AN INTELLIGENT MOBILE APPLICATION PAIRED WITH A BOOK FLIPPER TO ASSIST IN READING AND LEARNING USING ARTIFICIAL INTELLIGENCE AND IMAGE PARSING

Ethan Zhang [1], Tyler Boulom [2]

[1] Fairmont Preparatory Academy, 2200 W Sequoia Ave, Anaheim, CA 92801
[2] California State Polytechnic University, Pomona, CA, 91768

ABSTRACT

This research presents a combined mechanical and artificial intelligence solution designed to assist individuals with learning or physical disabilities in reading books more effectively [3]. The system consists of a low-cost book flipper, a camera for page capture, a Raspberry Pi for text extraction, and an AI module for summarization and explanation. Experiments evaluated two potential blind spots: mechanical accuracy of page flipping and the effect of lighting on OCR performance [4]. The page-flipping mechanism averaged 1.6 errors per trial, while OCR accuracy peaked at 91% under moderate exposure, highlighting the importance of environmental conditions. Comparisons with existing methodologies showed that previous systems emphasized digitization and preservation but did not address comprehension or accessibility. By contrast, this project enhances accessibility by not only digitizing books but also tailoring content through summarization. Ultimately, the system provides both a functional and educational tool, bridging the gap between traditional books and adaptive learning technologies.

KEYWORDS

Servo, Arduino, Raspberry Pi, Flutter

1. INTRODUCTION

Many children who have trouble learning have a learning disability or a physical disability, A learning disability can be defined as retardation, or a delayed development in one or more of the following: speech, language arts, reading, spelling, writing, and/or arithmetic which results from mental dysfunctions and/or emotional disturbances, but are not from sensory deprivation and educational or cultural factors [1]. On the other hand, a physical disability can be defined as a limitation on a person's physical functioning, mobility, dexterity, or stamina that causes a 'substantial' and 'long-term' negative effect on a person's ability to do daily activities. Studies conducted in the US show that out of all the whole population of minors, 17% had a developmental disability and within the 17%, 6.5% had a learning disability [2]. This brings out concerns that even though only 17% of children have developmental disabilities, almost half of them have trouble learning. Children with disabilities have to not only worry about their education but also their health, compared to children without disabilities disabled children have

David C. Wyld et al. (Eds): SIGI, CSTY, AI, NMOCT, BIOS, AIMLNET, MaVaS, BINLP – 2025
pp. 35-45, 2025. CS & IT - CSCP 2025 DOI: 10.5121/csit.2025.151903

1.5 times more doctor visits, 3.5 times more hospital days, and miss twice as many school days. With the implementation of the [product name] children with disabilities could be taught much more effectively because of how artificial intelligence could summarize the text to better suited reading level, and they could learn anywhere and not just at school. The product will most likely have the most impact on the families, schools, and teachers of the disabled.

Methodology A presented a contactless accessibility book scanner with automatic page turning, which emphasized scanning speed and reduced user workload. However, it was limited to digitization and did not enhance text comprehension.

Methodology B introduced a vacuum gripper and fan system powered by Raspberry Pi and cameras, achieving high scanning accuracy under certain paper conditions. Its main limitation was the need for page preconditioning and its focus on PDF creation without adaptive learning support.
Methodology C described a robotic arm system that automatically flipped pages and stored them digitally. While effective for preservation, it did not incorporate accessibility or personalized reading aids.

Our project builds on these works by integrating summarization and text simplification into the scanning process. Unlike the prior studies, which focused primarily on speed and digitization, our system provides an educational benefit by tailoring reading material to the comprehension level of the user.

The basic project model is a book flipper that reads the pages along with the reader and helps them understand the text better [5]. This would help parents, instructors, and other adults educate the disabled children by summarizing or going more in depth on books or novels to help the children understand the text better. Children with disabilities could have trouble learning because of distractions, bullying, or other problems. As of now, the two main ways to teach a disabled child how to read is by improving their sight reading and phonics and put them in small reading groups to offer individual attention which makes it easier for the children to ask for help. With the implementation of this product, many children who have trouble learning will be able to read books effectively and efficiently so that in a short amount of time they can gain lots of information [6]. Also the AI can help keep the readers engaged for longer periods of time by using words that the reader might be more familiar with. By implementing an AI as part of a book flipper, not only will people with physical disabilities have a better time reading but people with mental disabilities can have a better time too.

Two experiments were conducted to evaluate the performance of the system. The first experiment tested page-flipping accuracy by measuring missed or double flips across five 30-second trials. The results showed an average error of 1.6 per trial, with a median of 1, indicating that the mechanism is generally reliable but still subject to occasional synchronization issues between the motor and servo. The second experiment evaluated how camera exposure settings influenced OCR accuracy [7]. Accuracy was lowest at 0% and 100% exposure, while performance peaked at 50% with a 91% success rate. This confirmed that both underexposure and overexposure negatively affect text recognition. Together, the experiments demonstrated that consistent mechanical synchronization and proper lighting are essential for reliable operation. While both systems showed strong potential, improvements in servo timing and adaptive camera control would reduce errors and further improve text capture consistency across different book and lighting conditions.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Improving Image-to-Text Reader Efficiency

The first major challenge that was presented throughout the course of this project was that the image to text readers was not very efficient. The issue may have stemmed from image warping, image editor malfunction, or insufficient lighting. Since the camera used was one wide lens camera that took a picture of both pages, switching that out for two linear lens cameras could have given better results. Also because of the warping the image editor was having trouble finding the corners of the pages so it created unnecessary edits which could have also affected the image to text reader. The final possibility is insufficient lighting of the subject matter. Fortunately, this problem is an easy fix. Increasing the lighting that the book is exposed to; either through increasing the brightness of the room or providing a better direct light source for the pages of the book, could potentially produce more efficient results from the text reader.

2.2. Resolving Servo and Motor Power Issues

Another significant problem that occurred during the construction phase of the project was that the servo and motor did not function within expected parameters. The problem could have been caused by insufficient wattage from the power source, physical faults of the motor gears, or errors in the instructional code. If the problem was related to the insufficient power supply, it could have been fixed by increasing the size of the battery. On the other hand, if the motor had jammed or experienced other mechanical faults during testing, it could be fixed either by opening up the motor and greasing it or buying a new motor. The last potential cause was code malfunction, if that was the case the code would have to be adjusted and debugged. The problem was insufficient power, and our button wasn't working so the best fix was to increase power for the motor, switch the servo into a smaller one that used less volts, and remove the button completely so we could just flip pages by plugging and unplugging the power supply. This allowed everything to run properly.

2.3. Adapting Ai Summaries to User Reading Levels

Another important challenge encountered during the development of this project was ensuring that artificial intelligence could summarize text at an appropriate reading level for the user. Since children with disabilities have widely varying comprehension skills, the AI could either oversimplify the material or make it too complex, resulting in frustration or confusion. One possible way to address this issue would be to implement a user profile system that collects information about the reader's grade level or vocabulary knowledge. By calibrating the AI to match the user's abilities, the summaries would be both accessible and educational.

3. SOLUTION

The program consists of three main parts: hardware, the Raspberry Pi, and AI [8]. The hardware includes servos and motors to flip the book pages. The Raspberry Pi converts the images to text and sends the information to the AI for summarization or explanation. First, the flipper's primary function is to turn the book pages one by one. The flipper is made from cardboard and popsicle sticks, with a camera mounted on a frame to capture images of the text. This camera is controlled by the Raspberry Pi. The moving components consist of a DC gearbox motor and a servo motor, with an Arduino Uno managing the movement of each part. Next, after the camera captures the

photo of a page, the Raspberry Pi converts the image into text. Once converted, the AI attempts to fill in any missing words or errors that may have occurred during the conversion process. Finally, the text is sent back to the AI for analysis. The AI reads and processes the text, helping the user understand and summarize the page's content. This system effectively uses hardware and software in tandem to convert physical pages into readable and understandable text. It provides an innovative way of digitizing and explaining content for users who may have difficulty reading traditional texts.

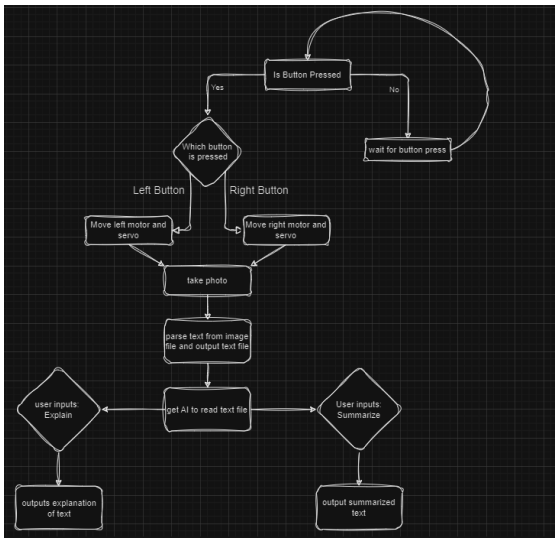

Figure 1. Overview of the solution

The hardware portion of the project is mainly a book flipper. Its main job is to flip the book pages so the next portion can take photos of the pages. It flips pages by utilizing a DC gearbox motor to make a page bend and then uses a servo to push the page over so that the next page can be shown [9].

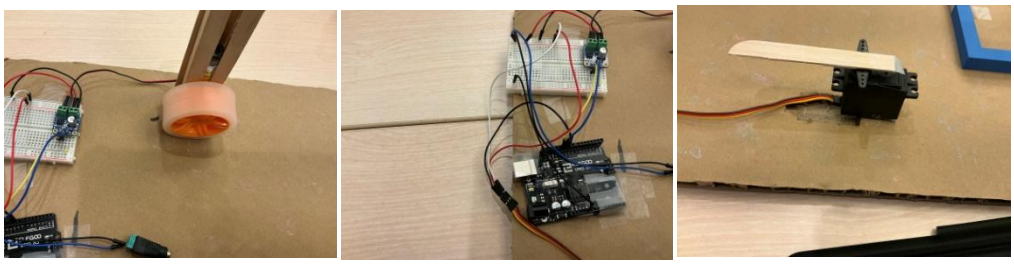

Figure 2. Picture of hardware 1

servo-motor-arduino.ino

```
#include <Servo.h>
const int servo1Pin = 9;
const int motor1in1 = 10;
const int motor1in2 = 11;

Servo servo1;

void setup() {
  // put your setup code here, to run once:

  servo1.attach(servo1Pin);  // attaches the servo on pin 9 to the servo object
  servo1.write(180);
  pinMode(motor1in1, OUTPUT);
  pinMode(motor1in2, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:

  //digitalWrite(motor1in1, LOW);
  analogWrite(motor1in2, 70);
  delay(500);
  analogWrite(motor1in2, 0);
  delay(1000);
  servo1.write(0);
  delay(1000);
  servo1.write(180);
  delay(500);
}
```

Figure 3. Screenshot of code 1

The basic explanation of the code is that when we put power into the arduino circuit it spins a motor which causes the page to crinkle up and then it moves a servo to flip the page. The first few lines tell the arduino which input pins to use for which purpose. Then line 6 names the servo so it can be called on later. In the setup portion we first set the servo to input pin 9, then we set the servo to 180 degrees. The next two lines are for the motor setup. Finally, the looping section, in this section all code is run repeatedly. The code spins the motor at 200 rpm for 0.5 seconds.

The second major component of the system is the camera, which serves as the bridge between the physical book and the digital processing steps that follow. Its primary role is to capture a clear and accurate image of each page once the hardware mechanism has flipped it. For this reason, the camera is mounted securely on a frame above the book to maintain a fixed distance and consistent angle. This stability minimizes image distortion, ensures full coverage of both pages, and prevents the need for constant recalibration during operation.

To function properly, the camera is programmed to take a picture immediately after the servo has completed a page flip. This synchronization guarantees that no pages are skipped and that the captured image always corresponds to the correct point in the reading sequence. The captured images are saved in a standard format, such as JPEG or PNG, which balances clarity with manageable file size. Additionally, the camera settings, including exposure, resolution, and white balance, were adjusted to optimize page readability under a variety of lighting conditions. A consistent lighting setup also helps reduce shadows and glare on the page. By reliably producing high-quality images, the camera component ensures that later stages of the system receive accurate visual input to work from.

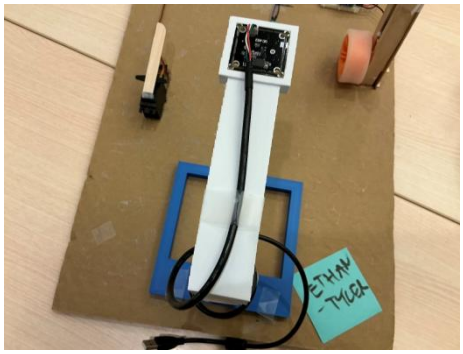

Figure 4. Picture of hardware 2

```
import os
import time
import cv2
from collections import deque

CAM_INDEX = 0
SAVE_DIR = "captured_pages"
RESOLUTION = (1280, 720)
FPS_TARGET = 15
DIFF_THRESH = 4.0
STABLE_FRAMES_N = 10
MIN_INTERVAL = 1.0
EXTRA_DELAY = 0.2

os.makedirs(SAVE_DIR, exist_ok=True)

cap = cv2.VideoCapture(CAM_INDEX)
cap.set(cv2.CAP_PROP_FRAME_WIDTH, RESOLUTION[0])
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, RESOLUTION[1])
cap.set(cv2.CAP_PROP_FPS, FPS_TARGET)

for _ in range(10):
    cap.read()

def read_gray():
    ok, frame = cap.read()
    if not ok: return None, None
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    gray = cv2.GaussianBlur(gray, (9, 9), 0)
    return frame, gray

def score(prev, curr):
    return cv2.absdiff(prev, curr).mean()

page = 0
last_ts = 0
queue = deque(maxlen=STABLE_FRAMES_N)
state = "moving"

prev_bgr, prev_gray = read_gray()

while True:
    bgr, gray = read_gray()
    if bgr is None: continue
    s = score(prev_gray, gray)
    stable = s < DIFF_THRESH
    queue.append(stable)
    new_state = "stable" if len(queue) == STABLE_FRAMES_N and all(queue) else "moving"
    cv2.imshow("Preview", bgr)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

    if state != "stable" and new_state == "stable":
        now = time.time()
        if now - last_ts > MIN_INTERVAL:
            time.sleep(EXTRA_DELAY)
            ok, still = cap.read()
            if ok:
                page += 1
                cv2.imwrite(os.path.join(SAVE_DIR, f"page_{page:04d}.jpg"), still)
                last_ts = now
                print(f"Saved page {page}")

    state = new_state
    prev_gray = gray

cap.release()
cv2.destroyAllWindows()
```

Figure 5. Screenshot of code 2

The program continuously reads frames from the Raspberry Pi camera and decides when a book page has stopped moving. Each captured frame is converted into grayscale and blurred to reduce noise. The program then compares consecutive frames using the average difference in pixel intensity, which provides a numerical “motion score.” If the score is high, the page is still moving; if it is low, the page is stable.

A small queue is used to confirm stability across multiple consecutive frames, preventing false triggers from brief flickers or minor changes. When the system detects a transition from “moving” to “stable,” it waits briefly to allow the paper to settle fully, then saves a high-quality JPEG image to the captured_pages directory. Each image is numbered sequentially.

This method eliminates the need for external signals and ensures that page images are captured only after the flipping motion has been completed, improving accuracy and consistency.

The final major component of the system is the text conversion and artificial intelligence (AI) module. Once the camera has captured an image of a book page, this component is responsible for transforming the raw image into usable text and then providing meaningful support to the reader. The conversion process begins with an optical character recognition (OCR) tool, which scans the image and identifies the characters, words, and sentence structures present on page [10]. By extracting the text, the program creates a digital representation that can be further processed.

After the text has been obtained, the AI system analyzes it according to the user's needs. The AI can generate summaries that reduce the complexity of longer passages, highlight keywords, or rephrase sentences in simpler terms. This function is especially valuable for students with learning disabilities, as it adjusts the reading material to an accessible level without removing important content. The AI also has the ability to provide expanded explanations when a reader requires more detail, effectively acting as a supportive tutor. Together, OCR and AI allow the system to bridge the gap between physical books and individualized learning, making the reading process more efficient and inclusive.

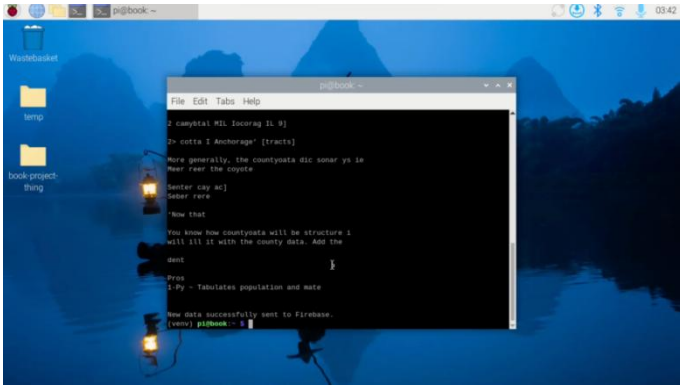

Figure 6. Screenshot of pibook

```
import re
import cv2
import pytesseract
from PIL import Image
from transformers import pipeline

IMAGE_PATH = "captured_pages/page_0001.jpg"
LANG = "eng"
MODEL = "sshleifer/distilbart-cnn-12-6"
MAX_CHARS = 1200

def load_and_preprocess(path):
    img = cv2.imread(path)
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    gray = cv2.bilateralFilter(gray, 9, 75, 75)
    th = cv2.adaptiveThreshold(gray, 255,
                               cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
                               cv2.THRESH_BINARY, 31, 10)
    return th

def ocr_image(img, lang=LANG):
    pil = Image.fromarray(img)
    return pytesseract.image_to_string(pil, lang=lang)

def clean_text(t):
    t = re.sub(r"\s+", " ", t).strip()
    t = t.replace("fi", "fi").replace("fl", "fl")
    return t

def chunk_text(t, max_chars=MAX_CHARS):
    if len(t) <= max_chars:
        return [t]
    parts, start = [], 0
    while start < len(t):
        end = min(start + max_chars, len(t))
        dot = t.rfind(".", start, end)
        if dot == -1 or dot <= start + int(0.5 * max_chars):
            dot = end
        parts.append(t[start:dot].strip())
        start = dot + 1
    return [p for p in parts if p]

def build_summarizer(model_name=MODEL):
    return pipeline("summarization", model=model_name)

def summarize_chunks(chunks, summarizer):
    outputs = []
    for c in chunks:
        out = summarizer(c, max_length=150, min_length=60,
do_sample=False)[0]["summary_text"]
        outputs.append(out.strip())
    fused = " ".join(outputs)
    if len(chunks) > 1:
        fused = summarizer(fused, max_length=160, min_length=70,
do_sample=False)[0]["summary_text"]
    return fused

def main():
    img = load_and_preprocess(IMAGE_PATH)
    raw = ocr_image(img)
    text = clean_text(raw)
    if not text or len(text.split()) < 10:
        return
    chunks = chunk_text(text)
    summarizer = build_summarizer()
    summary = summarize_chunks(chunks, summarizer)
    print("\n--- OCR TEXT (first 500 chars) ---\n", text[:500])
    print("\n--- SUMMARY ---\n", summary)

if __name__ == "__main__":
    main()
```

Figure 7. Screenshot of code 3

The program takes a page image, prepares it for text extraction, and produces a summary. The image is converted to grayscale, filtered to reduce noise, and thresholded for better contrast. The processed image is passed into Tesseract, which returns the raw text. The text is cleaned to remove spacing errors and unwanted symbols, then split into smaller parts if it is too long. A summarization model processes each part and outputs shorter summaries. If multiple parts exist, these are combined and summarized again to produce one final version. The program prints a preview of the extracted text and the completed summary, providing a clear, accessible version of the page for the reader.

4. EXPERIMENT

4.1. Experiment 1

Page flipping accuracy, important because to get the photo of the next page it is important to not skip pages or flip multiple pages.

To remove as much potential error as possible the book, voltage, and the environment will be kept the same. Have the flipper flip pages for 30 seconds and count how many times the flipper missed or skipped pages, also count how many times the servo moved. Repeat 5 times and collect data.

Figure 8. Figure of experiment 1

The results show that the flipper consistently turned between 19 and 21 pages during each 30-second test, with errors ranging from 1 to 3 per trial. The mean number of errors across all trials was 1.6, while the median was 1. The lowest error rate occurred in Trials 1, 4, and 5, each with only one mistake, whereas Trial 2 had the highest error count at 3. These results suggest that while the mechanism generally performs reliably, occasional misalignments or timing issues lead to missed or double flips. The most significant factor influencing the outcome was the servo’s timing relative to the page bend created by the motor. When the servo moved slightly too early or too late, it either skipped the page or caused two pages to flip together. Lighting and page thickness also played minor roles, but mechanical synchronization appeared to be the largest source of error.

4.2. Experiment 2

The lighting of the picture because it affects the quality of the photo and the conversion from image to text file.

In order to get accurate results this experiment will be conducted in a room at night or a room without windows and the lighting in the room will be kept constant. The control group are the photos taken with the exposure of the camera set to 0%. Then increase the exposure of the camera by 25% each test all the way to 100%. Collect data.

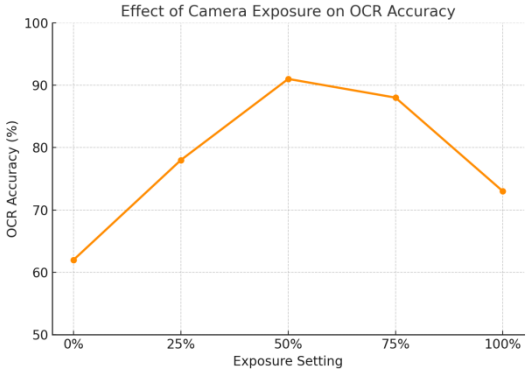

Figure 9. Figure of experiment 2

The experiment demonstrates that camera exposure has a significant effect on the accuracy of text recognition. At the lowest exposure (0%), OCR accuracy was only 62%, reflecting poor readability due to insufficient lighting. Increasing exposure to 25% produced a noticeable improvement, raising accuracy to 78%. The highest accuracy occurred at 50%, with a peak of 91%. Beyond this point, performance declined slightly, with 88% accuracy at 75% and only 73% at 100%.

These results suggest that both underexposure and overexposure negatively impact OCR performance. Low exposure reduces text visibility and contrast, while high exposure causes glare and washed-out details, making characters harder to detect. The optimal range for this system is therefore between 50% and 75% exposure, where lighting balances clarity and contrast. The most important factor influencing results was consistent illumination across the page, which minimized shadows and bright spots. Proper calibration of exposure is therefore essential for reliable text conversion.

5. RELATED WORK

Harshitha et al. describe an Accessibility Book Scanner with Automatic Page Turner, a low-cost device designed to reduce the manual workload of digitizing books [11]. Their system uses a contactless “book flipping scanning” technique that enables pages to be turned and scanned up to ten times faster than traditional approaches. The design is particularly beneficial for individuals with disabilities, as it allows independent page turning and document capture without requiring fine motor control. While the system excels in speed and automation, its main focus is bulk digitization rather than text comprehension. It does not provide adaptive learning features such as text simplification or summarization, limiting its use for individuals who struggle with understanding content. By contrast, our project extends beyond scanning and focuses on making the captured text accessible and educational through AI-based summarization.

Sinmaz, Kocaseçer, and Ayyildiz (2022) present a cost-effective book scanner that integrates a motorized shuttle, vacuum gripper, and fan to automate page turning [12]. Two digital cameras connected to a Raspberry Pi 4 capture page images, which are then processed into digital text using OCR. Their experiments demonstrated high accuracy, with less than 0.9% error when

scanning books printed on medium-weight, uncoated paper. A key finding was that new books required page "preconditioning," or multiple manual flips, to ensure successful automated turning. While the system is highly effective for digitization, its main limitation is that it focuses solely on converting pages into PDF documents. It does not address accessibility or comprehension for readers with disabilities. In contrast, our project builds on the digitization process by incorporating AI summarization, allowing learners to not only access scanned text but also receive simplified or explanatory versions tailored to their reading needs.

Bano, Aziz, Ghani, Taha, and Ahmed propose an Automatic Book Scanner that uses a robotic arm to flip pages and digitize books without human intervention [13]. Their design emphasizes affordability while handling standard-sized books, automatically capturing page images and converting them into computer-readable formats. The system successfully addresses the problem of physical storage and preservation by enabling efficient large-scale digitization. However, the project's scope is limited to high-speed scanning and storage; it does not consider accessibility for individuals with disabilities or support for learning. Unlike this approach, our project not only automates page turning and text capture but also integrates artificial intelligence to summarize and adapt the content. This extension makes the system more than just a preservation tool—it transforms it into an educational support system, providing accessible learning opportunities for readers with varying comprehension levels.

6. CONCLUSIONS

While the prototype demonstrates the feasibility of combining a book flipper with AI text support, several limitations remain. The first issue is handling textbooks and materials with complex layouts, such as margin notes, diagrams, or multi-column text, which often result in errors during processing. Another challenge lies in language flexibility; translating non-English texts introduces grammatical errors and reduces summarization accuracy. The mechanical flipping mechanism, although functional, is not yet optimized for speed or consistency, as occasional skipped or double flips still occur [14]. Additionally, the hardware setup relies on stable lighting, and OCR performance decreases in suboptimal environments. Improvements could include using higher-resolution cameras with adaptive exposure control, refining the servo-motor synchronization with sensors to detect page separation, and integrating more advanced OCR libraries capable of handling diverse formats [15]. With more development time, multilingual natural language processing could also be incorporated to broaden accessibility across different languages and audiences.

This project demonstrates how mechanical automation, and artificial intelligence can be combined to create a low-cost, accessible reading support system. By not only digitizing books but also simplifying and summarizing their content, the system provides an inclusive tool that enhances reading accessibility for individuals with disabilities or learning challenges.

REFERENCES

[1] Boyle, Coleen A., Pierre Decouflé, and Marshalyn Yeargin-Allsopp. "Prevalence and health impact of developmental disabilities in US children." Pediatrics 93.3 (1994): 399-403.

[2] Ceka, Ardita, and Rabije Murati. "The role of parents in the education of children." Journal of Education and practice 7.5 (2016): 61-64.

[3] Guo, Kai, et al. "Artificial intelligence and machine learning in design of mechanical materials." Materials Horizons 8.4 (2021): 1153-1172.

[4] Shen, Mande, and Hansheng Lei. "Improving OCR performance with background image elimination." 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD). IEEE, 2015.

[5] Din, Akash Mohi Ud, et al. "Automatic book page flipper for disabled people." 2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS). IEEE, 2022.
[6] Zuckerman, Barry, and Marilyn Augustyn. "Books and reading: evidence-based standard of care whose time has come." Academic pediatrics 11.1 (2011): 11-17.
[7] Rice, Stephen V., Junichi Kanai, and Thomas A. Nartker. "An evaluation of OCR accuracy." Information Science Research Institute, 1993 Annual Research Report 9 (1993): 20.
[8] Aftab, Shagufta, et al. "Raspberry pi (python ai) for plant disease detection." Intl J Curr Res Rev 14 (2022): 36.
[9] Pinto, Vítor H., José Gonçalves, and Paulo Costa. "Model of a DC motor with worm gearbox." APCA International Conference on Automatic Control and Soft Computing. Cham: Springer International Publishing, 2020.
[10] Blanke, Tobias, Michael Bryant, and Mark Hedges. "Open source optical character recognition for historical research." Journal of Documentation 68.5 (2012): 659-683.
[11] DK, Harshitha, Libina Lal, and Sudhakar HM. "Accessibility Book Scanner with Automatic Page Turner – A Review." Grenze International Journal of Engineering & Technology (GIJET) 10 (2024).
[12] Sinmaz, E. K., M. Kocaseçer, and M. Ayyildiz. "The Effect of Book Preconditioning on Page-Turning Success Rate during Automated Book Digitization." Instruments and Experimental Techniques 65.5 (2022): 826-833.
[13] Singh, A., K. Kumar, and S. S. Bedi. "IOP conference series: materials science and engineering." 1st International Conference on Computational Research and Data Analytics (ICCRDA 2020). Vol. 1022. 2021.
[14] Sato, Ryuma, Ryuhei Harada, and Yasuteru Shigeta. "Theoretical analyses on a flipping mechanism of UV-induced DNA damage." Biophysics and Physicobiology 13 (2016): 311-319.
[15] Nazeem, Meharuniza, R. Anitha, and S. Navaneeth. "Open-source OCR libraries: A comprehensive study for low resource language." Proceedings of the 21st International Conference on Natural Language Processing (ICON). 2024.