# AN ARTIFICIAL INTELLIGENCE-BASED ALGORITHM FOR THE VISUALLY IMPAIRED TO LISTEN TO THECONTENTS OF TEXT IN AN IMAGE

E.A.G.A. Edirisinghe
Department of Electrical and Electronic Engineering
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
anu.edirisinghe77@gmail.com

J.A.S.Y. Jayasinghe
Department of Electrical and Electronic Engineering
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
shehanijay@gmail.com

***Abstract*—This paper presents an algorithm that can be used in a smart Computer vision device for the visually impaired. Complete or virtually full loss of eyesight is referred to as blindness. Words in an image object can be difficult for the blind to decipher. This program allows the blind to read written and handwritten material with the same ease as sighted people. For the benefit of the visually impaired, this system's primary purpose is to convert the written text included in an image into spoken language, and in order to create the algorithm, we use Artificial Intelligence. Those who are blind or have low vision will find this approach especially helpful to listen to necessary content.**

***Keywords-Artificial Intelligence; Visual Impairment; Algorithm; Computer Vision; Text to Speech***

## I. INTRODUCTION

About 270 million individuals, or 3.88 percent of the global population, are blind. Until the 2000s, the visually challenged relied on Braille to read printed material. The major problem with Braille is that so few people can read it and so few books are converted to Braille. Individuals who are blind may now read printed and handwritten text just as easily as sighted individuals. Full or virtually full loss of eyesight is referred to as blindness. Those who are visually impaired but not completely blind may usually read standard-sized print or use magnification aids to read larger fonts. Many helpful devices for Braille readers and writers have been developed by modern technology [1, 2, 3].

Visually impaired persons have a significant challenge when attempting to extract text from color pictures. One of the most common uses of mobile phones nowadays is sending and receiving text messages, however, those who are blind or have low vision cannot do this [4].

Some current Braille book-making systems now have audio playback capabilities, making this format accessible to those who are blind or visually impaired. Because of the difficulties associated with Braille, it is clear that a new Text-to-Speech (TTS) conversion approach is required [5]. A text-to-speech system uses voice synthesis to turn written text into spoken words. The plan is to set up a method through which a visually impaired individual may learn new vocabulary. Many individuals lack functional or even minimal eyesight. They may learn more about the text with the aid of this strategy of turning it into speech. Those who are visually impaired may use this method to learn the meanings ofprinted words.

Text-to-speech technology enables visually impaired people to access a wide variety of content, such as books, articles, webpages, emails, and other digital documents. By converting text to speech, they may listen to the content and interpret it using aural cues, allowing for independent information access. This access to information is essential for involvement in school, employment, social activities, and everyday life. They can read and interpret materials without the need for sighted help by transforming written content to speech. This independence promotes self-esteem, autonomy, and a sense of control. Even though visually impaired people have access to Text-to-Speech technologies for typed content, handwritten text is more challenging to interpret because it is difficult to be understood by screen readers or assistive devices. Text-to-speech conversion can help overcome this barrier by converting handwritten text into machine-readable text and then converting it to speech using Optical Character Recognition (OCR) technology [11]. This feature enables visually challenged people to read handwritten notes, letters, greeting cards, and other handwritten items. Natural language processing and machine learning, which are two recent advances in text-to-speech technology, have considerably increased the quality and accuracy of speech synthesis.

Referring several additional context aspects of research publications inspired the development of the suggested system of an intelligent device that can read printed and written texts to help the visually handicapped. The primary objective of this research is to create an AI-powered computer vision algorithm for the visually impaired. The developed technique provides a user-friendly and efficient means of converting images to voice at little cost. Not only does it help the visually impaired save time and effort, but it also gives them more freedom in their daily lives.

In order to make textual information accessible to as many people as possible, this algorithm integrates several methods, such as text recognition, both typed and handwritten and text-to-speech conversion. Part of the technique depends on the robust Tesseract OCR engine, which is implemented using the Pytesseract library. This feature is ideal for recognizing and removing typed text from digital photographs of any type or font. To prepare for future processing, it analyzes the photos and extracts the textual information to offer an accurate representation of the text present. Recognizing handwritten text is more difficult than scanning printed text because of the wide variety of handwriting styles. However, the method takes a multi-stage solution to this problem. Images of handwritten text are segmented using various methods to make the subsequent recognition process more manageable. Then, the algorithm is trained using neural networking, in particular Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), on a large dataset of handwritten characters. Combining RNN output with Connectionist Temporal Classification (CTC) allows for the generation of the final recognized text by the algorithm. The gTTS (Google Text-to-Speech) module is used by the algorithm after the text has been extracted from images so that it can be converted to speech. The algorithm uses cutting-edge speech synthesis technology to read the text aloud, allowing the visually impaired to get a full comprehension of the visual material.

The algorithm incorporates these separate features so that it can read and understand both typed and handwritten language, making it more accessible to people with visual impairments. It's a novel approach that helps people with vision loss live as independently and fully as possible in all aspects of society as able-bodied people do.

Section I of the paper discusses the introduction, problem statement, and a comparison between existing systems and implemented systems while section II describes the whole methodology. Moreover, section III illustrates the results and section IV discusses the obtained results. Furthermore, section V concludes the entire research.

### A. *Problem Statement*

The number of blind and visually impaired person is estimated to be in the millions. The inability to read printed and written text has a devastating effect on the daily lives of the blind [6]. There are a number of aids for the visually impaired, but understanding them is still an important skill for them to acquire. There are a few benefits to the currently available text recognition technologies.

It's difficult for the blind to use currency. They are unable to tell the difference between grocery shops and eating establishments. They can tell the difference between people, cars, and animals only based on the sounds they make.

There are colleges and institutions that cater to students with disabilities. There is a wide range of demands, and not all of them need specialized facilities or educational programs. People with visual impairments, for instance, may learn alongside their non-disabled peers given the right opportunities and accommodations. Because schools catering to students with disabilities are either not readily available, private, or prohibitively costly, the vast majority of blind and visually impaired persons did not complete formal education.

### B. *Existing Systems*

Researches have conducted various studies regarding visually impaired people to listen/read to the contents of text in an image. They have used many technologies for this task. However, there is a lack of studies on converting both typed and handwritten texts into spoken language. The primary goal of this literature review is to compare and contrast the accuracy of various approaches.

To improve the accuracy of the text and character detection and identification modules, learning growth took use of the established framework for training with neural networks and the new unsupervised feature. Using very elementary techniques, Wang and the authors [13] have integrated these two models to create a comprehensive text recognition system. Both the ICDAR 2003 and SVT datasets have been utilized. For the first dataset, the 62-way character classifier approach yielded an accuracy of 83.9% for a cropped character [13].

Koo and Kim have developed a system for new scene text identification relied heavily on machine learning techniques. To improve precision, two distinct classes of classifiers were built; the first is used to produce candidates, while the second is used to exclude non-textual candidates. In order to make use of multi-channel data, an innovative approach has been created. In this investigation, they utilized data from two different ICDAR collections: 2005 and 2011. Several assessment methodologies have shown that this approach is very effective [14].

Different distributed language modeling was also employed in photoOCR, Bissacco et al. [15] implemented a system to recognize and extract text from images using machine learning methods. The system was designed to be able to read text from even the most difficult of pictures, such as those with low resolution or blurriness. Google Translate is only one example of an application that has made advantage of this mechanism. This system has relied on the ICDAR and SVT databases. According to the data, it takes around 600ms to analyze a single picture for text detection and identification [15].

With an ER (External Regions) detector covering around 94.8% of the characters, and a processing time of an image with 800600 resolution being 0.3s on a standard personal computer, a real-time text recognition and localization system has been developed by Neumann and Matas (2012). The system uses the ICDAR 2011 and SVT datasets. On a regular PC, the approach took an average of 0.3 seconds to execute while processing an 800600 picture. The approach obtained a recall rate of 64.7% on the ICDAR 2011 dataset. SVT image recognition was 32.9% [16].

Oriented Stroke recognition is a technique that uses a sliding window to combine two other approaches for text recognition and localization in the paper was written by Neumann and Matas (2013). A portion in the picture with certain strokes in a specific direction and location has been identified as the character or letter. The experiments conducted using the ICDAR 2011 dataset revealed an improvement in recall of 66% above baseline techniques [17].

For those without sight, there is a dearth of accessible technology. Some of these systems even aid in reading the text, in addition to aiding with navigating and identifying objects. They take the written word and make it audible. These systems have the following drawbacks.

- It has trouble distinguishing between letters of varying colors.
- Different font styles might prevent the content from being read.

### *C. Implemented System*

The system can be used for both typed and handwritten texts using two different algorithms.

The main objectives of the implemented system are as follows.

- Ability to read both typed and handwritten text just as easily.
- Ability to identify text written in a variety of styles and formats such as Arial, Lucida and Italic.
- Ability to recognize and extract texts of different colors.

## II. METHODOLOGY

The following Fig.1 shows the architecture diagram of the implemented system, which has two main processes. The implemented system takes an image as input, the text within it is recognized using an AI algorithm and stored in a text file. The speech processing algorithm takes this text file as input and produces spoken words.

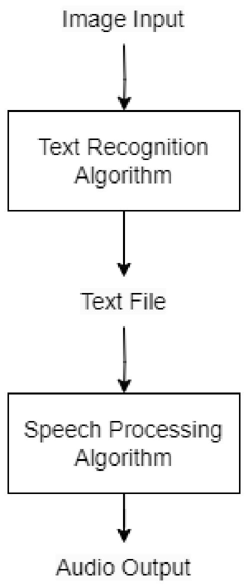
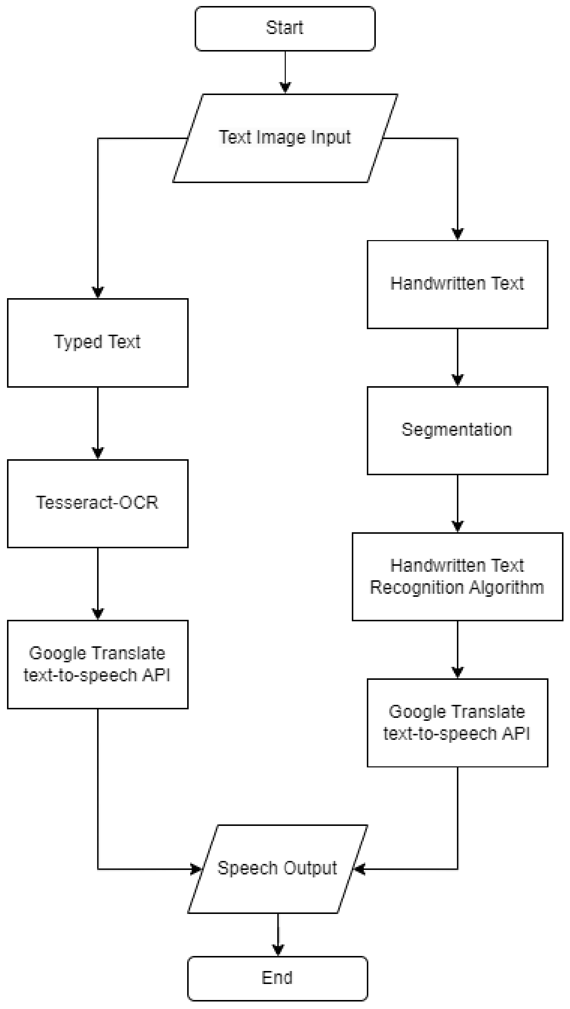


Figure 1. System Architecture

The system is designed under two main algorithms. One is for typed texts and the other one is to recognize handwritten texts.

The first algorithm makes use of Pytesseract, a wrapper for the Tesseract OCR engine, to extract typed text from digital images. These extracted data are then converted to speech using the gTTS library. Even though Pytesseract enables considerable accuracy for typed text, handwritten text cannot be accurately or reliably extracted by it. Hence the second algorithm is based on the research conducted by Harald Scheidl [7] for handwritten text recognition in historical documents. This algorithm allows the recognition of words or word lines. Prior to the implementation of this algorithm, a segmentation process occurs to divide text paragraphs to lines. After recognizing these text lines they are once again fed to the gTTS algorithm to obtain their audible format. An overview of this system is given in Fig. 2.

Figure 2. System Overview


Start
Text Image Input
Typed Text
Tesseract-OCR
Google Translate text-to-speech API
Handwritten Text
Segmentation
Handwritten Text Recognition Algorithm
Google Translate text-to-speech API
Speech Output
End

### *A. Typed Text Recognition*

#### *1) Pytesseract*

Pytesseract is a Python library that provides a straightforward method for using the Tesseract OCR engine [12]. Text recognition may be performed on photos in a variety of formats, including PNG, JPEG, TIFF, and PDF, using Pytesseract. It makes use of Tesseract's ability to analyze the image and find textual material within it. The library can handle both typed and handwritten text, though the accuracy will vary depending on the image quality and text clarity.

One of the few components of Tesseract that have been released is the line detection method [8] and it also can read

Pages with curved baselines [9]. The skewed page may be detected by the line-finding method without the need to de-skew, which prevents unnecessary data loss.

According to [10], Tesseract's structure is a sequential pipeline. Input is expected to be a binary image with polygonal text sections defined, if desired. To begin, we use linked component analysis to determine the general shapes of the constituent parts. Because of this, Tesseract can readily process white-on-black text. The contours are combined into blobs before being arranged into lines of text. The text's fixed pitch or proportionality is determined by studying its lines and regions. Character cells are used to break up fixed-pitch text into its component parts. Words in proportional text are separated by both hard and soft spaces. Tesseract uses a two-pass method for word recognition. Words are initially tried to be identified, and those that pass muster are used to train an adaptive classifier. In a subsequent pass over the page, the adaptive classifier can make corrections to the recognition accuracy.

Tesseract uses a number of methods for locating lines and words. The line finding method sorts blobs into text lines according to their height. Baselines are estimated using a least median of squares fit, and missing character associationsare restored. Quadratic splines can be used to accommodate curved baselines. Word recognition requires the separation and labeling of individual characters. Using the concave vertices of the outline as a guide, Tesseract looks for places to make cuts in linked characters. An association uses an A* search to locate the optimal segmentation of maximally chopped blobs into candidate characters if the word is still not recognized well enough.

The static character classifier in Tesseract compares the retrieved features from the unknown characters to the prototype characteristics in the training data. A class pruner and similarity computation between characteristics and prototypes are used during categorization. Tesseract's minimum linguistic analysis is accomplished by having the permute module choose the most appropriate word string from a number of different classes. The term with the smallest total distance rating is selected by multiplying it by a different constant for each category.

In addition to its static classifier, Tesseract makes use of an adaptive classifier that makes use of the same features and classifier but with alternative normalization methods. To enhance discriminating within individual documents, the adaptive classifier is trained using the results of the static classifier.

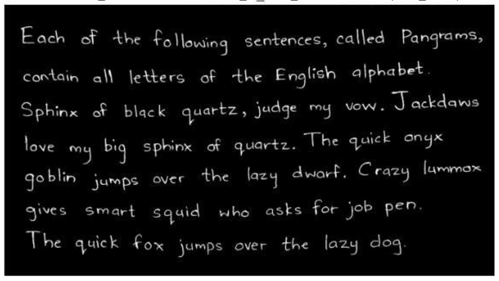
*2) gTTS*

gTTS (Google Text-to-Speech) is a Python library and command line interface (CLI) tool for communicating with Google Translate's text-to-speech API. The text fed into the gTTS library can contain various languages as well as punctuation, special characteristics and formatting. Once the it is provided with the input text that should be converted to speech, the text is processed to handle any language specific requirements and special characters. Then gTTS sends a request to Google translation Text-to-speech API serviceover the internet. Once Google TTS API service receives the request, speech synthesis is applied on the texts and an audio file is generated. The audio file received by the gTTS library is then locally stored as an mp3 file and played in real time.

### *B. Handwritten Text Recognition*

#### *1) Segmentation*

Segmentation process starts with converting the image to 2D grayscale image. By using a Sobel filter on the binarized image (Fig. 3) to detect edges, the horizontal projection profile is drawn to find the line height of the image as given in Fig. 4. Using a threshold line (Fig. 5), the peaks aredetected which correspond to areas with text (Fig. 6) and theregions to draw the segmentation lines are recognized. By using the A* search algorithm to find the shortest between two points, the segmentation lines are drawn (Fig. 7) and theimage is divided into smaller images containing single lines (Fig. 8).



Each of the following sentences, called Pangrams,
contain all letters of the English alphabet.
Sphinx of black quartz, judge my vow. Jackdaws
love my big sphinx of quartz. The quick onyx
goblin jumps over the lazy dwarf. Crazy lummox
gives smart squid who asks for job pen.
The quick fox jumps over the lazy dog.

Figure 3. Binarized Image

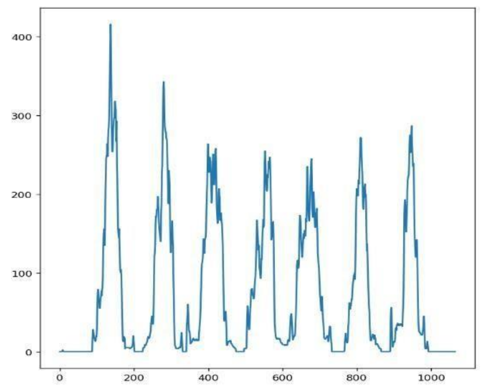


Figure 4. Horizontal Projection Profile

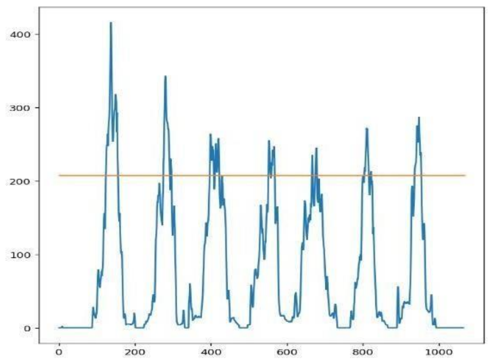


Figure 5. Threshold Line

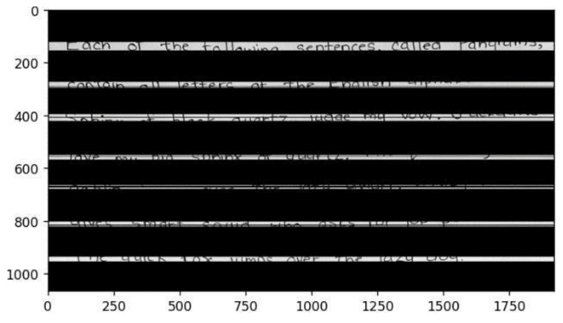


Figure 6. Areas with recognized text

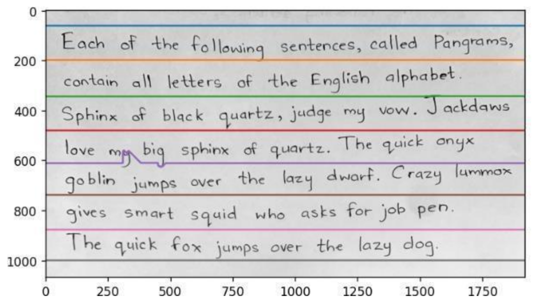


Figure 7. Recognized Segmentation Lines

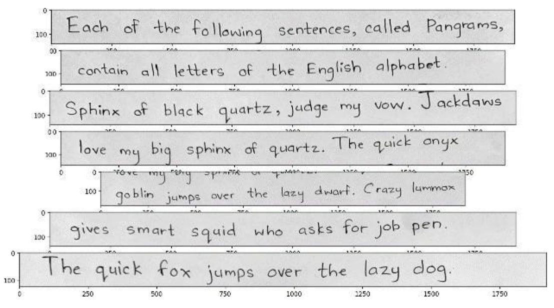


Figure 8. Segmented Lines

*2) Handwritten Text Recognition (HTR)*

This algorithm as implemented by [7] utilizes Neural Networking and consists of 5 Convolutional Neural Networks (CNNs) and 2 Recurrent Neural Networks (RNNs) to build a character probability matrix. For RNN the Long Short-Term memory implementation is used. The IAM dataset consisting of 79 characters and the Connectionist Temporal Classification (CTC) blank label are used to train the Neural Network. The RNN output is processed using CTC to obtain the final text.

The CNN layers have been trained to extract useful information from the input image. Convolution operations with filter kernels of increasing sizes (5x5 in the first two layers, 3x3 in the last three layers) are applied at each successive layer in order to record significant visual patterns. After applying the non-linear RELU function, the pooling layers are used to reduce the image size without losing any of the useful feature maps. Each entry in the 32-element sequence produced by the CNN layers has 256 features.

The feature sequence acquired from the CNN layers is fed into RNN layers, which in this case uses an LSTM implementation. With RNNs, important data can be shared between successive iterations. The RNN output sequence is converted into a 32 by 80 character classification matrix. The matrix contains 79 unique IAM dataset characters and asingle blank label CTC character.

The CTC procedure is used to train a neural network by calculating the loss value using the RNN output matrix and the ground truth text. The CTC operation performs the decoding of the RNN output matrix into recognized text during the inference phase. The CTC method does not require explicit segmentation and allows for flexible alignment between anticipated characters and their places in the image. Finding the most likely character at each time step, creating the optimal path, and finally deleting redundant characters and blanks yield the final recognized text.

*3) gTTS*

The text strings thus obtained from individual images are joined together to build the full text which is then converted to speech using the gTTS library.

## III. Results

The dataset used for this research was self-made and the sample texts used for this contain Pangrams or sentences that consist of all letters of the alphabet. Because they offer a standardized set of text containing all the letters of the alphabet, pangrams are useful for text recognition. In order to thoroughly test the accuracy and capability of the text recognition system, pangrams can be used to guarantee that every letter is used at least once.

### *A. Typed Text Results*

Each of the Figures of Fig. 9 - 15 contains original images and their extracted text. The text with the white background corresponds to the original text image while the text with black background corresponds to the recognized text saved in the text file.

First, the system was tested for typed texts with different font styles as shown in Fig. 9 - 11.

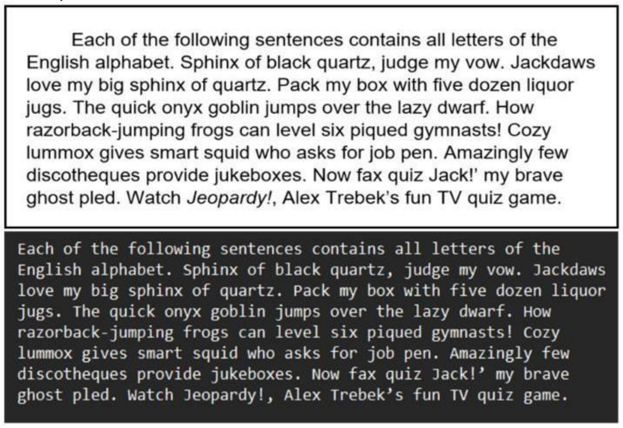
Fig. 12 illustrates the tested typed image which used italic font style while in Fig. 13-15, the system was tested using different colored typed texts and font style “Times New Roman”.

### *B. Handwritten Text Results*

The handwritten text image shown in Fig. 16 is used to test the handwritten text recognition algorithm. Fig. 17 shows the segmented lines of the handwritten text image.

The text recognized from the HTR algorithm is shown in Fig. 18.

1) Arial

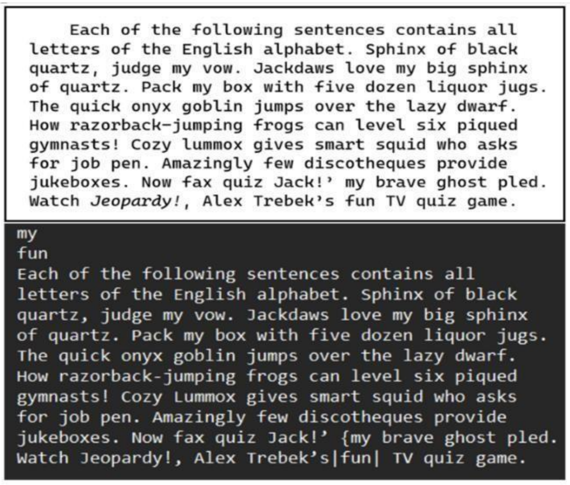


Each of the following sentences contains all letters of the English alphabet. Sphinx of black quartz, judge my vow. Jackdaws love my big sphinx of quartz. Pack my box with five dozen liquor jugs. The quick onyx goblin jumps over the lazy dwarf. How razorback-jumping frogs can level six piqued gymnasts! Cozy lummox gives smart squid who asks for job pen. Amazingly few discotheques provide jukeboxes. Now fax quiz Jack!' my brave ghost pled. Watch *Jeopardy!*, Alex Trebek's fun TV quiz game.

```
Each of the following sentences contains all letters of the
English alphabet. Sphinx of black quartz, judge my vow. Jackdaws
love my big sphinx of quartz. Pack my box with five dozen liquor
jugs. The quick onyx goblin jumps over the lazy dwarf. How
razorback-jumping frogs can level six piqued gymnasts! Cozy
lummox gives smart squid who asks for job pen. Amazingly few
discotheques provide jukeboxes. Now fax quiz Jack!’ my brave
ghost pled. Watch Jeopardy!, Alex Trebek’s fun TV quiz game.
```

Figure 9. Arial font style

2) Cascadia

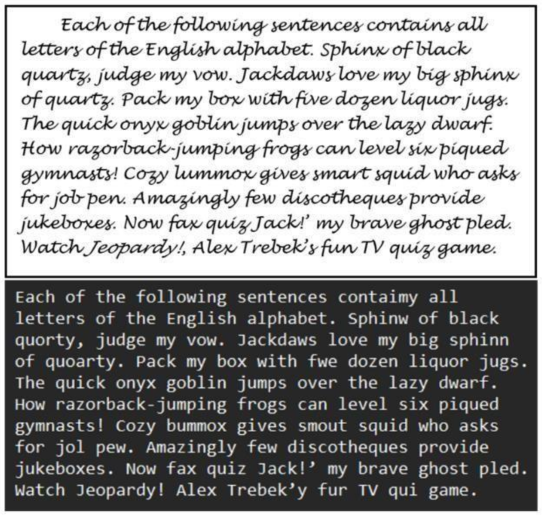


Each of the following sentences contains all letters of the English alphabet. Sphinx of black quartz, judge my vow. Jackdaws love my big sphinx of quartz. Pack my box with five dozen liquor jugs. The quick onyx goblin jumps over the lazy dwarf. How razorback–jumping frogs can level six piqued gymnasts! Cozy lummox gives smart squid who asks for job pen. Amazingly few discotheques provide jukeboxes. Now fax quiz Jack!' my brave ghost pled. Watch *Jeopardy!*, Alex Trebek's fun TV quiz game.

```
my
fun
Each of the following sentences contains all
letters of the English alphabet. Sphinx of black
quartz, judge my vow. Jackdaws love my big sphinx
of quartz. Pack my box with five dozen liquor jugs.
The quick onyx goblin jumps over the lazy dwarf.
How razorback-jumping frogs can level six piqued
gymnasts! Cozy Lummox gives smart squid who asks
for job pen. Amazingly few discotheques provide
jukeboxes. Now fax quiz Jack!’ {my brave ghost pled.
Watch Jeopardy!, Alex Trebek’s|fun| TV quiz game.
```

Figure 90. Cascadia font style

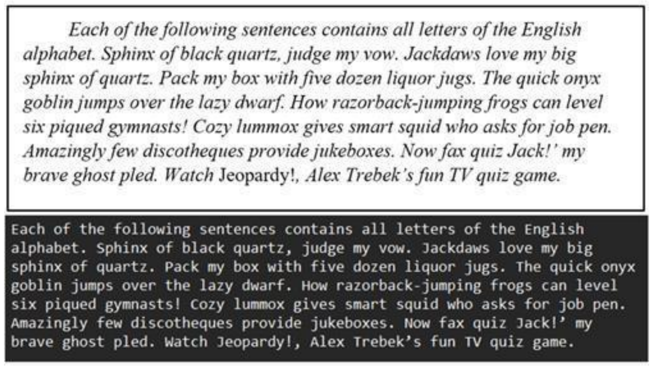
3) Lucida



Each of the following sentences contains all letters of the English alphabet. Sphinx of black quartz, judge my vow. Jackdaws love my big sphinx of quartz. Pack my box with five dozen liquor jugs. The quick onyx goblin jumps over the lazy dwarf. How razorback-jumping frogs can level six piqued gymnasts! Cozy lummox gives smart squid who asks for job pen. Amazingly few discotheques provide jukeboxes. Now fax quiz Jack!' my brave ghost pled. Watch *Jeopardy!*, Alex Trebek's fun TV quiz game.

```
Each of the following sentences contaimy all
letters of the English alphabet. Sphinw of black
quorty, judge my vow. Jackdaws love my big sphinn
of quoarty. Pack my box with fwe dozen liquor jugs.
The quick onyx goblin jumps over the lazy dwarf.
How razorback-jumping frogs can level six piqued
gymnasts! Cozy bummox gives smout squid who asks
for jol pew. Amazingly few discotheques provide
jukeboxes. Now fax quiz Jack!’ my brave ghost pled.
Watch Jeopardy! Alex Trebek’y fur TV qui game.
```

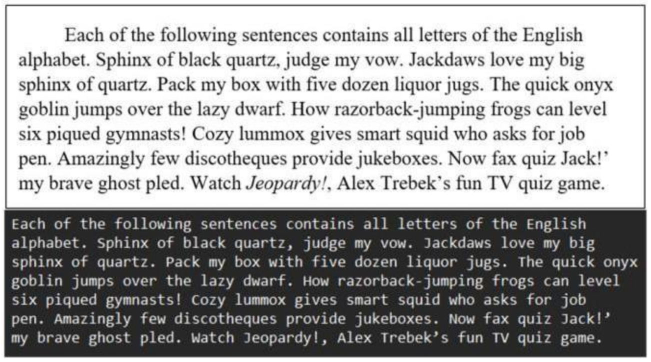
*Figure 101. Lucida font style*



*Each of the following sentences contains all letters of the English alphabet. Sphinx of black quartz, judge my vow. Jackdaws love my big sphinx of quartz. Pack my box with five dozen liquor jugs. The quick onyx goblin jumps over the lazy dwarf. How razorback-jumping frogs can level six piqued gymnasts! Cozy lummox gives smart squid who asks for job pen. Amazingly few discotheques provide jukeboxes. Now fax quiz Jack!' my brave ghost pled. Watch* Jeopardy!, *Alex Trebek's fun TV quiz game.*

```
Each of the following sentences contains all letters of the English
alphabet. Sphinx of black quartz, judge my vow. Jackdaws love my big
sphinx of quartz. Pack my box with five dozen liquor jugs. The quick onyx
goblin jumps over the lazy dwarf. How razorback-jumping frogs can level
six piqued gymnasts! Cozy lummox gives smart squid who asks for job pen.
Amazingly few discotheques provide jukeboxes. Now fax quiz Jack!’ my
brave ghost pled. Watch Jeopardy!, Alex Trebek’s fun TV quiz game.
```

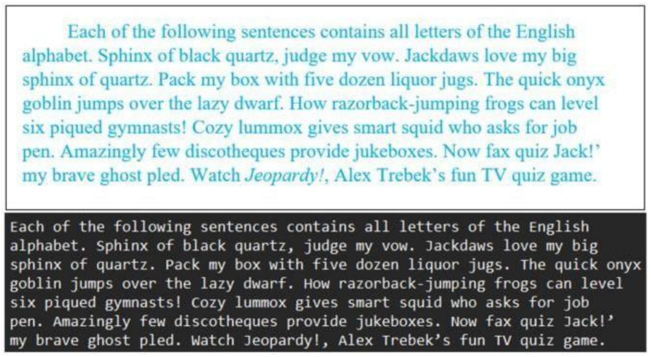
Figure 112. Italic font style



Each of the following sentences contains all letters of the English alphabet. Sphinx of black quartz, judge my vow. Jackdaws love my big sphinx of quartz. Pack my box with five dozen liquor jugs. The quick onyx goblin jumps over the lazy dwarf. How razorback-jumping frogs can level six piqued gymnasts! Cozy lummox gives smart squid who asks for job pen. Amazingly few discotheques provide jukeboxes. Now fax quiz Jack!' my brave ghost pled. Watch *Jeopardy!*, Alex Trebek's fun TV quiz game.

```
Each of the following sentences contains all letters of the English
alphabet. Sphinx of black quartz, judge my vow. Jackdaws love my big
sphinx of quartz. Pack my box with five dozen liquor jugs. The quick onyx
goblin jumps over the lazy dwarf. How razorback-jumping frogs can level
six piqued gymnasts! Cozy lummox gives smart squid who asks for job
pen. Amazingly few discotheques provide jukeboxes. Now fax quiz Jack!’
my brave ghost pled. Watch Jeopardy!, Alex Trebek’s fun TV quiz game.
```

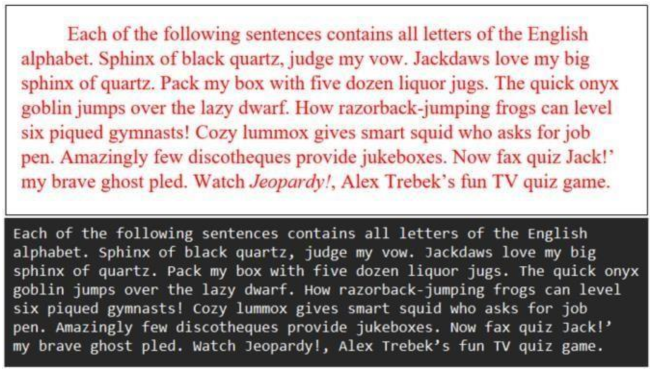
Figure 123. Black font color



Each of the following sentences contains all letters of the English alphabet. Sphinx of black quartz, judge my vow. Jackdaws love my big sphinx of quartz. Pack my box with five dozen liquor jugs. The quick onyx goblin jumps over the lazy dwarf. How razorback-jumping frogs can level six piqued gymnasts! Cozy lummox gives smart squid who asks for job pen. Amazingly few discotheques provide jukeboxes. Now fax quiz Jack!' my brave ghost pled. Watch *Jeopardy!*, Alex Trebek's fun TV quiz game.

```
Each of the following sentences contains all letters of the English
alphabet. Sphinx of black quartz, judge my vow. Jackdaws love my big
sphinx of quartz. Pack my box with five dozen liquor jugs. The quick onyx
goblin jumps over the lazy dwarf. How razorback-jumping frogs can level
six piqued gymnasts! Cozy lummox gives smart squid who asks for job
pen. Amazingly few discotheques provide jukeboxes. Now fax quiz Jack!’
my brave ghost pled. Watch Jeopardy!, Alex Trebek’s fun TV quiz game.
```

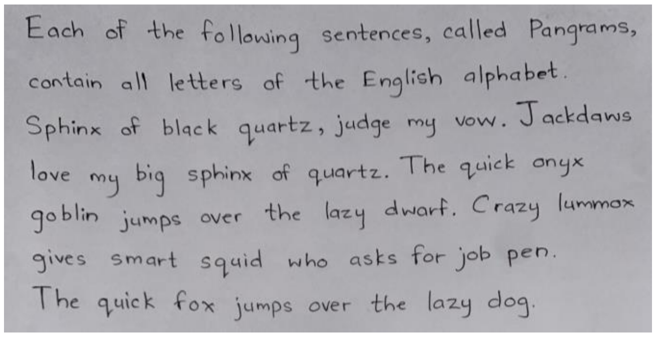
Figure 134. Blue font color



Each of the following sentences contains all letters of the English alphabet. Sphinx of black quartz, judge my vow. Jackdaws love my big sphinx of quartz. Pack my box with five dozen liquor jugs. The quick onyx goblin jumps over the lazy dwarf. How razorback-jumping frogs can level six piqued gymnasts! Cozy lummox gives smart squid who asks for job pen. Amazingly few discotheques provide jukeboxes. Now fax quiz Jack!' my brave ghost pled. Watch *Jeopardy!*, Alex Trebek's fun TV quiz game.

```
Each of the following sentences contains all letters of the English
alphabet. Sphinx of black quartz, judge my vow. Jackdaws love my big
sphinx of quartz. Pack my box with five dozen liquor jugs. The quick onyx
goblin jumps over the lazy dwarf. How razorback-jumping frogs can level
six piqued gymnasts! Cozy lummox gives smart squid who asks for job
pen. Amazingly few discotheques provide jukeboxes. Now fax quiz Jack!’
my brave ghost pled. Watch Jeopardy!, Alex Trebek’s fun TV quiz game.
```

Figure 145. Red font color



Each of the following sentences, called Pangrams,
contain all letters of the English alphabet.
Sphinx of black quartz, judge my vow. Jackdaws
love my big sphinx of quartz. The quick onyx
goblin jumps over the lazy dwarf. Crazy lummox
gives smart squid who asks for job pen.
The quick fox jumps over the lazy dog.

Figure 156. Handwritten text image

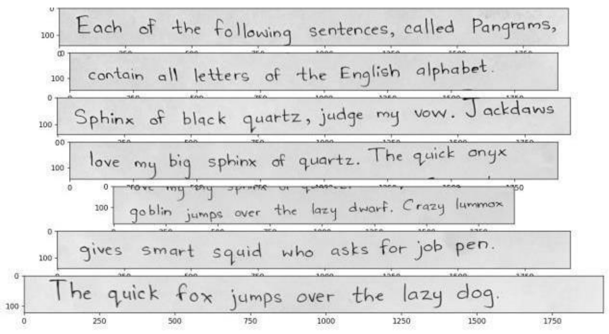
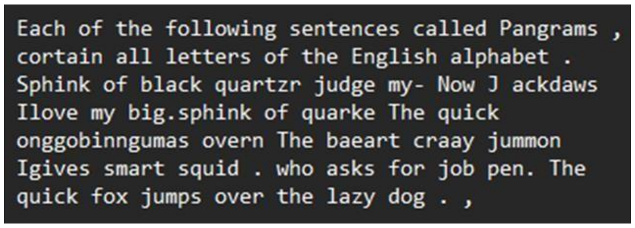


Figure 167. Segmented lines of the handwritten text image



```
Each of the following sentences called Pangrams ,
cortain all letters of the English alphabet .
Sphink of black quartzr judge my- Now J ackdaws
Ilove my big.sphink of quarke The quick
onggobinngumas overn The baeart craay jummon
Igives smart squid . who asks for job pen. The
quick fox jumps over the lazy dog . ,
```

Figure 178. The text recognized from the HTR algorithm

### C. *Text to Speech Conversion*

The Fig. 19 shows the playing of the mp3 media file stored after the text to speech conversion.

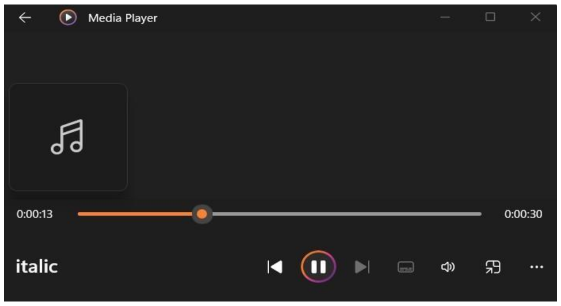


Figure 19. Speech converted media file

### D. *Validation of Results*

The evaluation metrics used are precision, recall and F1 score for validating the correct recognition of letters and error percentage for validating the correct recognition of words. The text to speech recognition (after the texts were extracted) was 100% accurate and this was validated by listening to each word.

The precision, recall and F1 score of all the letters of the alphabet, in both typed and handwritten text, were calculated according to (1), (2), and (3), where TP is the number of True Positives, FP is the number of False Positives and FN is the number of False Negatives. The results of these are given in table I.

$$Precision = \frac{TP}{TP+FP} \qquad (1)$$

$$Recall = \frac{TP}{TP+FN} \qquad (2)$$

$$F1\ Score = \frac{2 \times Precision\ \times Recall}{Precision + Recall} \qquad (3)$$

As a mean of calculating the accuracy of the typed text recognition algorithm, the error percentage was calculated for each font type/color and is shown in table II. This was done by comparing the recognized text with the original text at the word level. The percentage of incorrectly recognized words or characters were calculated to evaluate the system's performance.

The calculation of error percentage is given in (4) where ‘EP’ is the error percentage, ‘n’ is the Number of incorrect words recognized, and ‘N’ is the Total number of words in the image

$$EP = \frac{n}{N} \times 100\ \% \qquad (4)$$

According to table I, precision, recall and F1 score are nearly equal for many typed letters while, some typed letters hold a higher precision value. On the other hand, most of the handwritten letters present different precision, recall and F1 score. In table I, only a few handwritten letters show the same value for precision, recall and F1 score.

TABLE I. PRECISION, RECALL AND F1 SCORE OF LETTERS

| Letter | Typed | | | | | | Handwritten | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TP | FP | FN | Precision | Recall | F1 score | TP | FP | FN | Precision | Recall | F1 score |
| a | 26 | 0 | 2 | 1.0000 | 0.9286 | 0.9630 | 18 | 3 | 2 | 0.8571 | 0.9000 | 0.8780 |
| b | 9 | 1 | 1 | 0.9000 | 0.9000 | 0.9000 | 5 | 1 | 0 | 0.8333 | 1.0000 | 0.9091 |
| c | 13 | 0 | 0 | 1.0000 | 1.0000 | 1.0000 | 9 | 0 | 0 | 1.0000 | 1.0000 | 1.0000 |
| d | 10 | 0 | 0 | 1.0000 | 1.0000 | 1.0000 | 5 | 0 | 1 | 1.0000 | 0.8333 | 0.9091 |
| e | 35 | 0 | 0 | 1.0000 | 1.0000 | 1.0000 | 21 | 2 | 0 | 0.9130 | 1.0000 | 0.9545 |
| f | 12 | 0 | 0 | 1.0000 | 1.0000 | 1.0000 | 7 | 0 | 1 | 1.0000 | 0.8750 | 0.9333 |
| g | 13 | 0 | 0 | 1.0000 | 1.0000 | 1.0000 | 8 | 2 | 0 | 0.8000 | 1.0000 | 0.8889 |
| h | 15 | 0 | 0 | 1.0000 | 1.0000 | 1.0000 | 12 | 0 | 0 | 1.0000 | 1.0000 | 1.0000 |
| i | 20 | 0 | 1 | 1.0000 | 0.9524 | 0.9756 | 11 | 2 | 0 | 0.8462 | 1.0000 | 0.9167 |
| j | 9 | 0 | 0 | 1.0000 | 1.0000 | 1.0000 | 4 | 1 | 1 | 0.8000 | 0.8000 | 0.8000 |
| k | 9 | 0 | 0 | 1.0000 | 1.0000 | 1.0000 | 5 | 3 | 0 | 0.6250 | 1.0000 | 0.7692 |
| l | 17 | 1 | 1 | 0.9444 | 0.9444 | 0.9444 | 12 | 0 | 3 | 1.0000 | 0.8000 | 0.8889 |
| m | 12 | 1 | 0 | 0.9231 | 1.0000 | 0.9600 | 8 | 0 | 0 | 1.0000 | 1.0000 | 1.0000 |
| n | 15 | 1 | 3 | 0.9375 | 0.8333 | 0.8824 | 11 | 4 | 1 | 0.7333 | 0.9167 | 0.8148 |
| o | 29 | 3 | 0 | 0.9063 | 1.0000 | 0.9508 | 19 | 0 | 0 | 1.0000 | 1.0000 | 1.0000 |
| p | 11 | 0 | 0 | 1.0000 | 1.0000 | 1.0000 | 6 | 0 | 1 | 1.0000 | 0.8571 | 0.9231 |
| q | 9 | 0 | 0 | 1.0000 | 1.0000 | 1.0000 | 5 | 0 | 0 | 1.0000 | 1.0000 | 1.0000 |
| r | 14 | 1 | 1 | 0.9333 | 0.9333 | 0.9333 | 9 | 3 | 1 | 0.7500 | 0.9000 | 0.8182 |
| s | 22 | 0 | 2 | 1.0000 | 0.9167 | 0.9565 | 15 | 0 | 1 | 1.0000 | 0.9375 | 0.9677 |
| t | 19 | 0 | 0 | 1.0000 | 1.0000 | 1.0000 | 13 | 1 | 1 | 0.9286 | 0.9286 | 0.9286 |
| u | 16 | 1 | 0 | 0.9412 | 1.0000 | 0.9697 | 9 | 0 | 0 | 1.0000 | 1.0000 | 1.0000 |
| v | 8 | 0 | 1 | 1.0000 | 0.8889 | 0.9412 | 4 | 0 | 1 | 1.0000 | 0.8000 | 0.8889 |
| w | 10 | 3 | 0 | 0.7692 | 1.0000 | 0.8696 | 4 | 0 | 1 | 1.0000 | 0.8000 | 0.8889 |
| x | 7 | 0 | 2 | 1.0000 | 0.7778 | 0.8750 | 1 | 0 | 4 | 1.0000 | 0.2000 | 0.3333 |
| y | 10 | 4 | 0 | 0.7143 | 1.0000 | 0.8333 | 4 | 0 | 2 | 1.0000 | 0.6667 | 0.8000 |
| z | 6 | 0 | 3 | 1.0000 | 0.6667 | 0.8000 | 2 | 0 | 2 | 1.0000 | 0.5000 | 0.6667 |

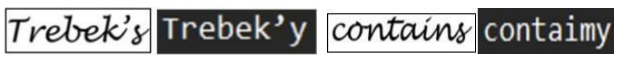
TABLE II. ERROR PERCENTAGE

| Font Type / Color | Number of | Error Percentage |
|---|---|---|
| Arial | 0/83 | 0% |
| Cascadia | 2/83 | 2.41% |
| Lucida | 13/83 | 15.66% |
| Italic | 0/83 | 0% |
| Black | 0/83 | 0% |
| Blue | 0/83 | 0% |
| Red | 0/83 | 0% |
| Handwritten | 16/53 | 30.19% |

## IV. DISCUSSION

As shown in Fig. 9 and Fig. 13 the text with simple font styles such as Arial and Times New Roman were recognized with high accuracy and no incorrect words were recognized. In Fig. 10 containing Cascadia text, the words “my” and “fun” have been extracted and saved prior to any other texts. This however has not affected the rest of the content recognized and the error rate was around 2.41%. However, some letters such as 's', 'x', and 'z' posed difficulties in recognition when using certain font styles such as Lucida. There were 15.66% errors when using the Lucida and this can be seen in Fig 11.

Fig.20 presents the words which had problems with recognizing the letter ‘s’.


Trebek's
Trebek'y
contains
contaimy
Figure 180. Letter ‘s’ typed text error

Texts that weren't able to recognize the letter 'z' using the implemented algorithm are shown in Fig. 21.

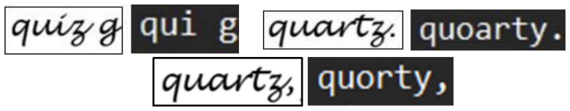


Figure 191. Letter 'z' typed text error

In Fig. 22 the letter 'x' is recognized as 'nn' and 'nw', while in Fig. 23 it is recognized correctly.

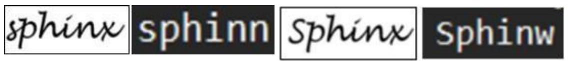


Figure 202. Letter 'x' typed text error

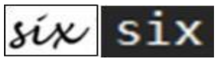


Figure 213. Letter 'x' typed text without errors

Previously mentioned Fig. 13-15 texts with different colors such as black, blue, and red were recognized without any errors. As well as the italic text shown in Fig. 12 was recognized correctly by the implemented system.

Handwritten text recognition results showed an error percentage of 30.19%, and this demonstrates the need of developing a more precise algorithm for recognizing handwritten text. As seen in Fig. 17, line 5 of the segmented lines contain a part of the line above. This is shown in Fig. 24. The issue with the recognized text can be seen clearly in Fig. 25 where the upper boundary of the image segment is shown using a red line.

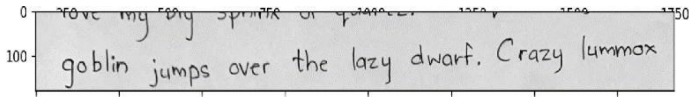
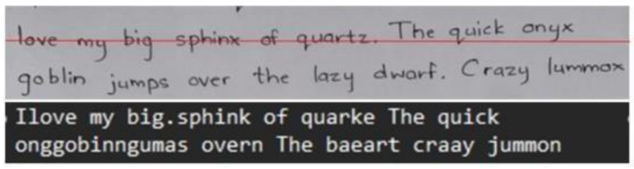


Figure 224. Line 5 of Fig. 11


love my big sphinx of quartz. The quick onyx
goblin jumps over the lazy dwarf. Crazy lummox

```
Ilove my big.sphink of quarke The quick
onggobinngumas overn The baeart craay jummon
```

Figure 235. Upper boundary of the image segment

This should be avoided for further research in order to ensure that the recognition algorithm will not take unnecessary data into consideration. By changing the parameters for the line offset this issue could be fixed. Hence, the implemented handwritten text recognition algorithm should be developed with more accuracy.

Fig.26 shows the handwritten word 'x' recognizing error.

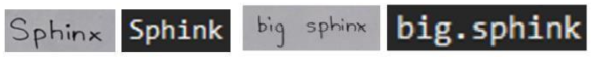


Figure 246. Handwritten word 'x' error

Moreover, as shown in Fig.27, the letter 'z' also hasn't been recognized correctly during handwritten text detection.

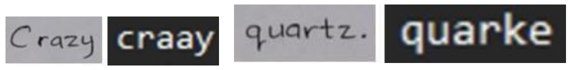


Figure 257. Handwritten letter 'z' error

The gTTS library was used to efficiently process the extracted text and produce spoken output in the implemented system. The audio output proved to be 100% accurate and using the gTTS API guaranteed high-quality, human-sounding speech synthesis, which improved the system's usability. Users can also listen to the converted text at their own pace because the spoken output can be preserved as MP3 files.

## V. Conclusion And Future Implementations

The system employs AI and computer vision methods to translate typed and handwritten text in photos into spoken format by using text-to-speech conversion algorithms. Extensive testing with a variety of fonts, colours, and handwriting has shown that the system can accurately recognize and transform text.

Error percentages for recognizing written text ranged from 0 for the most accurate font to 15.66% for the least accurate. However, there were problems with recognizing certain letters, such as the letters 's', 'x', and 'z', in particular font styles. The programme had a 30.19 percent mistake rate while trying to read handwriting. Recognizability was impacted by problems with line segmentation, which led to overlaps between neighboring lines. There is still room for improvement in the precision of handwritten text recognition.

Those who are blind or have low vision may benefit greatly from the implemented method. The implemented system can be developed to recognize texts in different languages such as Sinhala, and Tamil by improving the implemented two algorithms. Sinhala is spoken only in Sri Lanka, and its script is distinct from other South Asian languages because of its use of a cursive writing system. Vowels and modifying characters make up the Sinhala alphabet. To create the necessary vocal sound, a consonant is modified with one or more of the modifier symbols and a vowel may only occur at the beginning of a word.

There have been several major revisions to both the alphabet and the Language itself. The 60 letters that make up the contemporary Sinhala alphabet may be broken down into many separate groups. Sinhala letters are often written from left to right in horizontal lines, and their shapes are mainly spherical with no horizontal or vertical lines. Each and every Sinhala character has its own distinct form, with some sharing just superficial similarities. The Sinhala language has more than 15 adverbs and adjectives. More than a thousand letters with several pronunciations may be formed when the letters of the alphabet are combined with the aforementioned modifiers. The distinction between small and capital letters is also absent, unlike in English.

Automated segmentation of Sinhala handwriting may be challenging due to the prevalence of touching and overlapping letters in the script. Sinhala characters, in contrast to English, feature curves, making it more difficult to detect segmenting spots. The Sinhala language requires segmentation not just of individual characters but also of character modifiers. When written by hand, modifiers and their associated letters often touch or overlap. As was noted before, the identification procedure is challenging because of the distinctive cursive quality of Sinhala characters.

The southern Indian state of Tamil Nadu and the island nation of Sri Lanka are home to a combined total of several million native speakers of Tamil, one of the world's oldest languages. There are 156 letters in the Tamil alphabet, 12 of which are vowels and 23 of which are consonants (see Figure 1). Because of the bigger category set and the possibility for misunderstanding owing to similarities across handwritten letters, standalone Tamil character identification is a considerably tougher challenge than Latin character recognition.

Furthermore, the technique may be used in the future to identify the items shown in a picture.

## ACKNOWLEDGMENT

First of all, we would like to thank our family for their encouragement and support throughout the research process. We would also like to express our sincere gratitude to all the individuals and organizations that have contributed to the publication of this research paper.

## REFERENCES

[1] Bhuvaneshwari K V, Vaishnavi P, Rakshitha N, Swetha H U, Roopa Y.S, "Artificial Vision For Visually Impaired People Using Image", International Research Journal of Engineering and Technology (IRJET), vol 7, pp. 1444-1447, 2020.

[2] World Health Organization (WHO); World Bank. World Report on Disability 2011.

[3] World Health Organization (WHO), Blindness and Vision Impairment. 2021.

[4] S. Kurlekar, O.A.Deshpande, A.V.Kamble, A.A.Omanna, D.B.Patil, "Reading Device for Blind People using Python, OCR and GTTS", International Journal of Science and Engineering Applications, vol 9, pp. 49-52, 2020.

[5] Ministry of Health and Welfare. The Number of Registered Disabled Persons by Type of Disability and Gender Nationwide. 2020.

[6] Anusha Bhargava, Karthik V. Nath, PritishSachdeva and MonilSamel "Reading Assistant for the Visually Impaired", International Journal of Current Engineering and Technology,2015.

[7] H. Scheidl, "Handwritten Text Recognition in Historical Documents," B.S. Thesis, Faculty of Informatics, University of Technology, Vienna, 2018.

[8] R. Smith, "A Simple and Efficient Skew Detection Algorithm via Text Row Accumulation", Proc. of the 3rd Int. Conf. on Document Analysis and Recognition (Vol. 2), IEEE 1995, pp. 1145-1148.

[9] S.V. Rice, G. Nagy, T.A. Nartker, Optical Character Recognition: An Illustrated Guide to the Frontier, Kluwer Academic Publishers, USA 1999, pp. 57-60.

[10] R. Smith, "An overview of the Tesseract OCR engine," in IEEE, 2007, pp. 629–633.

[11] Suthar, Sanket B., et al. "Segmentation of Gujarati Handwritten Characters and Numerals with and Without Modifiers from the Scanned Document." Information and Communication Technology for Competitive Strategies (ICTCS 2022) Intelligent Strategies for ICT. Singapore: Springer Nature Singapore, 2023. pp. 335-348.

[12] ProjectPro."What is the pytesseract python library and How do you install it". https://www.projectpro.io/recipes/what-is-pytesseract-python-library-and-do-you-install-it (accessed 03 Aug 2022)

[13] Wang, T., Wu, D. J., Coates, A., & Ng, A. Y. "End-to-end text recognition with convolutional neural networks. In Pattern Recognition (ICPR)": 2012 21st International Conference on, 2012. pp. 3304-3308. IEEE.

[14] Koo, H. I., & Kim, D. H. "Scene text detection via connected component clustering and nontext filtering": IEEE transactions on image processing, 22(6), 2013. Pp. 2296-2305.

[15] Bissacco, A., Cummins, M., Netzer, Y., & Neven, H. "Photoocr: Reading text in uncontrolled conditions": In Proceedings of the IEEE International Conference on Computer Vision, 2013. pp. 785-792.

[16] Neumann, L., & Matas, J. "Real-time scene text localization and recognition. In Computer Vision and Pattern Recognition (CVPR)": 2012 IEEE Conference on, 2012. pp. 3538- 3545. IEEE.

[17] Neumann, L., & Matas, J. "Scene text localization and recognition with oriented stroke detection": In Proceedings of the IEEE International Conference on Computer Vision, 2013. pp. 97-104.