IJASER

International Journal of Applied Science and Engineering Review

ISSN: 2582-6271

Vol. 5, Issue.2, March-April 2024, page no. 134-156

To cite this article: Eva T. Roca, Ezekiel Jacob P. Sikat, Trixxie Margarette F. Garcia, Az-Zubair J. Ahmadul, Irish Yari H. Virata, Reese Ivan A. Billanes and Reese David P. Larrauri (2024). THE USE OF RASPBERRY PI AS A MAIN COMPONENT IN MAKING A TEXT RECOGNITION READER FOR CHILDREN, International Journal of Applied Science and Engineering Review (IJASER) 5 (2): 134-156 Article No. 189 Sub Id 292

THE USE OF RASPBERRY PI AS A MAIN COMPONENT IN MAKING A TEXT RECOGNITION READER FOR CHILDREN

Eva T. Roca, Ezekiel Jacob P. Sikat, Trixxie Margarette F. Garcia, Az-Zubair J. Ahmadul, Irish Yari H. Virata, Reese Ivan A. Billanes and Reese David P. Larrauri

Philippine School Doha, Doha, Qatar

DOI: https://doi.org/10.52267/IJASER.2024.5209

ABSTRACT

In contemporary society, the demands on parents' time made it challenging to engage in regular reading sessions with their children. In an effort to address this problem, this study created a text recognition reader for kids which used Raspberry Pi as the main component. The Raspberry Pi, a single-board computer that contains hardware components and sensor and controller interfaces that have user-friendly programming capabilities, high connectivity, and desktop functionality, was used to run the script that will capture an image from a webcam, binarize the captured frame, recognize the text, synthesize a voice using a reference, perform text-to-speech with the synthesized voice, and output the sound through a Bluetooth speaker. The Text Recognition Reader's results revealed that it was less accurate in detecting text from a raw image and more accurate when detecting text from a pre-preprocessed image. The Text Recognition Reader was also faster in detecting text from a pre-processed image, getting an average of 1.03 seconds, than a raw image, getting an average of 1.56 seconds. Given 50-word, 75-word, and 100-word reference text, the Text Recognition Reader was able to synthesize a voice with averages of 61.07 seconds, 81.70 seconds, and 106.6 seconds respectively. The Text Recognition Reader works more effectively and efficiently when working with pre-processed images than raw images in terms of both text detection accuracy and speed. It was also found that the processing time for voice synthesis heightened whenever the number of characters used in the utilized reference text increased. The Text Recognition Reader was able to detect text, synthesize a voice using the detected text, and read it out loud using the processed voice but also faced limitations. The Reader is highly sensitive to the lighting conditions of its environment.

KEYWORDS: Optical Character Recognition, Processing Time, Raspberry Pi, Text Recognition Reader, and Voice Synthesis.

1. INTRODUCTION

In present-day society, parents are frequently busy and have to manage various obligations, including work, household chores, and various other activities, making it difficult for them to dedicate enough time to read to their children consistently. Parents play an essential role in establishing the importance of reading and promoting positive thinking toward the activity of their children (Merga & Roni, 2018). Parental involvement is crucial for children's learning and has an important role in supporting their children's learning in educational activities, it can also promote family well-being in areas unrelated to reading and plays a significant impact in improving family reading outcomes (Ni et al., 2021). Parents who work longer hours tend to have less time for activities with their children, such as reading, which can negatively impact the child's cognitive and socio-emotional development (Kigobe, 2019). With the need for a device that would function remotely, the use of Raspberry Pi to create a Text Recognition Reader device would serve as an accessible and available tool for both parents and children.

When parents actively read books to their children, it has a major positive impact on their learning and fosters cognitive growth (Price & Kalil, 2019). Taking an active role in a child's education fosters a passion for learning by creating a supportive environment. Parental involvement in educational development is important as it has been shown to have a favorable impact on student progress and allows the parent to understand their child better (Liu et al., 2020). Parent-child book reading, particularly in the early years, fosters stronger parent-child bonds and improves academic accomplishment, significantly impacting language and literacy development and is a powerful indicator of future academic success (Xie et al., 2018).

In today's digital age, tools such as text-to-speech apps, online videos, and e-books on children's tablets and smartphones have become available. The popularity of smart mobile devices is increasing as they provide access to content through educational apps aimed at children through which they can read and learn from (Papadakis & Kalogiannakis, 2017). While phones are an option for a text recognition reader device, it would likely be the use of the parent's phone which would hamper important phone calls, messages, and notifications. Considering that only 10% of those in need genuinely have access to some assistive technology (Mensah-Gourmel et al., 2023), the lack of access to assistive devices for children can create barriers to their learning and development, especially when parents or guardians are not present to provide support. The text recognition reader can help bridge this gap by providing a familiar comforting voice for children to listen to as they engage in educational activities. This can enhance their independence and confidence and ultimately support their overall learning and development.

Parents interacting with their children by reading aloud increases their children's literacy and communication skills as they are exposed to a wide range of words and concepts that they might not encounter in their everyday interactions and allows the children to listen to a familiar voice while learning. The encouragement of print awareness through joint storytelling with a parent is crucial for the development of literacy and reading skills (Zivan & Horowitz-Kraus, 2020). With the use of a parent's synthesized voice in reading material like story books, children can feel more connected to the reading experience and become more engaged with the text. The text recognition reader offers a personalized reading and listening experience for children by simulating parent involvement to keep children motivated to learn by listening.

Raspberry Pi is a single-board computer that contains hardware components and sensor and controller interfaces that have user-friendly programming capabilities, high connectivity, and desktop functionality. It has enough peripherals (memory, CPU, power regulation) to begin running without the need for additional hardware, and it runs a complete operating system (Johnston & Cox, 2017). It includes versatile general-purpose input/output (GPIO) pins suitable for communication with various electronic devices such as LEDs, buttons, servos, motors, and sensors. Additionally, it includes a dedicated camera port (Jolles, 2021). Dela Cruz et al. (2022) employed a Raspberry Pi along with a webcam and speaker as its sub-components to develop a Face Mask Detecting Alarm System, leveraging the Pi's capability to execute various applications and perform complex computations, facilitating the coding process. With the use of a Raspberry Pi to create a text recognition reader, children without the presence of their parents can read written and printed materials that would otherwise be inaccessible to them, such as handwritten notes or documents. The device was integrated with speech synthesis and text-to-speech software to provide a personalized reading experience.

This study will benefit children who need assistance in reading different materials like storybooks when their parents are not present due to being busy with work and other responsibilities, as well as future researchers who may be interested in developing similar assistive devices. The findings of this study would be able to help children who are struggling to read or who have difficulty accessing reading materials. This study would also provide a much more affordable option as an assistive device to families or schools who may not have access to expensive assistive technology, as it uses scrap items and Raspberry Pi as its materials. Aiming to create an easy-to-use and intuitive Text Recognizing Reader, this study would especially help children who struggle to use high-end assistive devices that could be too complex for them to use.

Moreover, the results, findings, and data that will be presented in this study could be used as references for future researchers in accomplishing their studies that also deal with text recognition readers or anything similar.

1.1. Research Questions

The objective of this study is to create a Text Recognition Reader out of Raspberry Pi. Specifically, it answers the following questions:

1. What is the percentage accuracy of the Text Recognition Reader when detecting text from a:
   - 1.1 Raw Image and
   - 1.2. Pre-Processed Image?
2. How long is the response time of the Text Recognition Reader when detecting text from a:
   - 2.1 Raw Image and
   - 2.2. Pre-Processed Image?
3. How long is the processing time of the Text Recognition Reader to synthesize a voice given a reference voice to dictate text containing:
   - 3.1 50 words;
   - 3.2 75 words; and
   - 3.3 100 words?

1.2. Hypothesis

H1: It is possible to create a Text Recognition Reader for Children using Raspberry Pi as the main component.

2. METHODOLOGY

This study utilized the experimental design of research. Experimental researches seek to determine a relationship between the dependent and independent variables to be manipulated, measured, calculated, and compared (Singh, 2021). In this study, the Raspberry Pi is the independent variable, and the Text Recognition Reader is the dependent variable. Moreover, the quantitative method was used to quantify and analyze variables which involves the utilization and analysis of numerical data (Apuke, 2017).

2.1. Research Locale

The research study was conducted at Philippine School Doha as it will allow for the writing of the research paper and the making of the proposed product. Philippine School Doha is located in Doha, State of Qatar, Mesaimeer Area (Zone 56), Al Khulaifat Al Jadeeda Street (St. 1011).

2.2. Data Gathering Procedure

The procedure shows the step-by-step process of how to make a text recognition reader with the use of Raspberry Pi as the main component.

2.2.1. Ensuring security and upholding safety standards

(a) Wear personal protection equipment such as safety goggles, gloves, shoes, and a laboratory coat to prevent hazardous situations.

2.2.2. Setting up the Raspberry Pi

(a) Prepare a Raspberry Pi, MicroSD Card, and a Computer/Laptop.
(b) Connect the MicroSD Card to a computer or laptop.
(c) Install the Raspberry Pi Imager from https://www.raspberrypi.com/software/ and complete the setup by running the imager.
(d) Click the ‘Choose Device’ option and select ‘Raspberry Pi 4’.
(e) Click the ‘Choose OS’ option and select ‘Raspberry Pi OS Full (64-Bit)’ under Raspberry Pi OS (other).
(f) Click the ‘Choose Storage’ option and select your MicroSD Card.
(g) Click ‘Next’ and select the ‘Edit settings’ option when the OS customization prompt shows up.
(h) Tick the ‘Set hostname’ box and leave the default hostname (raspberrypi.local).
(i) Tick the ‘Set username and password’ box and set a desired username and password.
(j) Tick the ‘Enable SSH’ box and select the ‘Use password authentication’ option.
(k) Save and apply the setting changes made and proceed with the installation.
(l) Eject the MicroSD Card from the computer or laptop once installation is complete.
(m) Insert the MicroSD Card into its respective compartment in the Raspberry Pi.

2.2.3. SSH into the Raspberry Pi

(a) Login to your router’s website using a set username and password.
(b) Check for a device named ‘raspberry pi’ and take note of its IP address.
(c) Open the Windows Command Prompt as administrator and run the following command:

(1) ssh <username>@<raspberrypi_ipaddress>

2.2.4. Accessing and Coding the Software

(a) Download the XTTS folder from the following link:
(b) Extract the folder in a known location and take note of its directory.
(c) Using the Windows Command Prompt as administrator, transfer the folder from your desktop or laptop to the Raspberry Pi

(d) Install the Python Version Manager:

(1) curl https://pyenv.run | bash

(e) Access and open the bashrc file as administrator:

(1) sudo nano ~/.bashrc

(f) Add pyenv to PATH by pasting the following to the bashrc script:

(1) export PATH="$HOME/.pyenv/bin:$PATH”

(2) eval "$(pyenv init --path)"

(3) eval "$(pyenv virtualenv-init -)"

(g) Reset the Shell Terminal:

(1) export PATH="$HOME/.pyenv/bin:$PATH”

(h) Install all the required system packages to run pyenv:

(1) sudo apt-get install --yes libssl-dev zlib1g-dev libbz2-dev libreadline-dev libsqlite3-dev llvm libncurses5-dev libncursesw5-dev xz-utils tk-dev libgdbm-dev lzma lzma-dev tcl-dev libxml2-dev libxmlsec1-dev libffi-dev liblzma-dev wget curl make build-essential openssl

(i) Update pyenv:

(2) pyenv update

(i) Install Python 3.9.10 to support the main script:

(1) pyenv install 3.9.10

(j) Change the current directory to the folder installed in steps 1-3:

(1) CD XTTS

(k) Change the Python version of the XTTS directory to 3.9.10:

(1) pyenv local

(l) Install all the required packages to run the main script:

(1) pip install -r requirements.txt

2.2.5. Establishing Bluetooth connection with the Raspberry Pi

(a) Run the following commands on the Raspberry Pi Console:

(1) bluetoothctl

(2) power on

(3) scan on

(4) pair <macaddress_of_bluetooth_speaker>

(5) trust <macaddress_of_bluetooth_speaker>

(6) connect <macaddress_of_bluetooth_speaker>

2.2.6. Allowing Independent Execution of the Script on Startup

(a) Create a .sh script (e.g. startup.sh) in a folder relevant to your project:

(1) sudo nano startup.sh

(b) Paste the following code to the startup.sh file:

(1) #!/bin/bash

(2) cd user/directory/XTTS

(3) source .venv/bin/activate

(4) bluetoothctl

(5) power on

(6) connect <macaddress_of_bluetooth_speaker>

(7) exit

(8) python finalscript.py

(c) Set executable permission for startup.sh:

(1) chmod +x user/directory/XTTS/startup.sh

(d) Create an autorun desktop file in the Raspberry Pi's autostart directory:

(1) cd /home/user/.config/autostart

(2) sudo nano autorun.desktop

(e) Paste the following script inside autorun.desktop:

(1) [Desktop Entry]

(2) Exec=user/directory/XTTS/startup.sh

(3) Terminal=true

The script should run at startup whenever the Raspberry Pi either restarts or boots up.

2.2.7. Preparing Needed Materials for the Base and Platforms

(a) Prepare pieces of scrap plywood that have the dimensions of at least 15 inches by 10 inches.

(b) Be sure that the woodcutting machine is plugged and prepared for use.

(c) Have the small nails, screwdriver, sandpaper, ruler, and pencil prepared at hand.

2.2.8. Making the Base of the Box

(a) Mark and cut out plywood with the dimensions of 13 by 9 inches for the bottom of the base.

(b) For the left and right side walls (sides) of the box, mark and cut out two pieces of plywood with the dimensions of 9 by 2 inches.

(c) Cut out a small hole on one side of a piece in order to create a space for the cables to pass through.

(d) For the front side wall of the box, mark and cut out a piece of plywood with the dimensions of 13 by 0.5 inches.

(e) For the back side wall of the box, mark and cut out two pieces of plywood with the dimensions of 4 by 3 inches.

(f) Lay out the pieces of the cut plywood for all sides of the box and smooth out their edges with the sandpaper.

(g) Secure the pieces in place using the hammer and small nails, making sure the two pieces of plywood for the back side wall of the box are nailed to both left and right side of the box in order to create a gap in the center for the mobile phone holder to clamp on.

2.2.9. Creating the Platform for Storybooks

(a) Mark and cut out a piece of plywood with the dimensions of 13.5 by 9.5 inches.

(b) For the left and right side walls (sides) of the platform, mark and cut out two pieces of plywood with the dimensions of 9.5 by 0.5 inches.

(c) For the front side wall of the platform, mark and cut out a piece of plywood with a dimension of 13.5 by 0.5 inches.

(d) Lay out the pieces of the cut plywood for all sides of the platform and smooth out their edges with the sandpaper.

(e) Secure the pieces in place using the hammer and small nails.

2.2.10. Securing the Webcam on the Clamp

(a) Open the mobile phone holder.

(b) Place the webcam onto the mobile phone holder and check if its view is obstructed in any way, adjusting its position as needed.

(c) Secure the webcam onto the mobile phone holder using pieces of cut tape with scissors.

2.2.11. Testing and Adjusting the Box Pieces’ Fitting

(a) Place the platform for the storybooks on top of the base of the box.

(b) If there are ill-fitting pieces of any part of the box, mark them with a pencil and cut them off.

2.2.12. Assembling the Entire Device

(a) Assemble all the product components by first clamping the webcam that is attached to the phone holder on the back of the wooden box.

(b) Place the bluetooth speaker and raspberry pi on the inside of the box, connecting the webcam wire and raspberry pi power cable onto the raspberry pi. The cables should pass through the hole made on the side of the box.

(d) Place the desired storybook on top of the device to prepare for scanning.

(e) Adjust the position of the webcam as needed.

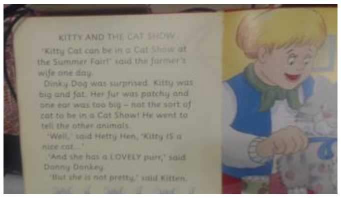
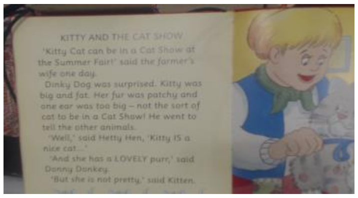
3. RESULTS

This section presents the results and interpretations of the data that were collected during the testing procedure in relation to the research questions.

3.1 Accuracy of the Text Recognition Reader when detecting text from a:

3.1.1. Raw Image

Table 1: Accuracy of the Text Recognition Reader Using a Raw Image Reference

<table>
<tr><th>Trial</th><th>Actual No. of Characters</th><th>Correctly Recognized Characters</th><th colspan="3">Incorrectly Recognized Characters</th><th>Accuracy</th></tr>
<tr><td rowspan="2">1</td><td>398</td><td>368</td><td colspan="3">134 characters</td><td rowspan="2">58.79%</td></tr>
<tr><td colspan="2">
KITTY AND THE CAT SHOW
'Kitty Cat can be in a Cat Show at the Summer Fair!' said the farmer's wife one day.
Dinky Dog was surprised. Kitty was big and fat. Her fur was patchy and one ear was too big – not the sort of cat to be in a Cat Show! He went to tell the other animals.
'Well,' said Hetty Hen, 'Kitty IS a nice cat…'<br>'And she has a LOVELY purr,' said Donny Donkey.
'But she is not pretty,' said Kitten.</td><td>104 Extra</td><td>28 Missing</td><td>2 Misinterpreted</td></tr>
<tr><td rowspan="2">2</td><td>398</td><td>364</td><td colspan="3">160 characters</td><td rowspan="2">51.26%</td></tr>
<tr><td colspan="2">
KITTY AND THE CAT SHOW
'Kitty Cat can be in a Cat Show at the Summer Fair!' said the farmer's wife one day.
Dinky Dog was surprised. Kitty was big and fat. Her fur was patchy and one ear was too big – not the sort of cat to be in a Cat Show! He went to tell the other animals.
'Well,' said Hetty Hen, 'Kitty IS a nice cat…'<br>'And she has a LOVELY purr,' said Donny Donkey.
'But she is not pretty,' said Kitten.</td><td>126 Extra</td><td>34 Missing</td><td>0 Misinterpreted</td></tr>
</table>

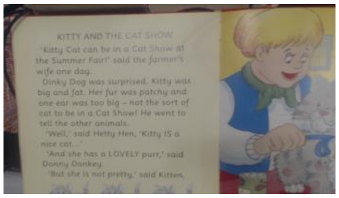
3	398	359	175 characters			**46.23%**
	 KITTY AND THE CAT SHOW 'Kitty Cat can be in a Cat Show at the Summer Fair!' said the farmer's wife one day. Dinky Dog was surprised. Kitty was big and fat. Her fur was patchy and one ear was too big – not the sort of cat to be in a Cat Show! He went to tell the other animals. 'Well,' said Hetty Hen, 'Kitty IS a nice cat...' 'And she has a LOVELY purr,' said Donny Donkey. 'But she is not pretty,' said Kitten.		136 Extra	38 Missing	1 Misinterpreted	
Average Accuracy:			$\frac{(58.79+51.26+46.23)}{3}$			**52.09%**

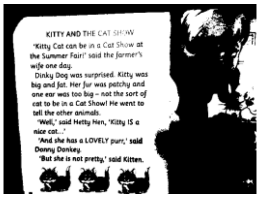
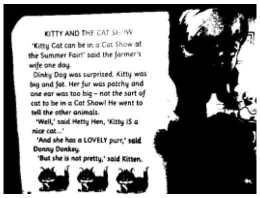
Table 1 shows the accuracy of the text recognition reader when detecting text from a page of a storybook. When detecting the text, the script did not pre-process the captured frame and instead used the raw image captured the moment the script was run. The accuracy of the text recognition reader was calculated by the formula, (Correctly Recognized Characters - Incorrectly Recognized Characters)/ Actual Number of Characters. The first trial recognized 368 characters correctly and 134 characters incorrectly, 104 of which were extra characters, 28 were missing characters, and 2 of which were characters misinterpreted by the text recognition reader. The first trial had an accuracy of 58.79% making it the most accurate among the three trials. The second trial recognized 364 characters correctly and 160 characters incorrectly, 126 of which were extra characters, 34 of which was a missing character, and 0 of which were characters misinterpreted by the text recognition reader. The second trial measured an accuracy of 51.26%. Lastly, the third trial recognized 359 characters correctly and 175 characters incorrectly, 136 of which were extra characters, 38 were missing characters, and 1 of which was a character misinterpreted by the text recognition reader, with an average of 46.23%.

Overall, the text recognition reader showed an average accuracy of 52.09% in the three trials when detecting text from a raw image reference. Based on the results, when a raw image is used as the reference text, the device performs less accurately. In comparison to an image that underwent binarization, thresholding, and contrast adjustment, the reader's accuracy in identifying text from an unedited, raw image was 36.69% poorer across the same three text prompts. According to these findings, using preprocessing methods would be beneficial as they could increase the accuracy of text recognition when working with raw photos.

The results of Table 1 agreed with findings from a similar study that observed that text recognition from natural photos is still a difficult task (Cheng et al., 2018). Complex images, specifically those that have not been pre-processed through techniques like binarization, thresholding, and contrast adjustment present additional challenges for OCR (Optical Character Recognition) systems. Binarization simplifies the image by converting the pixels into either black or white. This simplification helps enhance the contrast between the text and the background making it easier for the text recognition reader to recognize text. A study by Cheng et al. (2017) highlights a difficulty that text recognition systems face when dealing with complex and/or low-quality images which lead to poor results.

3.1.2. Pre-Processed Image

Table 2: Accuracy of the Text Recognition Reader Using a Pre-Processed Image Reference

Trial	Actual No. of Characters	Correctly Recognized Characters	Incorrectly Recognized Characters			Accuracy
1	398	395	60 characters			**83.92%**
	 KITTY AND THE CAT SHOW 'Kitty Cat can be in a Cat Show at the Summer Fair!' said the farmer's wife one day. Dinky Dog was surprised. Kitty was big and fat. Her fur was patchy and one ear was too big – not the sort of cat to be in a Cat Show! He went to tell the other animals. 'Well,' said Hetty Hen, 'Kitty IS a nice cat...' 'And she has a LOVELY purr,' said Donny Donkey. 'But she is not pretty,' said Kitten.		56 Extra	0 Missing	4 Misinterpreted	
2	398	395	29 characters			**91.96%**
	 KITTY AND THE CAT SHOW 'Kitty Cat can be in a Cat Show at the Summer Fair!' said the farmer's wife one day. Dinky Dog was surprised. Kitty was big and fat. Her fur was patchy and one ear was too big – not the sort of cat to be in a Cat Show! He went to tell the other animals. 'Well,' said Hetty Hen, 'Kitty IS a nice cat...' 'And she has a LOVELY purr,' said Donny Donkey. 'But she is not pretty,' said Kitten.		26 Extra	1 Missing	2 Misinterpreted	
3	398	394	34 characters			**90.45%**

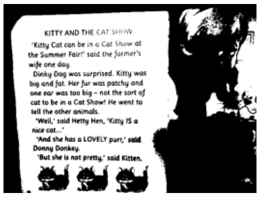
	 KITTY AND THE CAT SHOW 'Kitty Cat can be in a Cat Show at the Summer Fair!' said the farmer's wife one day. Dinky Dog was surprised. Kitty was big and fat. Her fur was patchy and one ear was too big – not the sort of cat to be in a Cat Show! He went to tell the other animals. 'Well,' said Hetty Hen, 'Kitty IS a nice cat...' 'And she has a LOVELY purr,' said Donny Donkey. 'But she is not pretty,' said Kitten.	30 Extra	1 Missing	3 Misinterpreted	
Average Accuracy:		$\frac{(83.92+91.96+90.45)}{3}$			**88.78%**

Table 2 shows the accuracy of the text recognition reader when detecting text from a page of a storybook. Unlike the trials from Table 1, the script pre-processed the raw image before detecting the text by binarizing, thresholding, and adjusting the contrast of the captured frame. The text recognition reader's accuracy was calculated with the same formula used for Table 1. The first trial recognized 394 characters correctly and 60 characters incorrectly, 56 of which were extra characters, 0 were missing characters, and 4 of which were characters misinterpreted by the text recognition reader. Trial 1 had an accuracy of 83.92% making it the least accurate among the three trials. The second trial recognized 395 characters correctly and 29 characters incorrectly, 26 of which were extra characters, 1 of which was a missing character, and 2 of which were characters misinterpreted by the text recognition reader. Trial 2 had the highest accuracy among the three trials with an accuracy of 91.96%. Lastly, the third trial recognized 394 characters correctly and 34 characters incorrectly, 30 of which were extra characters, 1 were missing characters, and 3 of which were characters misinterpreted by the text recognition reader., with an accuracy of 90.45%.

Overall, the text recognition reader showed an average accuracy of 88.78% in the three trials when detecting text from a pre-processed image reference. The findings show that the accuracy of the Text Recognition Reader is higher when using a pre-processed image as the reader's reference text. Using the same three text prompts, the reader was 36.69% more accurate in detecting text from an image that has been binarized and thresholded and whose contrast has been adjusted than an unprocessed image. Based on the results of the text recognition reader in reading text from pre-processed images, it is evident that pre-processing techniques impact the accuracy level of the device's optical character recognition process. Specifically, the findings of the study revealed that pre-processing techniques such as binarization, thresholding, and contrast adjustment can enhance the clarity and quality of text within the image which makes it easy for the text recognition reader to more accurately detect and interpret the text. High accuracy in text recognition can make information more accessible to children who are reading from the reader as well as to individuals with visual impairments due to the text recognition reader's reliability.

Preprocessing entails performing various operations on the input or scanned image. Its main purpose is to enhance image quality for segmentation by removing noise and enhancing the readability of characters (Awel & Abidi, 2019). A study by Mathur and Rikhari (2017) described OCR as a process containing many phases, including the pre-processing of the image. It was also stated that scanned images that undergo pre-processing and segmentation are not fully accurate due to the presence of noise and unnecessary details which cause disruptions in the detection of the characters in the image. Without image filtering, OCR software could have trouble correctly identifying characters from the photos, possibly due to inaccurate information interpretation (Maliński, et al., 2023). Both studies proved that pre-processing images is a necessary stage of optical character recognition that makes character detection easier and more accurate for text recognition readers.

3.2 Response time of the Text Recognition Reader when detecting text from a:

3.2.1. Raw Image

Table 3: OCR Response time using a Raw Image Reference

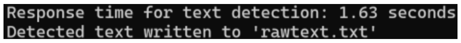
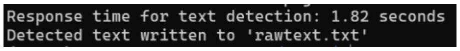
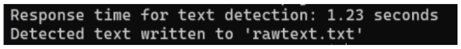
Trial	No. of Characters Recognized	Photos	Response Time (in seconds)
1	398 Characters	 Response time for text detection: 1.63 seconds Detected text written to 'rawtext.txt'	1.63 seconds
2		 Response time for text detection: 1.82 seconds Detected text written to 'rawtext.txt'	1.82 seconds
3		 Response time for text detection: 1.23 seconds Detected text written to 'rawtext.txt'	1.23 seconds
Average Response Time	$\frac{(1.63+1.82+1.23)}{3}$		1.56 seconds

Table 3 shows the response time in seconds of the text recognition reader when detecting text from a raw image. The script of the text recognition reader was coded to specifically measure the time it took for the optical character recognition (OCR) process to analyze the captured image and recognize the text within the frame. In each trial, the response time of the text recognition reader when detecting text from the same three text prompts used in the respective trials in Table 1 was measured. The average response time of the text recognition reader was calculated by getting the sum of the response times of all three trials and dividing it by the number of trials performed.

All three trials are given the same number of characters recognized for consistency. Trial 1 recognized 398 characters from the given prompt in 1.63 seconds. This shows a detection speed of 244.17 characters per second for the first trial. Trial 2 measured 1.82 seconds when recognizing the given 398-character prompt which shows a detection speed of 218.68 characters per second for the second trial. Lastly, Trial 3 recognized 398 characters from the given prompt in 1.23 seconds, which shows a detection speed of 323.58 characters per second. Table 3 opposes the study of Reul et al. (2018) wherein using numerous books that share typeface similarities will produce a generic model with extremely low error rates across a wide range of fonts.

3.2.2. Pre-Processed Image

Table 4: OCR Response Time using a Pre-Processed Image Reference

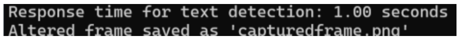
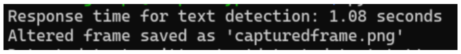
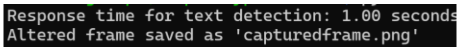
Trial	No. of Characters Recognized	Photos	Response Time (in seconds)
1	398 Characters	 Response time for text detection: 1.00 seconds Altered frame saved as 'capturedframe.png'	1.00 second
2		 Response time for text detection: 1.08 seconds Altered frame saved as 'capturedframe.png'	1.08 seconds
3		 Response time for text detection: 1.00 seconds Altered frame saved as 'capturedframe.png'	1.00 second
Average Response Time	$\frac{(1.00+1.08+1.00)}{3}$		1.03 seconds

Table 4 shows the response time in seconds of the text recognition reader when detecting text from a pre-processed image. Likewise, the script of the text recognition reader uses the same code from Table 3, the only difference being that the reference image was pre-processed before the text was detected and its duration was measured. Assessing the results of the three trials, the average response time of the text recognition reader when detecting text from a pre-processed frame is 1.03 seconds which is 0.53 seconds faster than the average when detecting text from a raw image. This suggests that the text recognition reader is quick when it comes to detecting the displayed text as it was able to recognize text prompts provided with 398 characters in less than 1 second on average.

Trial 1 recognized 398 characters from the given prompt in 1.00 seconds. This shows a detection speed of 398 characters per second for the first trial. Trial 2 measured 1.08 seconds when recognizing the given 398-character prompt which shows a detection speed of 368.52 characters per second for the second trial. Lastly, Trial 3 recognized 398 characters from the given prompt in 1.00 seconds, which is the same duration observed in Trial 1. Trial 3 shows a detection speed of 398 characters per second.

In digital image processing and analysis, tasks such as moving object detection or locating the region of interest in an image, such as the text in a damaged document, image binarization is a necessary pre-processing step. For a particular application, binary images require less storage memory and facilitate faster computations, document skew detection, and document layout analysis (Jindal et al. 2021).

3.3 Processing time of the Text Recognition Reader to synthesize a voice given a reference voice to dictate text containing:

3.3.1. 50 words

Table 5: Voice Synthesis Processing Time given a 50-word reference text

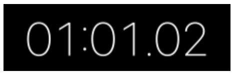
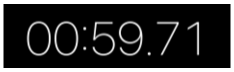
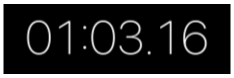
Trial	Reference Voice Used	Voice Synthesis Duration	
1	Person A	 01:01.02	61.02 seconds
2	Person B	 00:59.71	59.71 seconds
3	Person C	 01:03.16	63.16 seconds
4	Person D	01:00.39	60.39 seconds
Average Duration		$\frac{(61.02 + 59.71 + 63.16 + 60.39)}{4}$	= 61.07 seconds

Table 5 shows the voice synthesis processing time of the text recognition reader given a 50-word reference text in seconds. The duration for the voice synthesis of the four reference voices was measured using a stopwatch, starting when the voice synthesis script was run and stopping when a new voice had been synthesized. The voice references consist of the four speakers dictating the same body of text, all lasting 10 seconds. The average duration was calculated by summing the voice synthesis processing time of all four trials and dividing the sum by the number of trials. In trial 1, the voice of Person A was used as the reference voice which had a voice synthesis duration of 61.02 seconds. In trial 2, the voice of Person B was used as the reference voice which had a voice synthesis duration of 59.71 seconds. Trial 2 was observed to have the fastest voice synthesis processing time among all four trials. In trial 3, the voice of Person C was used as the reference voice which had a voice synthesis duration of 63.16 seconds. When compared to all four trials, trial 3 is observed to have the slowest voice synthesis processing time. Lastly, in trial 4, the voice of Person D was used as the reference voice. Lastly, Trial 4 had a voice synthesis processing time of 60.39 seconds. Interpreting the given data, the text recognition reader had an average voice synthesis processing time of 61.07 seconds given a 50-word reference text with the four trials.

3.3.2. 75 words

Table 6: Voice Synthesis Processing Time given a 75-word reference text

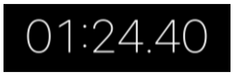
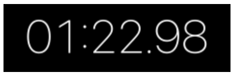
Trial	Reference Voice Used	Voice Synthesis Duration	
1	Person A	01:20.25	80.25 seconds
2	Person B	 01:24.40	84.40 seconds
3	Person C	 01:22.98	82.98 seconds
4	Person D	01:19.15	79.15 seconds
Average Duration		$\frac{(80.25 + 84.4 + 82.98 + 79.15)}{4}$	= 81.70 seconds

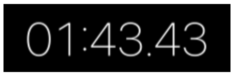
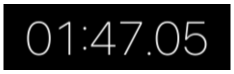
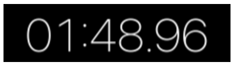
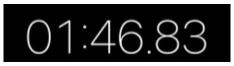
Table 6 shows the voice synthesis processing time of the text recognition reader given a 75-word reference text in seconds. In trial 1, Person A was used for the reference voice and had a voice synthesis duration of 80.25 seconds. Compared to the first trial in Table 5 given a 50-word reference text which resulted in a processing time of 61.02 seconds, the processing time has increased by 19.23

seconds. In trial 2, Person B was used for the reference voice and had a voice synthesis duration of 84.40 seconds making it the slowest processing time of all four trials. Compared to the second trial in Table 5 given a 50-word reference text which resulted in a processing time of 59.71 seconds, the processing time has increased by 24.69 seconds. In trial 3, Person C was used for the reference voice and was observed to only have a slight decrease in duration compared to trial 2, with a voice synthesis processing time of 82.98 seconds. Compared to the third trial in Table 5 given a 50-word reference text, which resulted in a processing time of 63.16 seconds, the processing time has increased by 19.82 seconds. Lastly, Person D was used as the reference voice for Trial 4 and had a voice synthesis duration of 79.15 seconds. This is also the fastest processing time among the four trials. Compared to the fourth trial in Table 5 given a 50-word reference text which resulted in a processing time of 60.39 seconds, the processing time has increased by 18.76 seconds.

Evaluating the results, there has been an increase in the average voice synthesis processing time when the text recognition reader is compared to Table 5 which was given a 50-word reference text to a 75-word reference text which had an average voice synthesis processing time of 81.70 seconds with the four trials. Moreover, a study by Zhang et al. (2021) compared the datasets of both the text-to-speech systems and the voice conversion systems. TTS systems are often trained with a large dataset of words while voice conversion systems face constraints due to limited datasets. Both this and the study support that the voice synthesis process naturally takes long periods of time when incorporated with text-to-speech systems due to the challenges of adapting voice characteristics from limited data sources in voice conversion systems.

3.3.3. 100 words

Table 7: Voice Synthesis Processing Time given a 100-word reference text

Trial	Reference Voice Used	Voice Synthesis Duration	
1	Person A	 01:43.43	103.43 seconds
2	Person B	 01:47.05	107.05 seconds
3	Person C	 01:48.96	108.96 seconds
4	Person D	 01:46.83	106.83 seconds

Average Duration	$\frac{(103.43 + 107.05 + 108.96 + 106.83)}{4}$ = 106.57 seconds

Table 7 shows the voice synthesis processing time of the text recognition reader given a 100-word reference text in seconds. The duration for the voice synthesis of the four reference voices was measured using a stopwatch, starting when the voice synthesis script was run and stopping when a new voice had been synthesized. In trial 1, the voice of Person A was used as the reference voice which had a voice synthesis duration of 103.43 seconds, wherein this was observed to have the fastest voice synthesis processing time among all four trials. Compared to the 75-word reference text, the voice synthesis processing time of Person A increased by 23.18 seconds. In trial 2, the voice of Person B was used as the reference voice which had a voice synthesis duration of 107.05 seconds. Compared to the 75-word reference text, the voice synthesis processing time of Person B increased by 22.65 seconds. In trial 3, the voice of Person C was used as the reference voice which had a voice synthesis duration of 108.96 seconds. When compared to all four trials, trial 3 is observed to have the slowest voice synthesis processing time. Furthermore, when compared to the 75-word reference text, the voice synthesis processing time of Person C increased by 25.98 seconds, the largest increase among all four trials. Lastly, in trial 4, the voice of Person D was used as the reference voice with a voice synthesis processing time of 106.83 seconds. Compared to the 75-word reference text, the voice synthesis processing time of Person D increased by 27.68 seconds. Interpreting the given data, the text recognition reader had an average voice synthesis processing time of 106.57 seconds given a 100-word reference text with the four trials.

The results showed that there is a correlation between the processing time of voice synthesis and the length of the reference texts used to produce the voice. This suggests that longer text prompts require more time for voice synthesis. Training on a large number of high-quality speech-transcript pairs is necessary for synthesizing natural speech, and supporting several speakers often requires tens of minutes of training data per speaker (Jia et al., 2019).

3.4 Hypothesis

The research's alternative hypothesis which states that it is possible to create a Text Recognition Reader for Children using Raspberry Pi as the main component is accepted. The researchers were able to construct a working program that can perform Optical Character Recognition to detect and recognize text from a storybook and synthesize a voice given a reference voice to dictate the recognized text.

4. DISCUSSION

Voice synthesis replicates voices, enhancing immersive learning for children in speech-related activities, such as reading storybooks. Text-to-speech has revolutionized textual content, providing equitable access to knowledge, enjoyment, and education, especially benefiting those with learning difficulties (Harini & Manoj, 2024). This technology aids children in improving the reading comprehension of students who use text-to-speech, showing increased material consumption while experiencing less fatigue and stress (Keelor et al., 2020). Utilizing a Raspberry Pi as a main component, this study aimed to create a text recognition reader that can detect, recognize, and dictate text using a synthesized voice.

This research mainly sought to measure the effectiveness of the text recognition reader aspect of the device in terms of its accuracy or correctness when reading detected text and the time it took to recognize the text. This test was done with the device using both a raw and pre-processed image as its reference. The effectiveness of the voice synthesizer aspect of the device was assessed by measuring the speed of the voice synthesis process given different reference voices and text prompts. By comparing the results of the raw image and pre-processed trials, it was found that the text recognition reader works more efficiently when using pre-processed images as its reference in terms of both text recognition accuracy and speed.

The results in Table 1 show that the text recognition reader had a difficult time in processing the text that was detected as shown in its average accuracy of 52.09% When the device used a raw image as its reference, most of the incorrectly recognized characters came from the extra characters that it detected. The results in Table 2 show that the text recognition reader is accurate when detecting text from a pre-processed image with an average accuracy of 88.77%. The results from Table 3 and 4 show that the text recognition reader detects and recognizes text faster from a pre-processed image with an average duration of 1.02 seconds than from a raw image which had an average of 1.56 seconds. This result is supported by a study by Michalak & Okarma (2019) which stated that binarizing images facilitates quicker processing, analysis, and text recognition. Lastly, the results in Tables 5-7 show that the duration of the text recognition reader's speech synthesis is proportional to the length of the reference text that it used. The more characters were in the reference text, the longer the voice synthesis process took. The slight differences in the voice synthesis duration can be caused by the difference of speaker's pronunciation and intonation of the words. Systems which use voice synthesis determine several factors of speech which differ from person to person such as the proper pronunciation of words, abbreviations, specialist terms, names and other words (Kuligowska et al., 2018). Therefore, by using a pre-processed image as its reference, the text recognition reader can detect text from a storybook quickly but can only synthesize a voice to dictate a 100-word reference text in an average duration of 107 seconds.

The efficiency of the text recognition reader is higher in terms of both accuracy and speed when detecting text from a pre-processed image that has been binarized, thresholded, and whose contrast has been adjusted (88.77% and 1.02 seconds) than from a raw image (52.09% and 1.56 seconds). Increasing image contrast aids in proper binarization and alphanumeric character recognition, enhancing response time. Optimizing photo pre-processing through improved lighting, shadow prevention on storybook pages, reduced camera angle distortion, and clear character shapes enhances OCR software accuracy.

This study could serve as a model for future researchers developing projects which employ text-recognition and voice synthesis. Future researchers could explore more advanced TTS models with better speech synthesis technology that can synthesize faster and/or clone a voice that is close to the original. Training a personalized TTS model for customized voices can also make synthesizing voices faster and enables the creation of voices that use different accents, languages, and dialects.

Though the study was able to prove that it was feasible to make a text recognition reader with voice synthesis features using a Raspberry Pi as its main component, several limitations of the Raspberry Pi emerged during the procedure and data gathering procedure. The Raspberry Pi requires exact adjustments for maximum operation due to its great sensitivity to light conditions. To guarantee optimal performance, the lighting environment must often be carefully adjusted in order to achieve optimal results, which might be laborious. The Raspberry Pi is configured to use desktop peripherals in order to operate it and write software within the device. Consequently, making any changes to a code like adding a new reference voice for synthesis or debugging any errors may not be an option when not connected to a network. The Raspberry Pi and the laptop or computer used to write the code must be connected to the same wired or wireless network. The Raspberry Pi needs to be connected to the same wireless or wired network as the computer or laptop that will be used to write the code. The researchers also suggest utilizing a more powerful microprocessor instead of the Raspberry Pi Model 4. Upgrading the computer with additional processor cores and a CUDA-enabled graphics card can make the scripts run significantly faster.

REFERENCES

[1] Apuke, O. (2017). Quantitative research methods: A synopsis approach. Arabian Journal of Business and Management Review, 6(10). https://doi.org/10.12816/0040336

[2] Awel, M., Abidi, A. (2019). REVIEW ON OPTICAL CHARACTER RECOGNITION. International Research Journal of Engineering and Technology (IRJET), 6(6), pp. 3666-3669. Retrieved from https://www.researchgate.net/publication/334162853_REVIEW_ON_OPTICAL_CHARACTER_RECOGNITION

[3] Cheng, Z., Xu, Y., Bai, F., Niu, Y., Pu, S., & Zhou, S. (2018). AON: Towards Arbitrarily-Oriented Text Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5571-5579. https://doi.org/10.48550/arXiv.1711.04226

[4] Cheng, Z., Bai, F., Xu, Y., Zheng, G., Pu, S., & Zhou, S. (2017). Focusing Attention: Towards Accurate Text Recognition in Natural Images. 2017 IEEE International Conference on Computer Vision (ICCV), pp. 5076-5084. https://doi.org/10.1109/ICCV.2017.543

[5] Dela Cruz, M., Fortes, M., Soriano, E., Reyes, J., Tiangco, R., & Felicitas, J. (2022). The Creation of a Face Mask Detecting Alarm System with the Use of Raspberry Pi as a Component. International Journal of New Technology and Research, 9(3). https://doi.org/10.31871/ijntr.9.3.4

[6] Harini, S., & Manoj, G. M. (2024). Text to Speech Synthesis. Arxiv. https://arxiv.org/abs/2401.13891

[7] Jia, Y., Zhang, Y., Weiss, R., Wang, Q., Shen, J., Ren, F., ... & Wu, Y. (2019). Transfer learning from speaker verification to multispeaker text-to-speech synthesis. Advances in neural information processing systems, 31. https://doi.org/10.48550/arXiv.1806.04558

[8] Jindal, H., Kumar, M., Tomar, A., & Malik, A. (2021). Degraded Document Image Binarization using Novel Background Estimation Technique. 2021 6th International Conference for Convergence in Technology (I2CT), 1-8. https://doi.org/10.1109/I2CT51068.2021.9418084

[9] Johnston, S. J., & Cox, S. J. (2017). The raspberry Pi: A technology disrupter, and the enabler of dreams. Electronics, 6(3), 51. https://doi.org/10.3390/electronics6030051

[10] Jolles, J. (2021). Broad-scale applications of the Raspberry Pi: A review and guide for biologists. Methods in Ecology and Evolution, 12(9), 1562– 1579. https://doi.org/10.1111/2041-210X.13652

[11] Keelor, J., Creaghead, N., Silbert, N., Horowitz-Kraus, T., & Author, C. (2020). Text-to-Speech Technology: Enhancing Reading Comprehension for Students with Reading Difficulty. Assistive Technology and Outcome Benefits, 14, 19–35. Retrieved from https://www.atia.org/wp-content/uploads/2020/06/ATOB-V14-A2-Keelor_etal.pdf

[12] Kigobe, J. (2019). Parental involvement in literacy development of primary school children in Tanzania. Digital Library of Open University of Tanzania, Retrieved from https://www.core.ac.uk/download/pdf/286426398.pdf

[13] Kuligowska, K., Kisielewicz, P., & Włodarz, A. (2018). Speech synthesis systems: disadvantages and limitations. International Journal of Engineering & Technology, 7(2.28), 234.

https://doi.org/10.14419/ijet.v7i2.28.12933

[14] Liu, Y., Sulaimani, M., & Henning J. (2020). The Significance of Parental Involvement in the Development in Infancy. Journal of Educational Research & Practice, 10(1), 161-166. https://doi.org/10.5590/JERAP.2020.10.1.11

[15] Maliński, K., & Okarma, K. (2023). Analysis of Image Preprocessing and Binarization Methods for OCR-Based Detection and Classification of Electronic Integrated Circuit Labeling. Electronics, 12(11):2449. https://doi.org/10.3390/electronics12112449

[16] Mathur, G., & Rikhari, S. (2020). Text Detection in Document Images: Highlight on using FAST algorithm. International Journal of Advanced Engineering Research and Science, 4(3), 275-284. https://dx.doi.org/10.22161/ijaers.4.3.43

[17] Mensah-Gourmel, J., Thépot, M., Gorter, J. W., Bourgain, M., Kandalaft, C., Chatelin, A., ... & Pons, C. (2023). Assistive Products and Technology to Facilitate Activities and Participation for Children with Disabilities. International Journal of Environmental Research and Public Health, 20(3), 2086. https://doi.org/10.3390/ijerph20032086

[18] Merga, M., & Roni, S., M. (2018). Empowering Parents to Encourage Children to Read Beyond the Early Years. The Reading Teacher, 72(2), 213-221. https://doi.org/10.1002/trtr.1703

[19] Michalak, H., & Okarma, K. (2019). Improvement of Image Binarization Methods Using Image Preprocessing with Local Entropy Filtering for Alphanumerical Character Recognition Purposes. Entropy (Basel, Switzerland), 21(6), 562. https://doi.org/10.3390/e21060562

[20] Ni, S., Lu, S., Lu, K., & Tan, H. (2021). The effects of parental involvement in parent–child reading for migrant and urban families: A comparative mixed-methods study. Children and Youth Services Review, 123, 105941. https://doi.org/10.1016/j.childyouth.2021.105941

[21] Papadakis, S., & Kalogiannakis, M. (2017). Mobile educational applications for children: What educators and parents need to know. International Journal of Mobile Learning and Organisation (IJMLO), 11(3). https://doi.org/10.1504/ijmlo.2017.085338

[22] Price, J., & Kalil, A. (2019). The effect of mother–child reading time on children's reading skills: Evidence from natural within-family variation. Child Development, 90(6), e688-e702. https://doi.org/10.1111/cdev.13137

[23] Reul, C., Springmann, U., Wick, C., & Puppe, F. (2018). Improving OCR Accuracy on Early Printed Books by combining Pretraining, Voting, and Active Learning. Journal for Language Technology and Computational Linguistics, 33(1), 3–24. https://doi.org/10.21248/jlcl.33.2018.216

[24] Singh, A. (2021). An introduction to experimental and exploratory research. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.3789360

[25] Xie, Q., Chan, C. H., Ji, Q., & Chan, C. L. (2018). Psychosocial effects of parent-child book reading interventions: A meta-analysis. Pediatrics, 141(4). https://doi.org/10.1542/peds.2017-2675

[26] Zhang, M., Zhou, Y., Zhao, L., & Li, H. (2021). Transfer Learning From Speech Synthesis to Voice

Conversion With Non-Parallel Training Data. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29, 1290-1302. https://doi.org/10.1109/TASLP.2021.3066047

[27] Zivan, M., & Horowitz-Kraus, T. (2020). Parent–child joint reading is related to an increased fixation time on print during storytelling among preschool children. Brain and Cognition, 143. https://doi.org/10.1016/j.bandc.2020.105596