

Braille Converter

Sindhiya Sengole Raj

A project report submitted in conformity with the requirements
for the degree of Master's of Science in Information Technology

Department of Mathematical and Physical Sciences
Faculty of Graduate Studies
Concordia University of Edmonton



BRAILLE CONVERTER

SINDHIYA SENGOLE RAJ

Approved:

Dr.Nasim Hajari, Ph.D.

Supervisor

Date

Committee Member Name, Ph.D.

Committee Member

Date

Dr. Alison Yacyshyn, Ph.D.

Dean of Graduate Studies

Date

Abstract

Braille is a writing system utilized by visually impaired people, including blind, deafblind, or with low vision. Visually impaired people can be read on embossed paper or refreshable braille displays connecting to computers and smartphone devices. Braille is key to literacy for the visually impaired, but globally, only < 1 percentage of books[36] were converted into Braille. For blind learners, some courses can feel out of reach because books and exams may not be readily available in a braille format. Furthermore, depending on the number of pages and complexity, it may take up to 20 business days[35] to transcribe, emboss, and bind more essential documents. In addition to the above statement, Braille books cost more than most college textbooks. For example, converting just five chapters of a science book, the average order, into Braille can cost up to \$15,000. The main aim of this project is to reduce the time taken to convert the document and make all the resources available for visually impaired people. It contains four main components: Braille to Text, Text to Braille, Handwritten to Braille, and Audiobook. The application diminishes the conversion time required to convert the text into Braille. For instance, the manual conversion takes 2 to 3 days[35] to convert text into Braille for a small document, but the application can convert a small document in a few minutes.

Keywords: Braille, Uncontracted Braille(Grade 1),Text to Braille, Contracted Braille(Grade 2), Braille to Text, Handwritten to Braille, Audiobook, Tesseract OCR.

Contents

1	Introduction	1
2	Literature review	3
3	BackGround	5
3.1	Tesseract	6
3.2	Key components of Tesseract	7
3.2.1	Dataset:	7
3.2.2	Pre-Processor:	7
3.2.3	Adaptive Thresholding:	7
3.3	Connected Component Analysis:	7
3.4	Mapping English character to Braille	9
4	Project Design	9
4.1	Functionalities:	9
4.1.1	Text to Braille:	11
4.1.2	Braille to Text	11
4.1.3	Handwritten Text to Braille	13
4.1.4	Audiobook	15
4.2	Plugin	17
4.2.1	PyPDF2	17
4.2.2	Pdfplumber	17
4.2.3	Camelot	17
4.2.4	Pytesseract	17
4.2.5	Pillow	17
4.2.6	Fitz	17
4.2.7	pyttsx3	18
5	User Interface	18
5.1	Text to Braille	18
5.2	Braille to Text	19
5.3	Handwritten to braille	19
5.4	Audiobook	22
5.5	Limitations	22
5.6	Manual Conversion Vs Automatic converter	23
6	Conclusions	23
6.1	Future Scope	23

List of Tables

- 1 Alphabets mapped to the Braille Characters 10
- 2 Manual Conversion Vs Braille Converter 23

List of Figures

1	Braille cell dimensions	1
2	Braille Cell Dimensions	2
3	Braille Characters for Alphabets[37]	2
4	Braille Characters for numbers[37]	2
5	OCR Process flow	6
6	Process of Optical Character Recognition	6
7	Alphabets with Braille characters[37]	8
8	Home Page of Braille Converter	11
9	Braille Characters for Alphabets	12
10	Braille Characters for numbers	12
11	Braille Dot Pattern	12
12	Braille to Text input file	13
13	Output of Braille to Text	14
14	Input for the Handwritten Text to Braille	14
15	Output from the Handwritten Text to Braille	15
16	Input for Audiobook	16
17	Output from Audiobook	16
18	Home page of Braille Converter	18
19	Functionalities of Braille Converter	19
20	Text-to-braille page	20
21	File uploading section in text-to-braille	20
22	File uploading section in braille-to-text	21
23	File uploading section in handwritten-to-text	21
24	File uploading section in Audiobook	22
25	File uploading section in Audiobook	22

1 Introduction

Around 2.2 billion people globally have a near or distant vision problem. However, over a billion of these cases have not been addressed. In Sub-Saharan Africa, near vision impairment is estimated to be over 80 percent. In high-income regions such as Western Europe, North America, and Asia-Pacific, the rate of near vision impairment is lower than 10 percent. Despite their disabilities, visually and blind people can still contribute to society. However, they have difficulty communicating with the sighted people due to the use of a different language[1].

Louis Braille was a teacher from France who created the Braille system, which is a binary writing and reading system for the blind. It was founded in 1824. The system is made up of cells, each of which has six raised dots that are arranged in three rows. These six dots can be flat or raised according to the character in the Braille script. The number of combinations the six raised dots can create is 64, including the empty character "space."

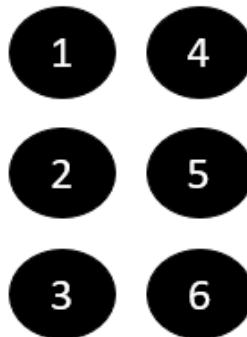
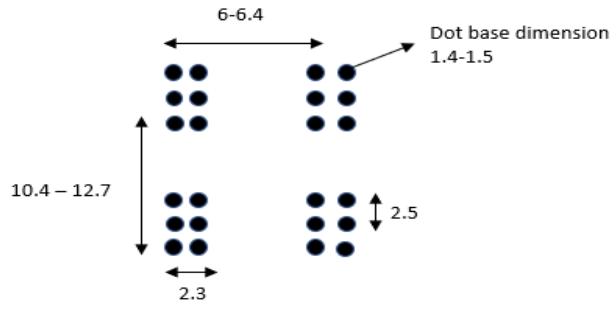


Figure 1: Braille cell dimensions

The dimensions of a Braille dot have been developed according to the tactile resolution of the person's fingertips. Accordingly, the horizontal and vertical space between dots in a character and the distance between cells representing a word is known. Dot height is approximately 0.02 inches (0.5 mm); the horizontal and vertical spacing between dot centers within a Braille cell is approximately 0.1 inches (2.5 mm); the blank space between dots on adjacent cells is around 0.15 inches (3.75 mm) horizontally and 0.2 inches (5.0 mm) vertically as shown in figure 1. A traditional Braille page is 11 inches by 11.5 inches and typically has a maximum of 40 to 43 Braille cells per line and 25 lines[2].

Although Braille cells are used worldwide, each cell's meaning depends on the language they are used to depict. For example, in English Braille, there are three levels of encoding: Grade 1, a letter-by-letter transcription used for basic literacy; Grade 2,



All dimensions are in Millimeter

Figure 2: Braille Cell Dimensions

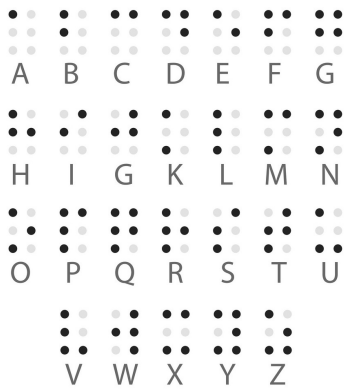


Figure 3: Braille Characters for Alphabets[37]

1	2	3
4	5	6
7	8	9
	0	

Figure 4: Braille Characters for numbers[37]

an addition of abbreviations and contractions[3]; and Grade 3, which contains over 300 abbreviations and contractions that reduce the amount of Braille codes needed to represent written text. Some of the contracted words are represented in figure 4 in its Braille format.

Braille can be seen as the world's 1st binary encoding method for illustrating the characters of a writing system. However, minimal numbers of Braille books are available for usage. Furthermore, printing Braille books is a time-consuming process. The requirement for special printers and software adds to their limited availability.

This project mainly focuses on converting a document or image file into its corresponding Braille Characters and vice versa. The text will be recognized and converted to respective Braille characters, and the presence of dots in the Braille cells must be identified to recognize the characters. By this, we can reduce the conversion time.

2 Literature review

Two decades ago, research began to assess and speed up the writing process for visually challenged people. Then, in 1965, touchscreen devices became part of their everyday life. It becomes easy for people to interact with the touchscreen without the help of other tools. Several applications have been developed to assist visually impaired people in accomplishing their day-to-day tasks. The application are Braille Tap and Nav Tap[4], V-Braille[5], TypeIn Braille[6], Braille Touch[7], TapTapSee[8], Braille Play[9], Edge Braille[10], Braille Ecran[11], BeSpecular[12], Color Teller[13], KNFBReader[14], Text to Speech Synthesizer[15], etc.

Several experimenters have used Braille in natural language conversion mechanisms, considering it an important research topic. Several approaches have been used in multiple studies to achieve better results for Braille in natural language predictions. Deep Learning and Machine learning techniques with distinct feature extraction methods were utilized in these studies.

A study by Hassan et al[16]. designed a feed-forward neural network with 190 hidden neurons in the hidden layer and 400 inputs in the input layer to convert English books to Braille characters. For this purpose, capital and small English alphabets, 0-9 numbers, and with space and without space were also used. The neural networks reached satisfactory results for converting text into Braille. The network was made with an uncomplicated structure that included several layers and neurons in each layer. For precision, an automated value thresholding algorithm was used. This method can convert the text to the respective braille character. Because of this method visually impaired can read books, novels, and so on with low cost, flexibility and portability.

Likewise, Braille character segmentation was performed using image processing methods. Then, the Support Vector Machine (SVM) with the Histogram of Oriented Gradients (HOG)[17], the feature extraction method, was used to translate English

Braille characters. The HOG (Histogram of Oriented Gradients) is a feature descriptor that extracts features from image data. It is commonly used for image or object detection tasks. For this purpose, Megha et al. suggested another SVM approach to have a preprocessing step like contrast enhancement and noise removal on images of Braille. In order to obtain accuracy, preprocessing was performed at each step to predict Braille to English characters. A better performance achieved 96% accuracy using this strategy. However, this study also worked only for scanned documents [17].

A study by Raghunandan et al[20]. has developed an algorithm capable of converting text to Braille and vice versa. The raw data was retrieved in the form of an image. Image acquisition and character segmentation, ASCII characters will be converted to a Braille script using this algorithm. Regardless, it has some boundaries to the scanned-based input method[18].

Li et al[19]. have used the HAAR feature extraction technique, including the SVM, to convert a scanned Braille document to extract the text from it. The HAAR wavelet has often been used in digital image processing as a feature extraction approach, and 2D HAAR wavelets detect the edges in different orientations. Cropped images, as well as original images, were used to train the SVM. First, the word segmentation and blank spaces were removed. Then, The database was accessed to compare the output, and in the Braille pad, the Braille characters were displayed. Finally, English text to equivalent Braille script.

In a study by Singh et al[21]., the system could read English and Hindi words. Once the space has been removed from the words, the words will be broken into corresponding characters; then, those words will be matched with the table[22]. Likewise, a study was conducted to convert scanned Braille Hindi, English, and Tamil words. For this, 20 scanned braille sheets were used to train. In addition, image enhancement and noise were removed to perform Braille to text conversion. As a result, 100 percent accuracy was obtained in converting English and Tamil braille scripts. Higher than 97% of accuracy was achieved in the rest.

The Braille converter is a web-based application that translates a scanned pdf file or a pdf file into Braille cells and sends it to a Braille embosser, producing a hard copy of the original text in Braille script and vice versa.

Few existing translators are available on the internet to convert text to Braille, such as two blind brothers, win Braille, Braille translator, and Braille master.

1. **Two Blind Brothers:**

It displays Braille as the user types characters and provides the conversion from Braille to English. This converter is OS-independent, and the language used is java[23].

2. **Win Braille:**

It is a text editor concentrated on those who need to decrypt text into Braille, and it can be printed on embossed paper[24].

3. **Braille translator:**

It is a text-to-Braille program that converts text documents into a format suitable for producing them through a Braille printer. The user can edit, save, use style templates and enable translation rules[25].

4. **Braille Master:**

As in [26], the Braille Master package comes with Windows and DOS versions. In addition, a large print facility suitable for partially sighted persons is also included in this package.

3 Background

In the Braille converter, two types of Grade systems have been used. The Grade system has two classifications, namely,

Uncontracted Braille, Grade 1:

It is the fundamental form of Braille. It utilizes all the alphabet in the English language and is often used by children or individuals who are beginners to reading and writing in Braille[27]. With uncontracted Braille, phrases are spelled out letter by letter. Regardless, this form of Braille takes up considerable space, and documents written in uncontracted Braille will be significantly prolonged.

Contracted braille, Grade 2:

It is a more complicated form naturally learned after learning uncontracted Braille. It is the most generally used form of Braille. Contracted Braille is a “shortcut system”[27] where one letter might convey an entire word. There are letter combinations, or contractions, that describe whole words without spelling out each letter in the word. This method diminishes the overall number of cells needed and the volume of pages required to print books and other written content. Contracted Braille takes up little space and enhances an individual’s pace in both writing and reading. This has been used in public areas.

3.1 Tesseract

It is an optical character recognition (OCR) an engine, which can be used on various operating systems, including Raspberry Pi OS. It is a freeware licensed under the Apache License Version 2.0 and was originally developed at Hewlett-Packard between 1985 and 1994. Tesseract was open-sourced by HP in 2005 and was later bought by Google. It is pretty more accurate as compared to other OCR engines. Most OCR frameworks are built on top of Tesseract, and thus it has become popular. Tesseract supports a wide variety of languages, including English, Spanish, Thai up to Tamil, Uzbek, and Yiddish.

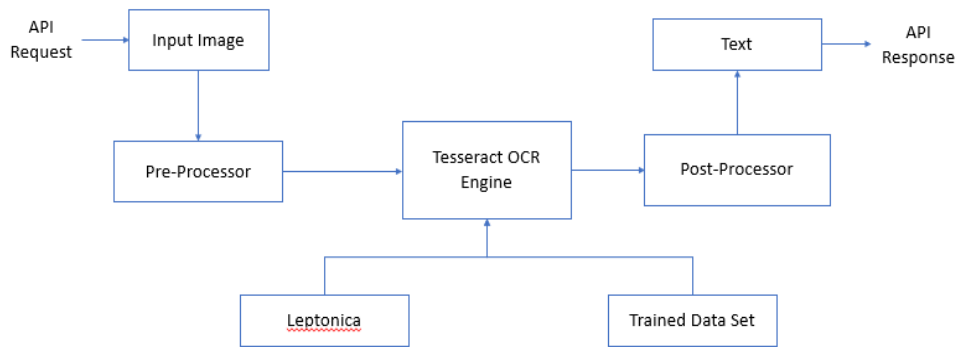


Figure 5: OCR Process flow

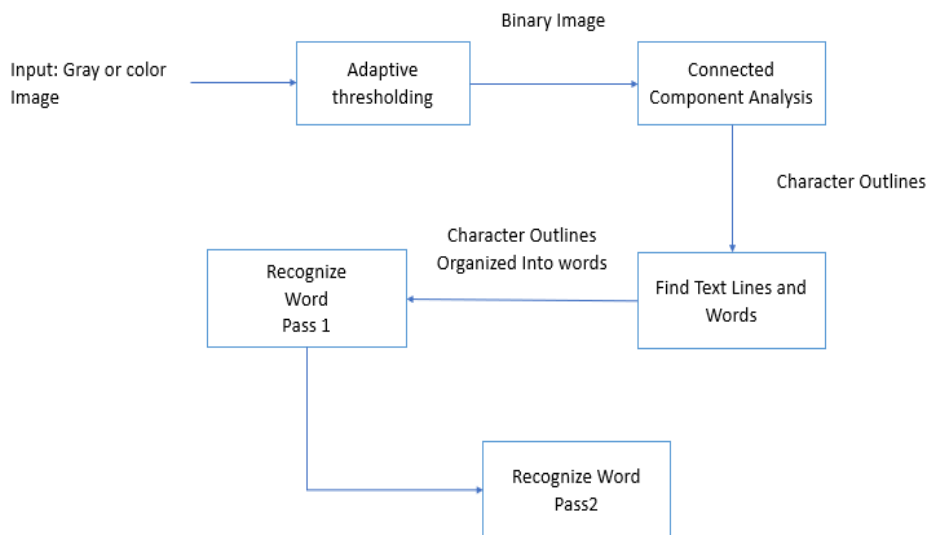


Figure 6: Process of Optical Character Recognition

3.2 Key components of Tesseract

3.2.1 Dataset:

Tesseract makes use of open source “eng.trained” data available in their tessdata best repository to train the model on English language.

3.2.2 Pre-Processor:

Tesseract is based on neural network system which makes use of LSTM network that delivers significantly higher accuracy. Here, the input image is processed and broken down into various chunks which can be single line in input image and the given to LSTM model line by line.

3.2.3 Adaptive Thresholding:

Pi Tesseract utilizes adaptive thresholding, where in the variations at different locations (spatial variations) and the variations in the intensity of light (luminous variations) are considered.

3.3 Connected Component Analysis:

In Graph Theory, connected component analysis is an algorithmic application where the parts with very high correlation i.e. subsets of connected parts are uniquely labeled based on a set of parameters i.e. given heuristic.

Let’s take example to understand OCR Process Flow using the following steps:

- **Input**

The scanned PDF document or an image file will be uploaded as an input.

- **Pre-processing of Input Image/document**

Based on intensities of each region foreground image object is differentiate from background by using adaptive thresholding. Through this process binary image is obtained which will be send to further step. Connected component analysis is an algorithmic application where the parts with very high correlation i.e. subsets of connected parts are uniquely labeled based on a set of parameters i.e. given heuristic.

- **Component Analysis Phase**

In this step character outlines of component are stored. With the help of

outlines, it is simple to detect and recognize black text on white background. These outlines are gathered together and nested into blobs.

- **Find Text Line and Words**

Using character spacing technique trying to get words form line. Blobs are arranged as a text line, and these lines are analyzed by proportional text. After this trying to recognize words in two passes. To identify a word, we need 2 passes. In the first pass, word is detected in its context and forwarded to pass 2. In the second pass, it is confirmed that word detected in pass 1 is correct.

- **Mapping English character with Braille**

Lookup tables are special tables which contain two columns, one for English character and one for its string representation which specifies the active dots in Braille. For grade one Braille, the tokenized text is checked inside lookup table. If any match exists, the corresponding string, consistent with values, will be sent.

CHARACTER	BRAILLE	BRAILLE DOTS
a	•	1
b	••	12
c	••	14
d	••	145
e	••	15
f	••	124

Figure 7: Alphabets with Braille characters[37]

- **Send text to Braille display unit**

The converted output from the previous step will be printed in a word document as an output.

3.4 Mapping English character to Braille

Consider the input sentence.

Example: "I LIKE TO READ BOOKS IN MORNING."

This sentence is deconstructed into tokens by Pi Tesseract OCR, which is as given below:

I,L,I,K,E,T,O,R,E,A,D,B,O,O,K,S,I,N,M,O,R,N,I,N,G.

In grade one Braille, all the characters are going to be matched from the search table which is able to lookup a section of our information. Lookup tables are special tables which contain two columns, one for English character and one for its string representation which specifies the active dots in Braille(as shown in Table 1).

For grade one Braille, the tokenized text is checked inside lookup table. If any match exists, the corresponding string, consistent with values, will be sent.

4 Project Design

The Braille Converter is developed as a web-based program accessible from the Internet. The Braille Converter is a translator which is capable of converting sentences with spaces and converting them into Braille. In Standard English Braille, many of the sixty-three cells will correspond to a letter of the alphabet or a special character. In addition, a few cells called contractions will represent short words or syllables frequently encountered in English.

When we convert English Text to Braille, the following chart is used as the database, the input text is matched for the corresponding Braille representation, and if the character matches, the corresponding Braille is displayed.

4.1 Functionalities:

The braille converter was developed with four functionalities that are capable of converting Text to Brail and vice versa in figure 8, such as,

1. Text to Braille,
2. Braille to Text,
3. Handwritten Text to Braille, and
4. Audiobook.

Table 1: Alphabets mapped to the Braille Characters

Characters	Decimal Code
a	chr(10241)
b	chr(10243)
c	chr(10249)
d	chr(10265)
e	chr(10257)
f	chr(10251)
g	chr(10267)
h	chr(10259)
i	chr(10250)
j	chr(10266)
k	chr(10245)
l	chr(10247)
m	chr(10253)
n	chr(10269)
o	chr(10261)
p	chr(10255)
q	chr(10271)
r	chr(10263)
s	chr(10254)
t	chr(10270)
u	chr(10277)
v	chr(10279)
w	chr(10298)
x	chr(10285)
y	chr(10301)
z	chr(10293)

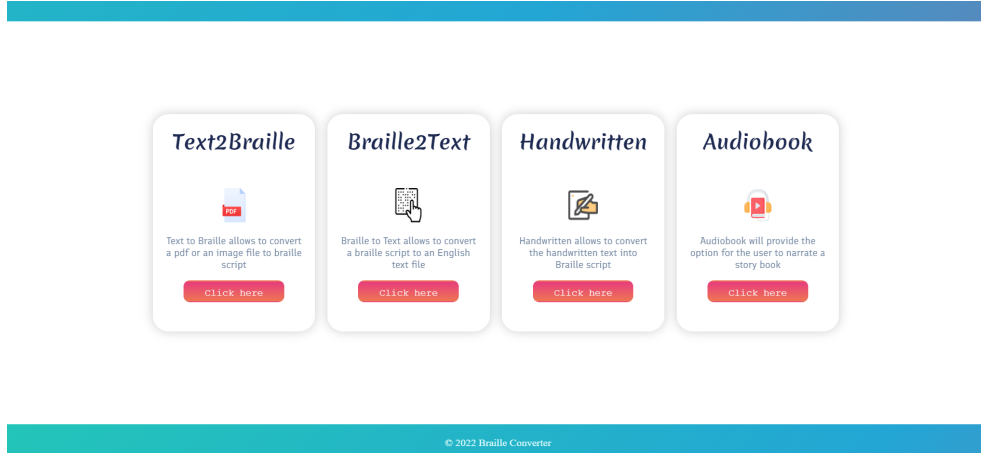


Figure 8: Home Page of Braille Converter

4.1.1 Text to Braille:

A scanned document, pdf, and image can be uploaded to the application. The resolution of scanned pdf and image files should not be less than 300 dpi to have good accuracy. For extracting the Text from scanned pdf and image files, OCR has been implemented to retrieve Text from the documents.

The below steps are followed when English Text is extracted and converted into braille as follows,

1. Read a character string,
2. Separate the words based on blank space,
3. Break the corresponding word into corresponding letters,
4. Access look-up the table for matching of characters of input string with look-up table characters,
5. If characters match, print the corresponding Braille character as it is,
6. Steps 4 and 5 will be reprised; the characters of the input string are matched with look-up table characters.

We can convert English Text to Braille in the steps mentioned above. This conversion is based on pattern matching. Then the output will be generated as a word document.

4.1.2 Braille to Text

The six dot cell representation of the Braille character could be numbered from 1 to 6, starting from the left of top to bottom right in the order left to right and top to bottom, as shown in figure 11.

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Figure 9: Braille Characters for Alphabets

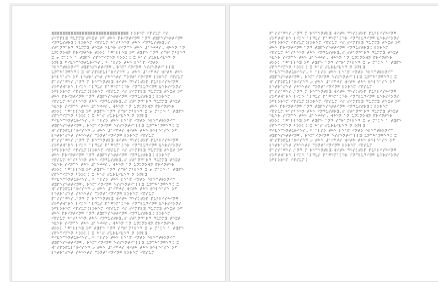


Figure 10: Braille Characters for numbers

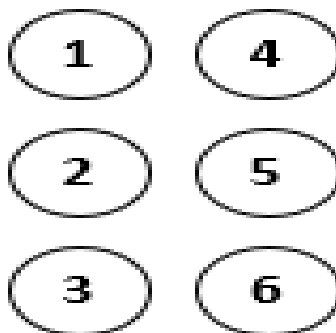


Figure 11: Braille Dot Pattern

In the Braille to Text conversion method, the Braille word document will be uploaded as an input, and the input will be parsed line by line, which by a sequence of steps is converted to appropriate Text. First, the Braille cells are identified and converted to a binary sequence. The binary sequence is then mapped to the corresponding alphabets or contracted words.

The steps below are followed for converting Braille to English text, such as

1. Read a Braille string.
2. Based on the blank space, the words will be separated.
3. The words will be split into letters.
4. Splitted Braille character will be matched with the Table and replaced with the appropriate English Text.
5. If characters match, print the corresponding Braille character as it is.
6. Steps 4 and 5 will be reprised; the characters of the input string are matched with table characters.

We can convert Braille Script to English Text in the steps mentioned above. This conversion is based on pattern matching. Then the output will be generated as a word document.

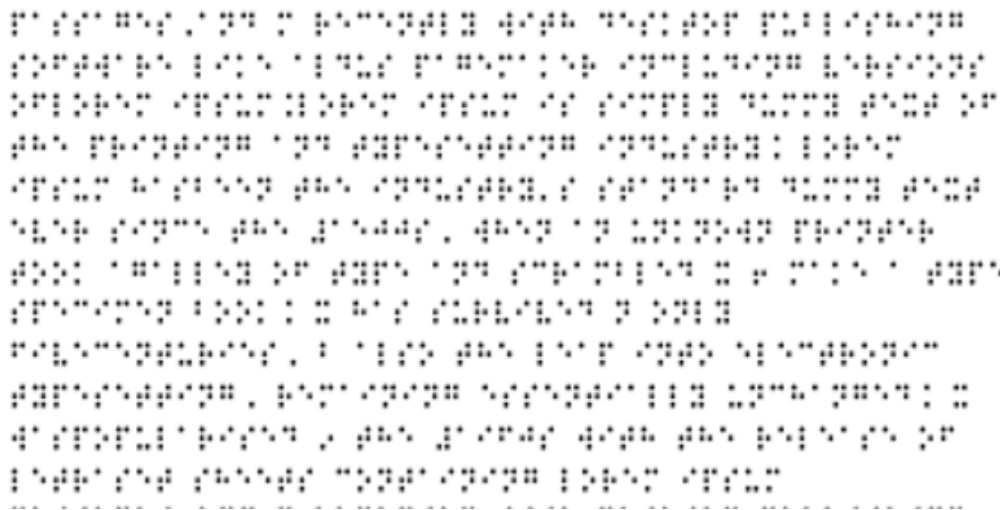


Figure 12: Braille to Text input file

4.1.3 Handwritten Text to Braille

The OCR is a tool used to recognize the characters in the image file, and for this purpose, the Pytesseract is used. Pytesseract is an OCR tool for python that also serves as a wrapper for the Tesseract-OCR Engine. It can read and recognize Text in images and is commonly used in python ocr image to text use cases[28].

Modern Braille script was invented by a French inventor named Louis Braille. Louis was born in Coupvray, France on 04 January 1809. His father was a leather-maker. At the age of three, little Louis was playing in his father's workshop and he accidentally stuck a stitching awl in one of his eyes. Later, his injured eye got infected. Due to lack of antibiotics, soon, the infection spread to his other eye too. Gradually, by the age of five, the Louis Braille lost sight in both of his eyes.

Figure 13: Output of Braille to Text

This is a handwritten
example
Write as good as you can.

Figure 14: Input for the Handwritten Text to Braille



Figure 15: Output from the Handwritten Text to Braille

Ensure the image is pre-processed appropriately to avoid all the ways tesseract output accuracy can drop. The following steps are used to enhance the image,

1. **Rescaling:**

Pytesseract works excellently on images with a DPI of at least 300 dpi, so it may be advantageous to resize images[29].

2. **Binarization:**

The image will be converted to black and white. Since the Pytesseract accomplishes this internally, the result can be suboptimal, mainly if the page background is of uneven darkness[29].

3. **Noise removal:**

Noise is a random variation of brightness or color in an image, making the text more challenging to read. Tesseract cannot remove certain types of noise in the binarisation step, which can cause accuracy rates to drop[29].

4. **Deskewing:**

A skewed image is when a page has been scanned when not straight. Pytesseract's line segmentation diminishes significantly if a page is too skewed, severely influencing the OCR quality. To handle this, rotate the page image so the text lines are horizontal[29].

By following the above steps, better results can be obtained.

4.1.4 Audiobook

In the Audiobook functionality, the uploaded pdf document will be divided into a single page, and the Pyttsx3 library from python is used to read from the split page, and it will be converted as an audio file.

THE GOLDEN BIRD

A certain king had a beautiful garden, and in the garden stood a tree which bore golden apples. These apples were always counted, and about the time when they began to grow ripe it was found that every night one of them was gone. The king became very angry at this, and ordered the gardener to keep watch all night under the tree. The gardener set his eldest son to watch; but about twelve o'clock he fell asleep, and in the morning another of the apples was missing. Then the second son was ordered to watch; and at midnight he too fell asleep, and in the morning another apple was gone. Then the third son offered to keep watch; but the gardener at first would not let him, for fear some harm should come to him: however, at last he consented, and the young man laid himself under the tree to watch. As the clock struck twelve he heard a rustling noise in the air, and a bird came flying that was of pure gold; and as it was snapping at one of the apples with its beak, the gardener's son jumped up

Figure 16: Input for Audiobook



Figure 17: Output from Audiobook

4.2 Plugin

4.2.1 PyPDF2

PyPDF2 is a pure-python library to work with PDF files. The pyPDF2 module has been utilized to work with the existing PDF files. A Python library to extract document information and content, split documents page-by-page and merge documents. The PdfReader () function reads from the pdf file[34].

4.2.2 Pfdplumber

Pfdplumber can read from both text and the table, and by using this library, the table can be separated from the rest of the text in the PDF file. The extract_text() has the potential to extract text and the extract_table() used to separate the table from the text.

4.2.3 Camelot

Camelot is a python library and can retrieve the table from a pdf file with an exact structure. The camelot.readpdf() is used to read from the pdf file.

4.2.4 Pytesseract

Python-tesseract is an optical character recognition (OCR) tool for python. It will recognize and “read” the text embedded in images. Python-tesseract will print the recognized text instead of writing it to a file. The image_to_string() converts text extracted from an image to a string[31].

4.2.5 Pillow

Since the Python-tesseract cannot read directly from the image, it requires the help of the Pillow library. Pillow is a Python Imaging Library (PIL), which adds support for opening, manipulating, and saving images. The current version identifies and reads a large number of formats. Write support is intentionally restricted to the most commonly used interchange and presentation formats[33].

4.2.6 Fitz

A Python binding that supports MuPDF 1.9a - “a lightweight PDF and XPS viewer.” MuPDF can access files in PDF, XPS, OpenXPS, and EPUB (e-book) formats, and it is known for its top performance and high rendering quality. Unfortunately, Pytesseract does not support PDF format; the Fitz is used to convert the .pdf format into .png format[32].

4.2.7 pyttsx3

pyttsx3 is a text-to-speech conversion library in Python. Unlike the other libraries, it works offline and agrees with Python 2 and 3. An application invokes the `pyttsx3.init()` factory function to get a reference to a `pyttsx3. Engine` instance is a very easy-to-use tool that converts the entered text into speech. The `pyttsx3` module supports two voices first is female, and the second is male, which is provided by “sapi5” for windows[30].

5 User Interface

The Braille converter is a web-based application developed with python programming that has four functionalities(as shown in figure 18 and 19), namely,

1. text to braille,
2. braille to text,
3. Handwritten to braille,
4. Audiobook.

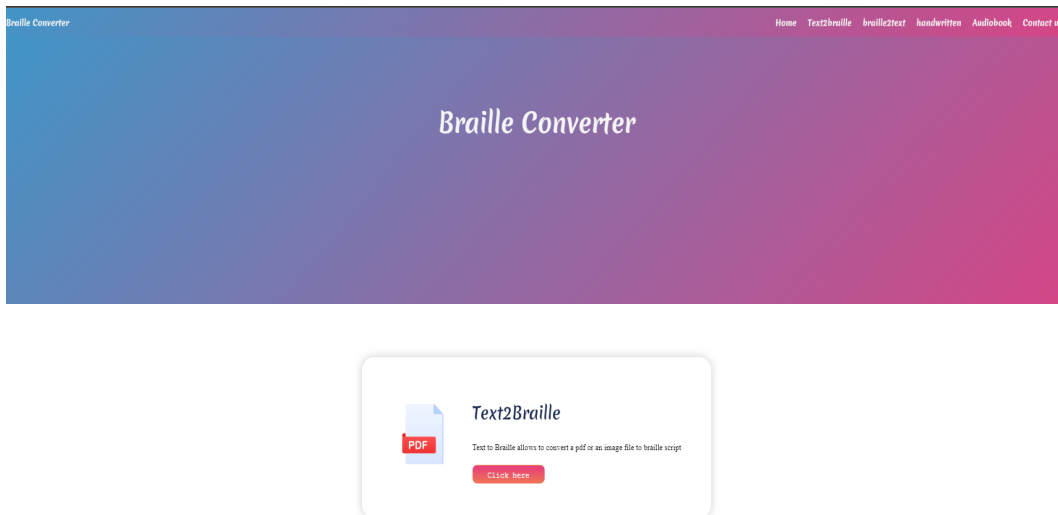


Figure 18: Home page of Braille Converter

5.1 Text to Braille

Text to braille allows three types of file input from the user(in figure 20): text-based PDF, scanned PDF, and image(.png,.jpg,.jpeg). Once the file is uploaded into the text to braille, the file will be divided page by page. Finally, the text will be

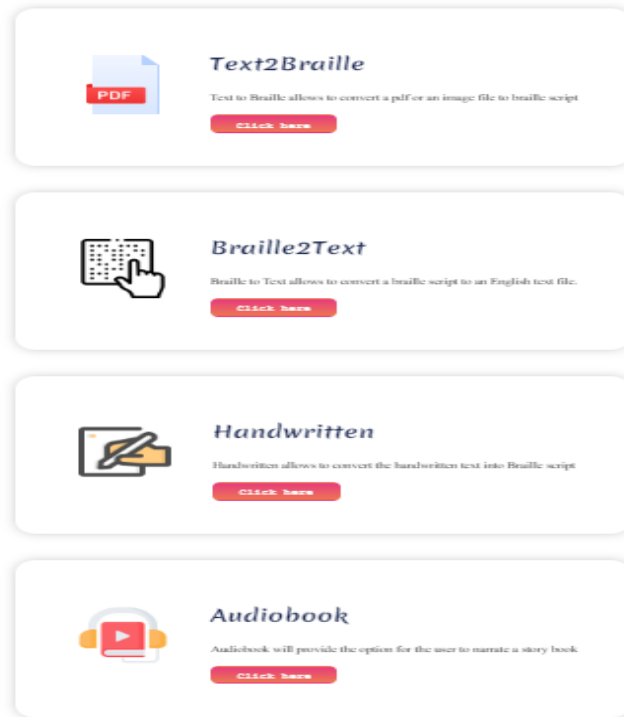


Figure 19: Functionalities of Braille Converter

extracted from the uploaded file and written into the input file in .txt format. Line-by-line characters will be matched with the equivalent braille character written in a word file page by page. Ultimately, all the pages will be merged as a single word file.

5.2 Braille to Text

The braille-to-text is the reverse functionality of the text to braille. It accepts the word file and.txt file from the user. This functionality will match the braille characters with the comparable English alphabet character. Finally, its output will be printed in the word file as a final output.

5.3 Handwritten to braille

The accepted file formats are png, jpg, and jpeg by this functionality. In addition, OCR is implemented to recognize the written text in the image. Finally, the extracted text from the image will be compared and converted into an appropriate braille character.

Text-to-Braille

Please upload Text-based PDF file

[Click here](#)

Please upload the scanned PDF file

[Click here](#)

Please upload the Image file

[Click here](#)

Figure 20: Text-to-braille page

The screenshot shows a file upload area with a dashed blue border. Inside, there is a document icon with a red horizontal line through it, and the text "Click here to add file". Below this is a blue progress bar that is 100% full. At the bottom of the upload area is a green "Convert" button with a loading spinner. Below the button, a message reads: "File [s_book.pdf](#) has been uploaded successfully. Now click 'Convert'".

Figure 21: File uploading section in text-to-braille



Figure 22: File uploading section in braille-to-text

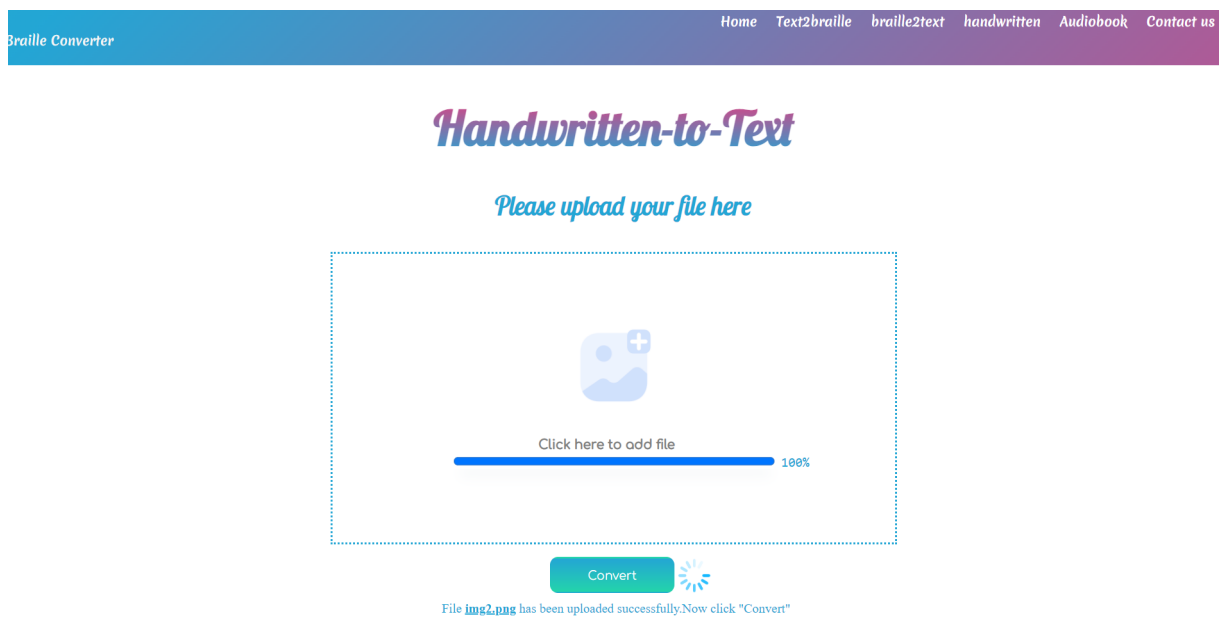


Figure 23: File uploading section in handwritten-to-text

5.4 Audiobook

Audiobooks are voice recordings of the textbook that you listen to instead of reading. Audiobooks can be exact word-for-word versions of books. PDF files are allowed in the Audiobook functionality, and once the file has been uploaded, it will be converted into an audio file by this audiobook.

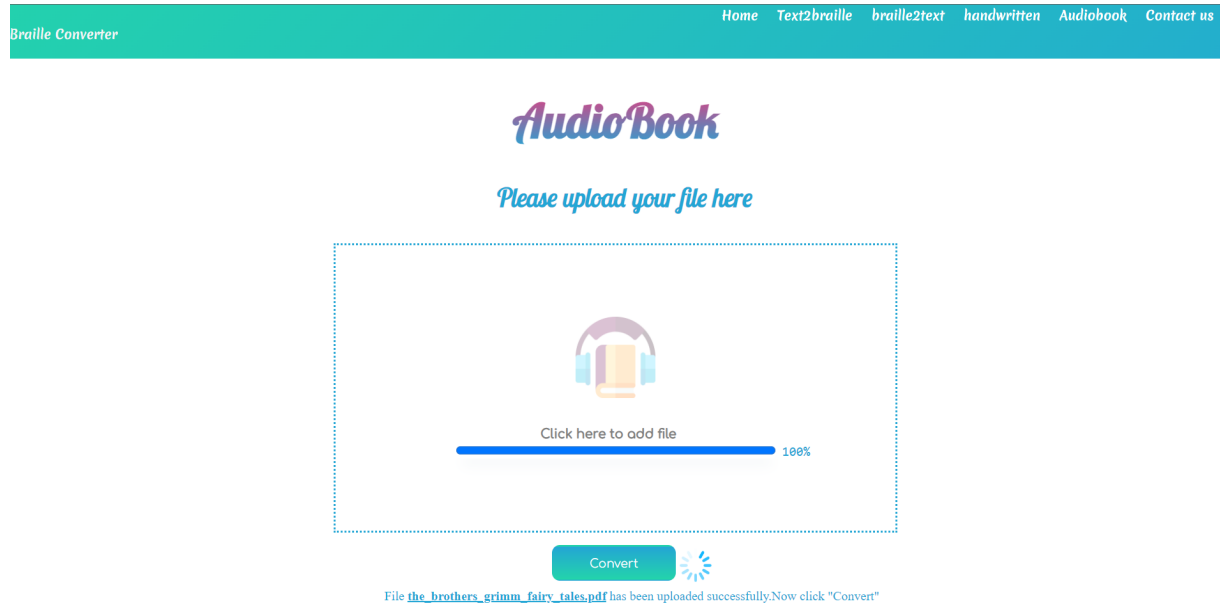


Figure 24: File uploading section in Audiobook

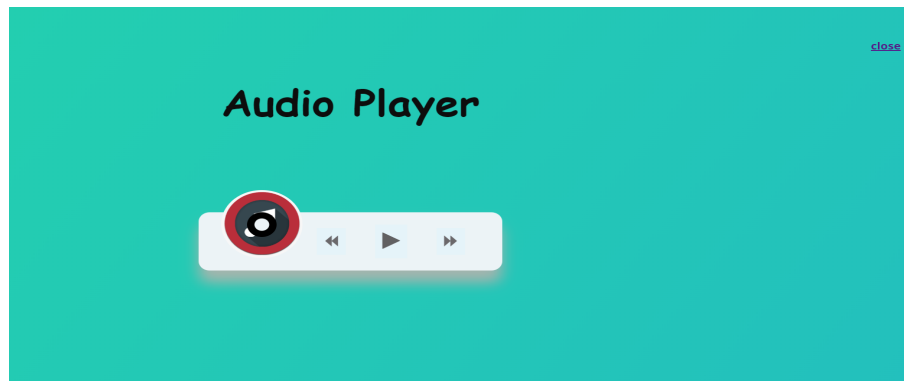


Figure 25: File uploading section in Audiobook

5.5 Limitations

1. Uploading multiple files simultaneously is not allowed in the Braille converter.
2. The Braille converter does not support an excel sheet files.

3. Doc, Docx, and txt files are not allowed in the text to braille functionality.
4. Pictures in the input file won't be retrieved unless it is a text-based picture. So, for instance, graphs or charts will be ignored.
5. Limited grade 2 Braille scripts are being used.
6. The Braille converter cannot retrieve a paragraph of handwritten text. In addition, it supports only a few lines of handwritten text.

5.6 Manual Conversion Vs Automatic converter

Table 2: Manual Conversion Vs Braille Converter

	Manual Conversion	Braille Converter
Conversion time	1 to 5 days	in minutes
Conversion Cost	\$5.50	free of cost
Graph and chart pictures	supported	not supported

6 Conclusions

The application can convert English text to grade 1 and grade 2 Braille script within a few minutes, and with the help of this application, the Braille documents can be printed out quickly. By this, they can be able to read all books in Braille script. In addition, the conversion cost will be reduced since it is an open-source application.

6.1 Future Scope

Though this application is efficient, there are points where it needs to be improved. For example, the application supports PDF and word files. In the future, it should be able to convert an excel sheet and graphs into Braille.

References

- [1] Visual impairment and blindness. W. H. Organization. (2021), Online: <https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment>
- [2] Development of a bangla character recognition (bcr) system for generation of bengali text from braille notation, Halder, Santanu and Hasnat, Abul and Khatun, Amina and Bhattacharjee, Debotosh and Nasipuri, Mita- International Journal of Innovative Technology and Exploring Engineering 2013

- [3] CONVERSION OF BRAILLE TO TEXT IN ENGLISH, HINDI AND TAMIL LANGUAGES(2021),S.Padmavathi,Manojna K.S., Sphoorthy Reddy . and Meenakshy. International Journal of Computer Science, Engineering and Applications.
- [4] Guerreiro, T.; Lagoá, P.; Santana, P.; Gonçalves, D.; Jorge, J. NavTap and BrailleTap: Non-Visual Texting Interfaces. In Proceedings of the Rehabilitation Engineering and Assistive Technology Society of North America Conference (Resna), Arlington, VA, USA, 27 June 2008
- [5] Jayant, C.; Acuario, C.; Johnson, W.A.; Hollier, J.; Ladner, R.E. VBraille: Haptic Braille Perception Using a Touch-Screen and Vibration on Mobile Phones. In Proceedings of the 12th International ACM SIGACCESS Conference on Computers and Accessibility, ASSETS 2010
- [6] Mascetti, S.; Bernareggi, C.; Belotti, M. TypeInBraille: A Braille-Based Typing Application for Touchscreen Devices. In Proceedings of the 13th International ACM SIGACCESS conference on Computers and Accessibility, Dundee Scotland, UK, 24–26 October 2011
- [7] Frey, B.; Southern, C.; Romero, M. BrailleTouch: Mobile Texting for the Visually Impaired. In Universal Access in Human-Computer Interaction; Springer: Berlin/Heidelberg, Germany, 2011; pp. 19–25.
- [8] CloudSight TapTapSee—Blind and Visually Impaired Assistive Technology. Image Recognition API. Available online: <https://taptapseeapp.com/> (accessed on 2 September 2020).
- [9] Milne, L.R.; Bennett, C.L.; Ladner, R.E.; Azenkot, S. BraillePlay: Educational Smartphone Games for Blind Children. In Proceedings of the 16th International ACM SIGACCESS Conference on Computers Accessibility, Rochester, NY, USA, 20–22 October 2014; ACM: New York, NY, USA, 2014;
- [10] Mattheiss, E.; Regal, G.; Schrammel, J.; Garschall, M.; Tscheligi, M. Edge-Braille: Braille-Based Text Input for Touch Devices. *J. Assist. Technol.* 2015, 9, 147–158.
- [11] Siqueira, J.; Soares, A.d.M.F.A.; Silva, C.R.G.; de Oliveira Berretta, L.; Ferreira, C.B.R.; Felix, I.M.; Luna, M.M. BrailleÉcran: A Braille Approach to Text Entry on Smartphones. In Proceedings of the 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC), Atlanta, GA, USA, 10–14 June 2016
- [12] BeSpecular. Available online: <https://www.bespecular.com/> (accessed on 20 February 2020)
- [13] Kacorri, H.; Kitani, K.M.; Bigham, J.P.; Asakawa, C. People with Visual Impairment Training Personal Object Recognizers: Feasibility and Challenges. In

- Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, Denver, CO, USA, 6–11 May 2017; ACM: New York, NY, USA
- [14] Reader, K. KNFB Reader App Features the Best OCR. Available online: <https://www.knfbreader.com/> (accessed on 10 February 2020).
- [15] Priyadarsini, S.; Ajit, P.; Nayak, K.; Champati, S. A Survey on Speech Synthesis Techniques in Indian Languages. *Multimed. Syst.* 2020
- [16] Hassan, M.Y.; Mohammed, A.G. Conversion of English Characters into Braille Using Neural Network 1. *Iraqi J. Comput. Commun. Control Syst. Eng.* 2020
- [17] Perera, T.D.S.H.; Wanniarachchi, W.K.I.L.I. Optical Braille Recognition Based on Histogram of Oriented Gradient Features and Support-Vector Machine. *Int. J. Eng. Sci.* 2018,
- [18] Dharme, V.S.; Karmore, S.P. Designing of English Text to Braille Conversion System: A Survey. In *Proceedings of the 2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, Coimbatore, India, 19–20 March 2019
- [19] Li, J.; Yan, X.; Zhang, D. Optical Braille Recognition with Haar Wavelet Features and Support-Vector Machine. In *Proceedings of the International Conference on Computer, Mechatronics, Control and Electronic Engineering*, Changchun, China, 24–26 August 2020
- [20] Raghunadan, A.; MR, A. The Methods Used in Text to Braille Conversion and Vice Versa. *Int. J. Innov. Res. Comput. Commun. Eng.* 2017
- [21] Perera, T.D.S.H.; Wanniarachchi, W.K.I.L.I. Optical Braille Recognition Based on Histogram of Oriented Gradient Features and Support-Vector Machine. *Int. J. Eng. Sci.* 2018, 8, 19192–19195.
- [22] Singh, M.; Bhatia, P. Automated Conversion of English and Hindi Text to Braille Representation. *Int. J. Comput. Appl.* 2019, 4, 25–29.
- [23] <https://twoblindbrothers.com/pages/braille>
- [24] AbdulMalik S. Al-Salman, "A Bi-directional Bi-Lingual Translation Braille-Text System", *J. King Saud University*, Vol. 20, *Comp. Info. Sci.*, pp. 13-29, Riyadh
- [25] <https://www.brailletranslator.org/>
- [26] <http://www.braillemaster.com>
- [27] <https://tbase.com/what-are-the-different-types-of-braille/: :text=The%20two%20most%20common%20forms,most%20basic%20form%20of%20braille.>
- [28] <https://nanonets.com/blog/ocr-with-tesseract/: :text=Tesseract%20is%20an%20open%20source,a>
- [29] <https://tesseract-ocr.github.io/tessdoc/ImproveQuality.html#rescaling>

- [30] <https://www.geeksforgeeks.org/python-text-to-speech-by-using-pyttsx3/>
- [31] <https://pypi.org/project/pytesseract/>
- [32] <https://pypi.org/project/PyMuPDF/1.9.2/>
- [33] <https://zetcode.com/python/pillow/>
- [34] <https://www.journaldev.com/33281/pypdf2-python-library-for-pdf-files>
- [35] <https://brailleworks.com/turnaround-time-braille-printing/>
- [36] <https://www.theguardian.com/commentisfree/2012/aug/17/peter-white-books-braille-ereaders>
- [37] <https://www.pharmabraille.com/pharmaceutical-braille/the-braille-alphabet/>