Google Translate we obtained English to Tamil accuracy of 94% for Tamil to English is 93% whereas in Google Cloud Text-to-Speech API we obtained an accuracy of 79.66%.

5.2 FUTURE SCOPE

Motion sensor-based Text Extraction: A pen equipped with motion sensor to predict the letters written by the user. Instead of using image data or on-screen stroke data, analyze the acceleration and gyroscopic data of the pen using ML techniques to classify the characters while the user is writing.

Speech Customization: Providing users with the ability to customize speech output according to their preferences could be a valuable addition. This could include options for adjusting voice pitch, speed, and volume, as well as incorporating user-specific pronunciation rules or preferences.

Integration with Natural Language Processing (NLP): Integrating NLP techniques could enhance the project's capabilities by allowing for more context-aware translation and speech generation. This would enable the system to better understand and interpret the meaning behind the text, resulting in more accurate translations and more natural-sounding speech output.

Voice Recognition and Interaction: Integrating voice recognition capabilities would allow users to interact with the system using voice commands, making the overall user experience more intuitive and hands-free. This could involve features such as voice-activated image capture or voice-controlled language selection.

Mobile Application Development: Creating a dedicated mobile application for this project would make it more accessible and convenient for users on smartphones and tablets. The application could provide a user-friendly interface, additional features, and offline capabilities for situations where internet connectivity may be limited.