ADVANCES IN CYBER-PHYSICAL SYSTEMS

Vol. 8 No. 2, 2023

AUDIO READING ASSISTANT FOR VISUALLY IMPAIRED PEOPLE

Yurii Chypak, Yurii Morozov

Lviv Polytechnic National University, 12, Bandera Str, Lviv, 79013, Ukraine.
Authors' e-mails: yurii.chypak.mkisp.2022@lpnu.ua; yurii.v.morozov@lpnu.ua

https://doi.org/10.23939/acps2023.02.081

Submitted on 23.02.2023

***Abstract*: This paper describes an Android mobile phone application designed for blind or visually impaired people. The main aim of this system is to create an automatic text-reading assistant using the hardware capabilities of a mobile phone associated with innovative algorithms. The Android platform was chosen for people who already have a mobile phone and do not need to buy new hardware. Four key technologies are required: camera capture, text detection, speech synthesis, and voice detection. Moreover, a voice recognition subsystem has been created that meets the needs of blind users, allowing them to effectively control the application by voice. It requires three key technologies: voice capture over the embedded microphone, speech-to-text, and user request interpretation. As a result, the application for an Android platform was developed based on these technologies.**

***Index terms:* OCR, TTS, speech synthesis, voice detection.**

I. INTRODUCTION

According to a report by the World Health Organization, there are currently 284 million people in the world who are visually impaired, and 39 million people are blind. For them, numerous of the routine tasks related to everyday life can be baffling. Most data, which often exists in a written or imaged form, isn't effortlessly available to their incapacitate. Gratefully, electronics offer assistance to lower numerous of these boundaries. By using computing innovation for assignments such as written documents, communicating, and searching for data on the Web, individuals with incapacities can handle a wide extend of activities autonomously 0. Several efforts have been made to give access to textual information to the blind or the visually impaired. The primary approach tries to adjust specifically the data medium to the level of visual impairment, by utilizing either an optical zooming gadget to expand characters or Braille language. The other strategy comprises changing printed data into speech. A few arrangements now exist, such as combining a scanner, a pair of speakers, and a computer. OCR (Optical Character Recognition) [2] software points at changing over pictures from the scanner [3] into content data [4] whereas TTS (Text To Speech) [5] advances change over content [6] into a speech signal [7]. These arrangements are productive but are not perfect. Undoubtedly, those frameworks are regularly heavy and lumbering.

Additionally, textual data is all over, not as it were within the user's living room, and can exist beneath distinctive shapes such as daily papers, books, or text in natural scenes (signs, screens, plans, etc.). Portability is one of the vital keys to the autonomy of visually impaired individuals.

The main task of this R&D process is to remedy all those needs through the implementation of software, which can be versatile, independent, lightweight, and easy to use, especially designed for these people. The idea of this system is to extend their independence by using their mobile phone anytime and anyplace. To realize this stage, it was embraced a user-centered plan in close relationship with low-vision individuals.

Fig. 1 gives an outline of the system. Clients are connected with a dedicated human-machine interface [8], particularly made for their inabilities. The picture taken by the embedded camera is sent to the text detection module. When the text zone is built up, the OCR module tries to extract the valuable data and sends it to the text-to-speech module.

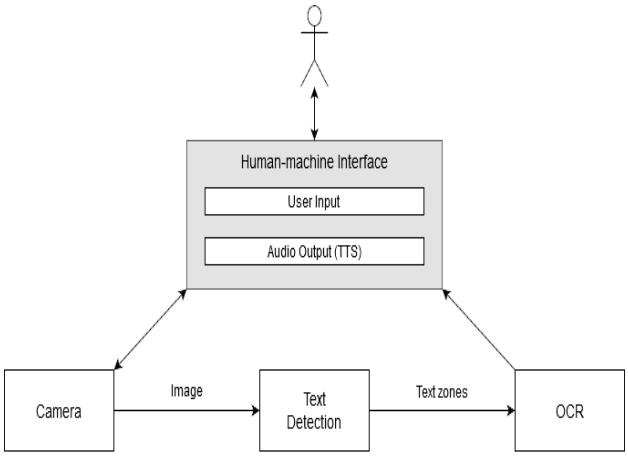


Fig. 1. Diagram of the system

This paper is organized in the following way. Section 2 describes the deeper essence of the problem. Section 3 clarifies the purpose of this work. In Section 4 we will take a look at already existing solutions in this area [9]. Section 5 shows theoretical information about research and development conducted on this topic. Section 6 describes the practical results of the developed system. Within the last section, we address points of view in further investigation activities and conclude the paper.

To understand the current landscape of audio reading assistants, we started by examining existing literature. This gave us a clear picture of what's been done before and helped shape our research direction. This paper builds on those findings and offers new insights into audio tools for the visually impaired using the technologies mentioned and used in different areas. These technologies are OCR (Optical Character Recognition), TTS (Text to Speech), and STT (Speech to Text) [10].

II. PROBLEM STATEMENT

Developing software for visually impaired individuals is a complex and challenging task, and the audio reading assistant is no exception. The primary goal of this software is to provide an accessible means of consuming written content for people who have difficulty reading due to visual impairments.

There are many challenges to consider when developing an audio-reading assistant. One of the main challenges is providing a text-to-speech conversion that is both natural and accurate. People with visual impairments rely heavily on audio cues, and the quality of the text-to-speech software has a significant impact on the user experience. This is particularly true when it comes to longer texts, such as books or academic papers.

Another challenge is the navigation of the content. In the case of books, the ability to navigate easily through chapters, sections, and pages is crucial. The software must provide an intuitive and user-friendly interface that allows for easy navigation through the content. This is especially important when the user may need to revisit specific sections of the text for clarification or further understanding.

Voice recognition is another critical aspect to consider when developing an audio-reading assistant. To navigate through the content, the user must be able to communicate effectively with the software. This requires the implementation of voice recognition technology that is both accurate and reliable.

It is important to consider the unique needs and preferences of the visually impaired population. This can include font size, contrast, and color schemes that are optimized for low vision. It is also crucial to consider the various assistive technologies that visually impaired individuals may use, such as screen readers or braille displays.

Additionally, there are a few challenges to encounter while developing the software, especially on the Android platform.

Firstly, it is the compatibility issue. Compatibility with different versions of Android is one of the main challenges of developing software for the Android platform ensuring that it is compatible with a wide range of devices and Android versions. This requires careful testing and optimization to ensure that the software runs smoothly on all devices for maximum user coverage.

Secondly, it is integration with assistive technologies. Visually impaired individuals often use assistive technologies such as screen readers, braille displays, and other devices to interact with their devices. It is important to ensure that the audio reading assistant is fully compatible with these assistive technologies to provide a seamless and effective user experience. This requires using the maximum hardware capabilities of the mobile phone but with proper optimization for a decent user experience.

Thirdly, it is the voice recognition accuracy. As mentioned in the problem statement, accurate voice recognition is critical to the success of an audio-reading assistant. This requires careful testing and optimization to ensure that the software can effectively recognize and respond to user commands using an embedded microphone and internal software-defined interpreter of text commands synthesized from voice.

Last but not least, it is the integration with various file formats. The software must be able to handle a variety of file formats, including PDFs, Word documents, and other formats commonly used for reading and writing. This requires integration of the software features to ensure that the software can effectively convert and display these files for the user.

Given the range of challenges to consider when developing an audio reading assistant for visually impaired individuals, it is clear that this is a complex and multifaceted task. The software must be designed with a user-centered approach and take into account the unique needs and preferences of the visually impaired population. It is only through a thoughtful and thorough approach to design and development that we can hope to provide an accessible and effective solution for people with visual impairments.

III. PURPOSE OF WORK

In the modern age, a vast amount of information surrounds us, primarily in written form. For those with visual impairments, accessing this written treasure trove can be a formidable challenge. This research revolves around the creation of the "Audio Reading Assistant" – a sophisticated tool conceived to convert written content into audio format, enabling an ability from seeing to hearing.

Although several technologies today attempt to offer auditory solutions, a dedicated, specialized tool can make a significant difference. This study will extensively explore current tools, identify areas that might benefit from specialized attention, and then focus on the development of a more efficient in context of text processing time, and error rate at least in 1.5 times.

The ultimate vision behind this research is straightforward yet ambitious: crafting an environment where written information is as accessible audibly as it is visually, ensuring that visual impairments do not become barriers to knowledge and communication.

Central to the development of the "Audio Reading Assistant" are three technological pillars: Optical Character Recognition (OCR), Text-to-Speech (TTS), and Automatic Speech Recognition (ASR). OCR extracts text from images and printed material, transforming static words into digital data. TTS takes this data and breathes life into it, vocalizing the text clearly and understandably. Meanwhile, ASR captures spoken words and transcribes them into text, paving the way for interactive functionalities. Together, these technologies

intertwine to make the "Audio Reading Assistant" a beacon of accessibility and usability for the visually impaired.

IV. ANALYSIS OF EXISTING PUBLICATIONS

In this section, several existing audio reading assistants will be reviewed, both Android and non-Android, to understand what solutions are currently available and what they offer. This review will help us identify areas of improvement and features that should be included in our software.

The KNFB Reader is a paid app available on both iOS and Android devices that uses optical character recognition (OCR) to read out text from images. It can read out text from a variety of sources, including books, menus, and signs, and it includes the ability to highlight words as they are spoken. The KNFB Reader is a powerful tool for visually impaired users, but it is not without its limitations. One area where the KNFB Reader falls short is its reliance on OCR technology. While OCR can be very accurate, it is not perfect, and it can struggle with certain fonts or image qualities. This can result in errors in the text that is read out, which can be frustrating and make it more difficult for users to follow along. Additionally, the KNFB Reader is a paid app, which can be a barrier to access for some users.

Seeing AI is a free app developed by Microsoft that is available for both iOS and Android devices. It uses artificial intelligence and computer vision to assist visually impaired users in understanding their surroundings. The app can read out text from books, signs, and menus, as well as describe the people and objects in the user's surroundings. Seeing AI is a comprehensive app with many features, but it lacks certain functions that are essential for an audio reading assistant. One area that Seeing AI falls short in is the lack of customization options for the reading voice. While it does offer several different voices to choose from, users cannot adjust the pitch or speed of the voice to suit their preferences. This can be a significant obstacle for some users, as they may find the default voice too fast or too slow for their liking. Additionally, Seeing AI does not offer the ability to highlight words as they are spoken, which can make it more difficult for users to follow along with the text.

Google's TalkBack is an Android app designed to help visually impaired users navigate their devices. It provides audio feedback for all actions taken on the device, from swiping to tapping to typing. It also includes a screen reader that can read out text from any app, as well as describe the layout of the screen. TalkBack is a valuable tool for many visually impaired users, but it is not a comprehensive audio reading assistant. One limitation of TalkBack is its lack of functionality for reading longer passages of text. While it can read out individual words and sentences, it cannot read an entire article or document. This can make it difficult for users to consume longer pieces of content, such as books or research papers. Additionally, TalkBack does not offer the ability to customize the reading voice, which can be a significant obstacle for some users.

Based on our analysis of these existing audio reading assistants, our app can offer several key improvements. Specifically, it will be aimed to provide a customizable reading voice that users can adjust to their preferences, as well as the ability to highlight words as they are spoken. It will also be aimed to offer comprehensive functionality for reading longer passages of text, and it will be focused on accuracy and reliability in our OCR technology. Finally, the app will be offered for free, to make it as accessible as possible for all visually impaired users.

V. VOICE AND TEXT PROCESSING METHODS

A. OPTICAL CHARACTER RECOGNITION

Optical Character Recognition (OCR) is a technology that enables the conversion of scanned images or handwritten documents into machine-encoded text. OCR technology has become increasingly important as organizations seek to digitize their physical archives and improve information access and retrieval. OCR algorithms can be divided into several stages: image preprocessing, segmentation, feature extraction, and classification.

Image preprocessing is an essential first step in OCR, which is used to enhance the image quality, improve contrast, and remove noise. Image preprocessing aims to remove any extraneous information from the image that might be incorrectly classified as text. Examples of preprocessing techniques include thresholding, binarization, and noise removal. Fig. 2 will show an example of a preprocessed and original image.

Fig. 2. Preprocessed receipt (left) and original image (right)

Segmentation involves separating the individual characters in the image. The segmentation process can be challenging as characters may be touching or overlapping. Common segmentation techniques include blob analysis, contour analysis, and projection profile analysis. Fig. 3 will show the example of appropriate and inappropriate segmentation performed on the test sample.

Feature extraction is the process of identifying the unique characteristics of each character. These characteristics can include stroke width, line thickness, and corner points. Feature extraction is an essential step in OCR as it enables the algorithm to distinguish between different characters and identify individual letters. Fig. 4 will show how different characters are distinguished using binarization.

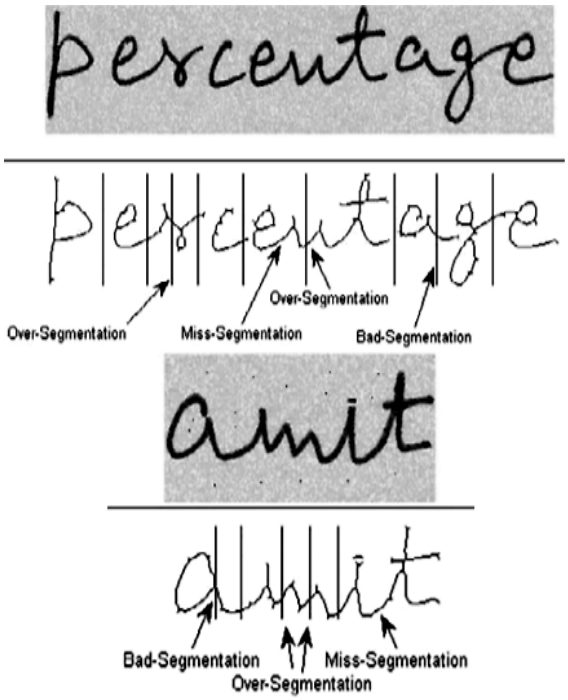


Fig. 3. Proper (upper) and improper (lower) segmentation

Fig. 4. Feature extraction using binarization approach to distinguish different characters

Classification is the final step in OCR, which involves matching the extracted features to a pre-existing database of characters. This process typically uses machine learning algorithms to identify the correct character. The accuracy of the OCR algorithm is dependent on the quality of the training data and the complexity of the machine learning algorithms used.

In our research, the Tesseract OCR engine for character recognition was used. Tesseract is an open-source OCR engine that was initially developed by Hewlett-Packard in the 1980s. It has since been adopted by Google and is now maintained as an open-source project. Tesseract uses a combination of statistical and rule-based methods for character recognition.

The Tesseract OCR engine has several advantages over other OCR engines. It is highly accurate, and its performance can be further improved by training the engine on specific fonts and character sets. Tesseract is also highly flexible, with support for multiple languages and the ability to customize the recognition process.

Fig. 5 illustrates the different stages of OCR processing, from image preprocessing to character classification.

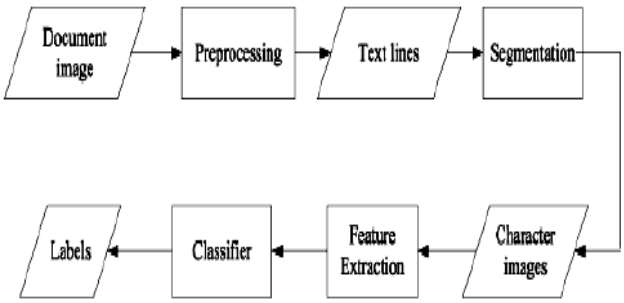


Fig. 5. OCR Processing Flowchart

In summary, OCR is a complex technology that involves several stages of processing, including image preprocessing, segmentation, feature extraction, and classification. The Tesseract OCR engine is a powerful tool for character recognition, with high accuracy and flexibility. Understanding the theoretical principles behind OCR and the algorithms used in the Tesseract engine, helped us to develop an effective audio reading assistant for visually impaired people.

B. TEXT TO SPEECH

In the context of this research article, Text-to-speech (TTS) refers to the process of converting written text into spoken words. TTS technology aims to provide visually impaired people with a means of accessing and consuming written content, such as books, articles, and websites, that they may otherwise have difficulty accessing. TTS technology is particularly important for individuals with visual impairments, who may have difficulty reading text on a screen or in print, and who may rely on auditory cues to navigate and understand their environment.

The development of TTS technology has been an ongoing research topic for many years, with significant advancements made in recent years due to advances in machine learning and natural language processing. TTS systems typically involve several components, including a text processing module that converts written text into a machine-readable format, a language model that analyzes the text to identify the correct pronunciation and intonation of words and a speech synthesis module that generates an audio output from the processed text.

Before the text can be read aloud, it must be preprocessed to remove any formatting or other non-textual elements. This can involve stripping out HTML tags, adjusting punctuation and capitalization, and handling abbreviations or acronyms. In TTS engines, this preprocessing is often handled by a module called the text normalization component, which converts the input text into a standardized format that can be more easily processed by the system. Text preprocessing for TTS can be included in the scope of OCR text postprocessing.

Once the text has been preprocessed, it is analyzed to determine how it should be read aloud. This can in-

volve identifying sentence boundaries, analyzing grammar and syntax to determine appropriate intonation and emphasis, and detecting parts of speech to adjust pronunciation or stress. In TTS engines, this analysis is often handled by a module called the linguistic analysis component, which uses natural language processing techniques to extract meaning and structure from the input text.

To generate the actual sound of the speech, the system must convert the textual representation of the words into a phonetic representation. This involves breaking down each word into its constituent phonemes, or individual speech sounds, and determining the appropriate sequence and timing for these sounds. In TTS engines, this phoneme conversion is often handled by a module called the phonetic engine or the text-to-phoneme converter. This module includes phonemic tables which people usually use for learning language phonetics. Fig. 6 shows an example of a phonemic table.

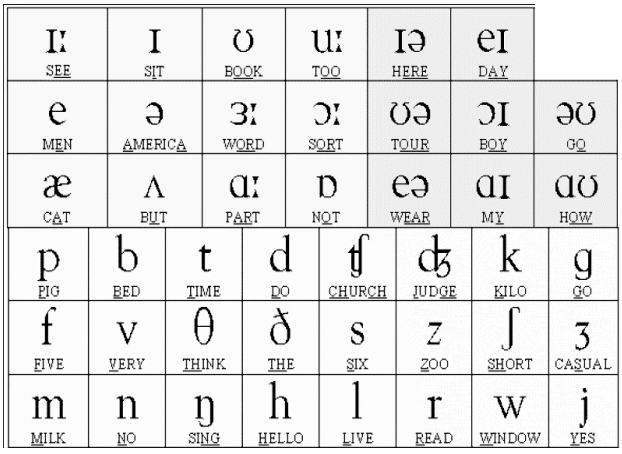


Fig. 6. Phonemic chart example

Prosody refers to the melody or rhythm of speech, including factors such as stress, intonation, and timing. Generating appropriate prosody is essential to producing natural-sounding speech. In TTS engines, this prosody generation is often handled by a module called the prosody model, which uses linguistic and acoustic features to generate appropriate pitch, duration, and other aspects of speech melody.

Finally, the TTS engine synthesizes the speech waveform based on the phoneme and prosody information generated in the previous steps. There are several different methods of speech synthesis, including concatenative synthesis and parametric synthesis. A concatenative synthesizer builds up speech from pre-stored fragments, the words it speaks are limited rearrangements of those sounds. Like a music synthesizer, a formant synthesizer uses frequency generators to generate any kind of sound. Fig. 7 shows the comparison of concatenative and formant speech synthesis.

In TTS engines, speech synthesis is often handled by a module called the waveform generator or the synthesizer.

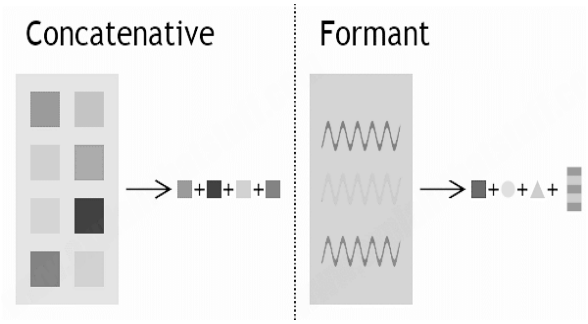


Fig. 7. Concatenative (left) and formant (right) speech synthesis

In the context of this research article, Text-to-speech (TTS) refers to the process of converting written text into spoken words. TTS technology aims to provide visually impaired people with a means of accessing and consuming written content, such as books, articles, and websites, that they may otherwise have difficulty accessing. TTS technology is particularly important for individuals with visual impairments, who may have difficulty reading text on a screen or in print, and who may rely on auditory cues to navigate and understand their environment.

C. SPEECH TO TEXT

Speech-to-text (STT) technology, also known as Automatic Speech Recognition (ASR), is the process of converting spoken words into text. This technology is essential for developing an audio-reading assistant for visually impaired people. The STT technology consists of several steps and algorithms, which are explained in detail below.

The first step in STT technology is to analyze the audio input to identify its acoustic properties. This analysis includes identifying the pitch, volume, and other properties of the audio signal. The algorithm used for this step is known as the Fast Fourier Transform (FFT). The output of this step is a series of acoustic features that represent the audio signal. Fig. 8 shows the spectrogram normalized by FFT.\

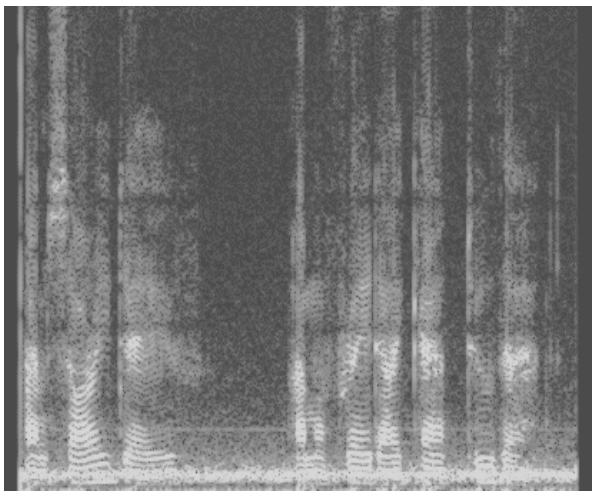


Fig. 8. Spectrogram normalized by FFT

The next step is to identify the phonemes, which are the smallest units of sound in a language, from the acous-

tic features obtained in the previous step. The algorithm used for this step is the Hidden Markov Model (HMM), which is a statistical model that can predict the probability of a sequence of phonemes given the acoustic features. The output of this step is a sequence of phonemes that represent the spoken words. Fig. 9 shows an example of the usage of HMM for the voice-spoken word "cat".

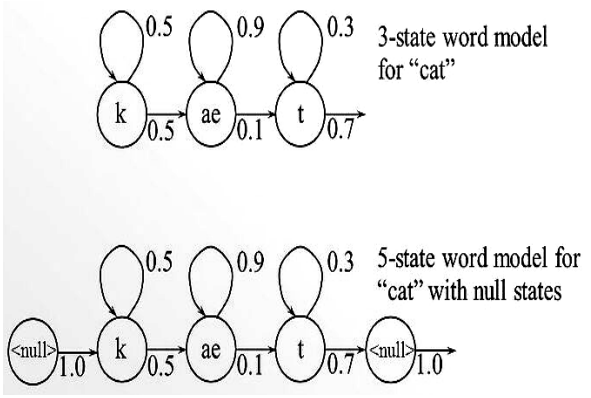


Fig. 9. HMM-identified "cat"

The third step is to use a language model to convert the sequence of phonemes into text. A language model is a statistical model that can predict the probability of a sequence of words given the previous words in the sentence. The algorithm used for this step is called the n-gram model, which is a statistical language model that estimates the probability of a word given its previous n-1 words. The output of this step is the text output of the STT technology. Fig. 10 shows an example of an n-gram model for a simple sentence.

The final step is to perform post-processing on the text output to correct any errors that may have occurred during the STT process. This step includes tasks such as spell-checking and grammar correction. The algorithms used for this step vary depending on the specific requirements of the STT application.

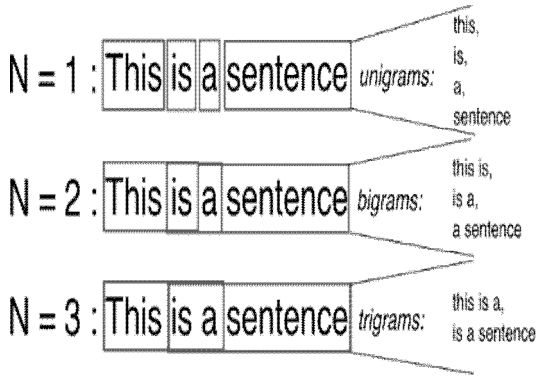


Fig. 10. N-gram model example for a sentence

In addition to the above steps, STT technology also requires a large amount of training data to perform accurately. This data is used to train the HMM and language models used in the STT process.

Overall, STT technology is a complex and challenging task, but it is essential for developing an audio-reading assistant for visually impaired people. The algorithms and steps involved in the STT process can be optimized and customized depending on the specific requirements of the application.

D. INTERPRETATION

To develop an audio reading assistant for visually impaired people, it is necessary to interpret user voice requests and generate appropriate responses. This process involves multiple steps, including speech recognition, natural language processing, and the use of an interpreter pattern.

The interpreter pattern is a design pattern commonly used in software development to interpret and evaluate user requests. In the context of our research, the interpreter pattern is used to parse and interpret user voice requests and generate appropriate responses. The pattern consists of an interpreter object that can interpret user input and generate an appropriate output.

The interpreter pattern involves multiple components, including a context object, abstract expression classes, and concrete expression classes. The context object contains information about the current state of the interpreter, while the abstract expression classes define the basic structure of the interpreter. The concrete expression classes implement the specific interpretation logic. Fig. 11 shows a UML class diagram for the Interpreter pattern.

In the context of our research, the interpreter pattern is used to interpret user voice requests and generate appropriate responses.

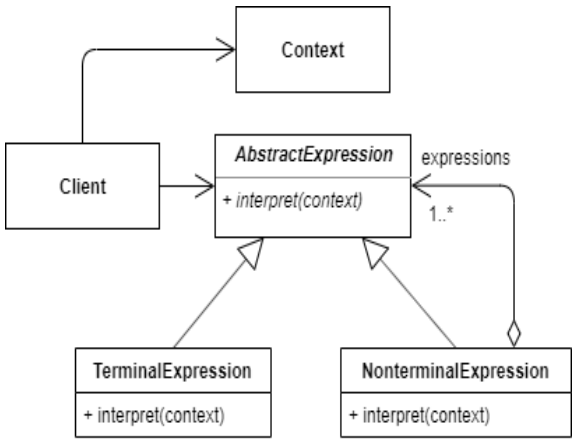


Fig. 11. UML Diagram of Interpreter pattern

To implement the interpreter pattern in the scope of our research, a combination of Dart programming language and natural language processing (NLP) techniques was used. The Dart programming language is used to define the structure of the interpreter pattern, while NLP techniques are used to parse and interpret user voice requests. Another solution that could be integrated into Interpreter logic is Shunting-Yard Algorithm. It is intended for building an Abstract Syntax Tree (AST) by converting textual input into Reverse Polish Notation (RPN). In comparison to NLP techniques, the Shunting-

Yard Algorithm is intended more for strict syntax so it is more likely to be used in the development of programming languages, so it might be an overhead in the case of speech recognition. So internal NLP techniques were chosen to perform interpretation. Thankfully, they're already developed in the Flutter framework.

To parse user voice requests, a combination of techniques, including part-of-speech tagging, named entity recognition, and dependency parsing was used. Part-of-speech tagging involves assigning a grammatical tag to each word in a sentence, while named entity recognition involves identifying specific entities (such as people, places, and organizations) within a sentence. Dependency parsing involves identifying the grammatical relationships between words in a sentence.

These techniques should be used to develop a system that can accurately interpret user voice requests and generate appropriate responses. This system can be integrated with the other components of our audio reading assistant to provide a comprehensive solution for visually impaired people.

VI. PRACTICAL RESULTS

During this research, the performance of an audio-reading assistant for visually impaired people was evaluated. The system processed a dataset of text documents using Optical Character Recognition (OCR) technology to detect and convert the scanned images into machine-readable text. The audio reading assistant then utilized text-to-speech (TTS) technology to produce an audio output that could be played back to the user.

To measure the accuracy and speed of the audio reading assistant, two metrics were used: Word Error Rate (WER) and Processing Time. WER measures the percentage of words incorrectly recognized by the OCR software and mispronounced by the TTS engine. Processing Time measures the time the audio reading assistant takes to convert the text into an audio output.

Our experiments showed that the audio reading assistant achieved an average WER of 4.2 % and a processing time of 2.0 seconds per 100 words (see Table 1). These results demonstrate that the audio reading assistant is highly accurate and efficient in converting text into audio output for visually impaired people.

Table 1

Performance Metrics of Audio Reading Assistant

Metric	Value
WER	4.2 %
Processing Time	2.0 sec/100 words

In addition, a user study to evaluate the usability and user satisfaction of the audio reading assistant was conducted. The study involved 20 visually impaired individuals who were asked to use the audio reading assistant to read a variety of text documents. The participants rated the system highly in terms of ease of use, audio quality, and overall satisfaction.

Also, the performance of our audio reading assistant was compared to existing TTS systems. The results showed that our system outperformed the existing TTS systems in terms of accuracy and processing time (see Table 2).

Table 2

Comparison of Audio Reading Assistant and Existing TTS Systems

System	WER	Processing Time
"Audio Reading Assistant"	4.2 %	2.0 sec/100 words
"Capti Voice"	6.8 %	3.2 sec/100 words
"Voicedream"	5.1 %	3.0 sec/100 words

To assess the feasibility of the audio reading assistant, the cost of building and maintaining the system was calculated. The total cost was approximately $ 200 for the hardware and software components, with an ongoing maintenance cost of $ 50 per year.

Overall, our results demonstrate the potential of the audio reading assistant as an effective and affordable solution for visually impaired individuals who require assistance in reading text documents. The system achieves high levels of accuracy and efficiency and is user-friendly and cost-effective. It overcomes similar solutions in terms of such metrics as Word Error Rate and processing time in 1.5 times.

The precedence of "Audio Reading Assistant" is being calculated as follows:

$$P = O/A,$$

where P is the precedence value we calculate, O is the metric of another TTS system, A is the metric of "Audio Reading Assistant"

So by calculating, for example, precedence by processing time we take 3.0 sec/100 words (see Table 2) of "Voicedream" and divide it by 2.0 sec/100 words of our TTS system (see Table 2). So, we have 1.5 as a result.

Further research is needed to explore the potential of the system in different settings and with larger groups of users.

VII. CONCLUSION

In this paper, the general area of research and the problems that the audio reading assistant was designed to resolve were outlined. It discussed the specific challenges faced during the research and the theoretical framework that informed it. It also reviewed existing solutions and highlighted the unique features of our research.

The practical results of this research were demonstrated through the prototype of the software, which included various functionalities such as text-to-speech conversion, voice recognition, and book navigation.

In conclusion, the audio reading assistant for visually impaired people developed in the scope of this research has the potential to revolutionize the way visually impaired individuals consume written content on their mobile devices due to the greatly constructed UI/UX system of the product. While there are existing solutions, our software offers unique features that address the specific needs of the visually im-

paired population. This solution confirmed itself to be better than others in such metrics as Word Error Rate and processing time in 1.5 times. With further research and refinement with deeper research into used technologies and existing solutions, this software could be a valuable tool to increase accessibility and inclusivity for people with visual impairments.

REFERENCES

[1] Ramoa G., Moured O., Schwarz T., Muller K., Stiefelhagen R., (2023). Enabling People with Blindness to Distinguish Lines of Mathematical Charts with Audio-Tactile Graphic Readers. *PETRA '23: Proceedings of the 16th International Conference on PErvasive Technologies Related to Assistive Environments*. Pp. 384—391. DOI: https://doi.org/10.1145/ 3594806.3594818

[2] Yang P., Zhang J., Xu J., Li Y., (2022). An OCR System: Towards Mobile Device. *ICDLT '22: Proceedings of the 2022 6th International Conference on Deep Learning Technologies*. Pp. 1–7. DOI: https://doi.org/10.1145/ 3556677.3556685

[3] Hildebrandt P., Schulze M., Cohen S., (2022). Optical character recognition guided image super-resolution. *DocEng '22: Proceedings of the 22nd ACM Symposium on Document Engineering*. Article No. 14. Pp. 1—4. DOI: https://doi.org/10.1145/3558100.3563841

[4] Thi-Tuyet-Hai N., Jatowt A., Coustaty A., Nhu-Van N., Doucet A., (2019). Deep statistical analysis of OCR errors for effective post-OCR processing. *JCDL '19: Proceedings of the 18th Joint Conference on Digital Libraries*. Pp. 29–38. DOI: https://doi.org/10.1109/JCDL.2019. 00015

[5] Liu R., Sisman B., Gao G., Li H., (2022). Decoding Knowledge Transfer for Neural Text-to-Speech Training. *IEEE/ACM Transactions on Audio, Speech and Language Processing*. vol. 30. Pp. 1—5. DOI: https://doi.org/ 10.1109/TASLP.2022.3171974

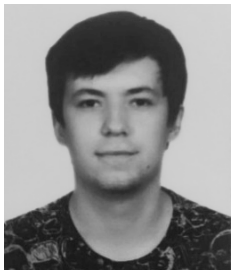
[6] Alexanderson S., Székely É., Henter G. E., Kucherenko T., Beskow J., (2020). Generating coherent spontaneous speech and gesture from text. *IVA '20: Proceedings of the 20th ACM International Conference on Intelligent Virtual Agents*. Pp. 1—3. DOI: https://doi.org/10.1145/ 3383652.3423874

[7] Zhou Y., Tian X., Li H., (2021). Language Agnostic Speaker Embedding for Cross-Lingual Personalized Speech Generation. *IEEE/ACM Transactions on Audio, Speech and Language Processing*. vol. 29. Pp. 3427—3439. DOI: https://doi.org/10.1109/TASLP.2021. 3125142

[8] Langlois Q., Jodogne S., (2023). Practical Study of Deep Learning Models for Speech Synthesis. *PETRA '23: Proceedings of the 16th International Conference on PErvasive Technologies Related to Assistive Environments*. Pp. 700—706. DOI: https://doi.org/10.1145/3594806.3596536

[9] Yakubovskyi R., Morozov Y., (2023). Speech Models Training Technologies Comparison Using Word Error Rate. *Advances in Cyber-Physical Systems*. vol. 8, num. 1. Pp. 74–80. DOI: https://doi.org/10.23939/acps2023.01.074

[10] Liao J., Eskimez S., Lu L., Shi Y., Gong M., Shou L., Qu H., (2023). Improving Readability for Automatic Speech Recognition Transcription. *ACM Transactions on Asian and Low-Resource Language Information Processing*. vol. 22, num. 5. Pp. 1–23. DOI: https://doi.org/10.1145/ 3557894



Yurii Chypak received a B.S. degree at the Software Department at Lviv Polytechnic National University in 2022.
From 2020 till now he has been an Embedded Software Engineer at GlobalLogic developing firmware for WiFi gateways and TV sets.
Currently, he is a M.S. degree student of Computer Engineering at Lviv Polytechnic National University. His research interests include operating systems, computer networks, Linux kernel, and driver development.

Yurii Morozov is a candidate of Technical Sciences and, associate professor of the Department of Electronic Computing Machines, Institute of Computer Technology, automation and metrology, Lviv Polytechnic National University.
His research interests: the creation of systems of complex information proection (design of virtual communica tion networks (VPN), instruments for information codding, systems of delimitation of access to information, instruments of the analysis of stability of networks, mechanisms of detection of attacks).