Journal of Theoretical and Applied Information Technology
15th December 2021. Vol.99. No 23

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

PERFORMANCE EVALUATION OF TWO BRAILLE-TO-SPANISH TRANSCRIPTION APPLICATIONS: MOBILE DEVICE VS. PC

[1]EDWAR JACINTO GOMEZ, [2]FREDY H MARTINEZ S, [3]FERNANDO MARTINEZ SANTA
[1,2,3] Facultad Tecnológica, Universidad Distrital Francisco José de Caldas, Bogotá, Colombia
E-mail: [1]ejacintog@udistrital.edu.co, [2]fhmartinezs@udistrital.edu.co, [3]fmartinezs@udistrital.edu.co

ABSTRACT

In Colombia, the participation of people with some disability in the different educational levels is still deficient. Although it has been growing since 2010, the CNPV estimates that the illiteracy rate for the population with some disability is still three times higher than the national figure. However, the country has been adopting a policy of Inclusive Education focused on reducing barriers to learning and participation for all, which implies a high level of commitment and understanding by the entire educational community to achieve good practices. When analyzing the specific case of people with visual impairment, it is evident that one of the main mediating agents of inclusive education is teachers. They, in some cases, have led their students' reading-writing process to be ineffective due to their lack of knowledge in the Braille language. An example of this is the significant amount of time they take to transcribe texts in Braille, decreasing the ability to evaluate or rectify whether or not the student understands a text, preventing timely feedback. This situation hinders how students learn, understand, and produce texts.

Therefore, the use of technological tools that facilitate the teaching work is proposed by developing applications for transcription of Braille text to literary Spanish both on mobile devices and personal computers to minimize this problem and support the teaching-learning process in the classroom. Initially, the article describes the application developed for mobile devices with an Android operating system and the software application developed for PCs using Matlab. Subsequently, the performance analysis of applications is presented in terms of processing time, transcription error percentages, and response to different test conditions. Finally, the results and improvements concerning previous works are presented in terms of versatility and robustness achieved in detecting Braille dots, cells, and characters with different lighting characteristics and types of sheets.

Keywords: *Braille, Visual Impairment, Inclusive Education, Software, OBR.*

1. INTRODUCTION

Globally, an estimated 93 million children (1 in 20, in the 0-14 age range) live with some form of disability, moderate or severe. In most low- and middle-income countries, children with disabilities are more likely to be out of school than any other group of children. First school enrollment rates for children with disabilities are meager, and even if they do attend school, these children are more likely to be early school dropouts. It translates into low literacy rates in adulthood, as their right to education is often not realized, delaying their access to other rights and creating multiple barriers to reaching their potential and participating effectively and equitably in their communities. [1]

In the case of Colombia, in terms of access to education for children with disabilities, the 2010 "Situation of Education in Colombia" report [2] shows that children between ages 3 and 5 without disabilities attend 11% more preschool than children with disabilities. In addition, it estimates that 1.2% of the student population attending primary school has a disability; in secondary and middle school, the proportion of adolescents and young people with disabilities who attend is 12%, while those of the same age without disabilities have an attendance rate of 72.9%. This decreasing participation in the educational system is finally reflected in access to higher education, which is estimated at only 5.4% of the population with disabilities [3]. In conclusion, the illiteracy rate for the population with disabilities is three times higher (22.5%) than the national figure (7%), suggesting that actions should be focused on the population with disabilities between the ages of 15 and 24, which in absolute terms represents 475,000 people in 2010. [4]

However, Colombia does not have an accurate figure of persons with disabilities and specific access to education currently, despite having different measurement tools. One measurement tool is the 2018 National Population and Housing Census (CNPV by its acronym in Spanish), which provides broad and general information on the population's living conditions with disabilities. Another measurement tool is the Registry for the Location and Characterization of the Population with Disabilities (RLCPD), which has a broad set of variables and a robust information exchange system but has limited population coverage and does not allow to account for the overall situation in the country. [5]

According to CNPV, 2018 was evidenced of an increase of persons with disabilities of 19.4% concerning the General Census of 2005. Statistics show disabled people were 2'624,898 to 2005 [6], 6.3% of the total population in the country, and for 2018 disabled people were 3'134.036, 7.1% of the total population in the country [7]. While in RLCPD, only 1,448,889 disabled people had been identified and characterized to December 2018, 2.9% of the total population of the country projected by the National Administrative Department of Statistics (DANE) for that year [8], less than half of the population with disabilities in the country identified by the CNPV for 2018.

Regarding the educational aspect, the CNPV of 2018 indicates that the educational level reached by persons with some disability has been: in preschool education about 1.1%, in primary education 43.1%, in secondary education 12.6%, in middle education 16%, in higher education 11.3% and in postgraduate education 1.6%, which except for the case of primary education, correspond to lower percentages than the educational levels reached by persons without disabilities (about between 30% to 50% less); in addition, about 14.3% of persons with disabilities have not reached any educational level, which compared to persons without disabilities (3.6%) is four times higher, showing a higher educational lag in persons with disabilities. Likewise, the 2018 CNPV estimates that the degree of illiteracy for the population with some disability is still approximately three times higher (17%) compared to the national figure (6.1%), evidencing the need to continue designing and implementing actions focused on population with disabilities to improve their access and permanence in education. [7]

Thereby, the country has been adopting an educational policy based on a framework of the global education policy of education for all, policies led by UNESCO since the late twentieth century, giving way to what is known as Inclusive Education [9]. Inclusive education seeks to provide appropriate responses to a perspective focused on diversity and respect for difference, reducing barriers to learning and participation for all. The development of inclusive education implies a very high level of commitment and understanding on the whole educational community to achieve good practices, requiring teachers to focus their educational work on the students most vulnerable to the processes of exclusion. [10]

On the other hand, when analyzing the specific case of people with visual impairment in Colombia, it is evidenced that this disability has become the most frequent and the most affected the daily performance [7], for 2005 visually impairment were 1'139,205 people, equivalent to 43.4% of the population with disabilities in the country for that year [6]. For 2018, it was increasing visual impairment be 1'948,3332 people, equivalent to 62.17% of the population with disabilities in the country and 4.41% of the total Colombian population according to CNPV [11].

In addition, regarding the development of inclusive education for people with visual impairment, it is evident that one of the main mediating agents of inclusive education is teachers. As it is the one who designs, incorporates, and optimizes the teaching-learning process in the classroom through the contents and tools that he/she integrates to respond to the different needs and abilities of the students, encouraging the participation of all students in the process.

However, according to information reported by the National Institute for the Blind (INCI) in its strategic plan for 2015-2018, it reveals a strong influence of teachers in the causes that the visually impaired population to fail to realize the right to preschool, primary and secondary education, due to the lack of teacher training to expand professional skills in the development of inclusive teaching practices for the visually impaired population.[12]

For example, in those educational institutions that include children and young people with visual impairment (blindness and low vision) or multi-deficit in their classrooms, students have relevant pedagogical support in the regular classroom. However, teachers are not aware of using the Braille system, decreasing the ability to evaluate or rectify whether or not the student understands a text. This situation hinders how students learn, understand, and produce texts.[13]

The reading-writing process requires the accompaniment of teachers with knowledge in Braille language [14] in order to transcribe the texts. However, making these transcriptions is a time-consuming task for the educator [15], becoming the teaching-learning process ineffective for visually impaired people, preventing timely feedback.

For this reason, technological alternatives or innovations have been developed [16], which can be offered by institutions for people with visual impairment, to allow them to access literacy through the Braille reading and writing process [17], using some portable device [18][19] for both the reading and writing process.

In the reading process, the alphabetic text is transcribed into Braille [20]. In this case, any alphabetic text is transformed semi-automatically into Braille language so that it can be read by a visually impaired person, either through printer-type devices [21] [22] [23], digitizers with hardware that emulate Braille characters mechanically [24] or through tactile screens [25].

In the writing process, it is transcribed from Braille to alphabetic text. In this case, it is sought that the visually impaired person writes a text in Braille, and this can be validated by alphabetic text read by a person without visual impairment. Either it is using image processing techniques or other artificial intelligence techniques [26] [27], performing transcription using a PC or a Smart-Phone [28], or even stand-alone microcontroller type devices [29].

Therefore, this article proposes the performance evaluation of two applications that use OBR [15] (Optical Braille Recognition) techniques for the transcription of literary Braille into Spanish through a series of image acquisition and processing processes. The above is to contribute to the improvement of literacy and teaching-learning processes through technological tools that facilitate the work of teachers and timely feedback to students.

In the first place, the article describes each of the applications developed, the "Premier" application for mobile devices and the "Hiperión" application for personal computers, in terms of technical conditions and requirements, as well as the methodology for image acquisition and processing used for Braille character detection and transcription. Secondly, the performance analysis of each application is presented in terms of processing time, transcription error percentages (accuracy), and response to different test conditions. Finally, the results and conclusions regarding the comparison of the applications are presented, analyzing the versatility and robustness achieved in the detection of dots, cells, and Braille characters under different illumination characteristics and types of sheets.

2. DESCRIPTION APPLICATIONS

In order to understand and compare the specific characteristics of each of the literary Braille to Spanish transcription applications described below, it is first necessary to know some generalities of the Braille system.

2.1 Braille System

People with visual impairment use Braille as their primary reading-writing system and as an alternative means of communication to visual. Braille system can be understood as a tactile mechanical system with a guide on paper to perform a binary encoding. It uses cells of 6 dots in relief organized in rectangular blocks of two columns by three rows, known as generator symbol or generator sign, source of all Braille characters to represent an alphabet or any other type of representation of symbols, such as numbers, musical notes, chemical symbols, and even some pictographic method. [14]

Consequently, being a system similar to the binary and having 6 points in relief, making 64 combinations or groupings is possible, obtaining complete signography to represent different contents [30], which can be attributed to pure literary languages like the case of Spanish. The configuration of a cell or generator sign can be seen in figure 1.

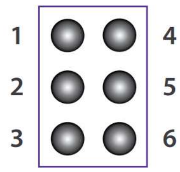


Figure 1: Generator Sign [31]

On the other hand, the cells having a relief can be perceived with the fingertips, which allows the visually impaired person to perform the reading process, which is done from left to right. For the manual writing process, which is done from right to left [32], a stylus with a certain thickness in its tip is used, a guide slate in making the reliefs on the paper, and a sheet of paper with specific texture characteristics and thickness [31]. These conditions of the paper must ensure that the Braille character resists contact with the fingertips and does not be easily erase what is written.

2.2 Mobile Device Application

The software application for a mobile device called "Premier" is an Android application on a smartphone [33]. It allows a person without visual impairment through a photograph taken to the text written in Braille system, obtain a transcribed text in Spanish, using OBR techniques to identify each of the cells and dots that make up a symbol or letter in Braille.

2.2.1 Initial technical considerations

When developing a software application on a smart device (phone or tablet), you must consider the structure of the operating system, Android, in this case. This OS is based on the Linux kernel and bases its structure on running applications programmed in JAVA, which generally can be run on any operating system. This style of architecture allows running software with acceptable performance and low power consumption.

The structure of an application in the Android operating system generates several layers or tiers [33], like this:

- the first uses the Linux kernel base to manage drivers, processes, and networks;
- the second, the layer of native Android libraries (libraries) written in C/C++ code, are accessed through the use of native JAVA interfaces (JNI);
- the third, the execution environment, is where the Dalvik - DVM virtual machine is located. It is optimized to reduce the memory used by applications and designed to run several instances of the virtual machine simultaneously, delegating memory management, threads, and SandBoxing (process isolation) to the underlying operating system. In the latest version of the Android operating system (Lollipop), Dalvik has been replaced by ART (Android Runtime);
- the fourth, the application framework, which houses the tools to manage the lifecycle of the application, resources, packages, sensors, and other add-ons necessary for the operation of applications;
- finally, the fifth or last one where much of the code of the applications created by the developers and the factory applications of the phone such as the dialer, contacts, browser, among others, is developed.

2.2.2 Software requirements

The development of the Premier application requires, from a software point of view, the previous installation of the OpenCV library, according to the structure of an application in the Android operating system. This library creates a common infrastructure for computer vision applications. It has more than 2500 algorithms optimized for several applications involving computer perception systems in commercial systems, which are growing every day.

It is important to note that this library is cross-platform and runs on Linux, Windows, Mac OS, and Android, of course. In addition, it can be implemented from different programming languages such as C/C++, Python, JAVA, and Matlab.

Additionally, OpenCV provides a set of samples or example source code, mainly written in JAVA, allowing initial handling of this library on a mobile device by executing simple tasks.

2.2.3 Image acquisition and preprocessing

In the process of acquiring the image of the Braille text to be transcribed by the Premier application, it is necessary to take into account the following aspects:

- The characteristics of the paper must be completely white bond paper, the paperweight of 90 to 120 grams, letter size, without additional content (ink, stains, wrinkles) that can generate noise;
- The conditions of the environment where the photo is taken should be well lit;
- The surface on which the sheet is placed should be a dark color to create contrast and allow the

Journal of Theoretical and Applied Information Technology
15th December 2021. Vol.99. No 23

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

application to detect the sheet's corners automatically.

The Premier application implements a graphical interface to load the images to transliterate as shown in figure 2, either from a file located in the multimedia gallery of the mobile device or directly from the device's camera; for this, it uses the algorithm presented in figure 3.

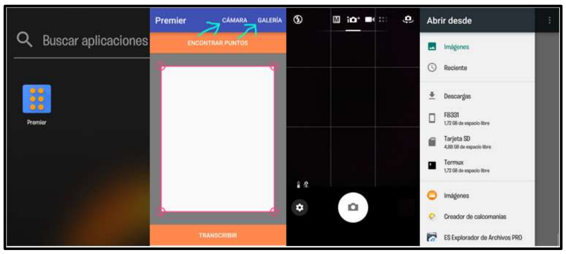


Figure 2: Premier Application [33]

```
Algorithm 1. Read Image Process
If select = 1
        Image_in = gallery
Else
        Image_in = open_camera
im = read_image.cv2(Image_in)
bitmap = im
```

Figure 3: Image Acquisition Algorithm

In addition, the application supports different image formats with a wide range of colors which are loaded into a bitmap variable, where the image is converted to a binary matrix with a value for each pixel in the different channels to operate. Thus, the image is loaded from the camera or gallery to manipulate the information using OpenCV libraries.

Subsequently, the image is read in RGB format, converted to grayscale, and the workspace is identified by applying a Canny filter for edge detection, thus recognizing the corners of the sheet using the algorithm shown in figure 4.

```
Algorithm 2. Edge detection
# Transform RBG file to gray scale image
gray_im = rbg2gray.cv2(im)
# Calculate the image borders with Canny filter
borders = Canny.cv2(gray_im)
# Iterative algorithm to detect the sheet corners
for columns
        for rows
                bordes(rows,colums) = min, max

        end
end
max,min = corners[2:2]
```

Figure 4: Edge and Corner Detection Algorithm

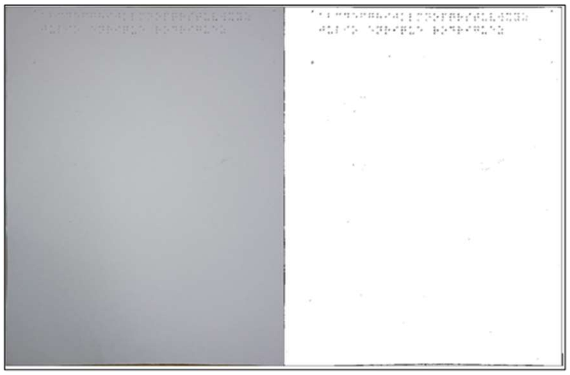
At this point, the application identifies exactly the working space (remember to take a picture on a dark surface), a gaussian filter is applied to the image, and thresholding is performed. If there is any problem with the algorithm mentioned above, then the corners of the sheet must be entered manually.

Note that when the application identifies the corners of the image to be analyzed, as shown in figure 5, it must ensure that the workspace is perfectly horizontal, i.e., aligned. For this, it must apply a perspective transformation method, using a homogeneous transformation matrix as shown in equation 1, which performs a transfer of the input image in angle and position, between the input matrix (tilted) and the output matrix (aligned) [34], this is known as a perspective shift.

Figure 5: Sheet Corner Identification[33]

$$\left.\begin{aligned} &(xa, ya) = (xr - g, yr) \\ &\textit{by analogy of triangles}: \\ &\frac{g}{yr} = \frac{ix}{1} \rightarrow g = ix \bullet yr \\ &\textit{Tilting at X amount ix}: \\ &R(x, y) \coloneqq A(x - ix \bullet y, y) \end{aligned}\right\} \quad (1)$$

A thresholding process is applied to the image resulting from the previous process, which has as input a grayscale bitmap and results in a binary matrix, numerical matrix with which it is easier to work. The thresholding process can be fixed or dynamic and applied to a sector of the image locally or the entire image, as illustrated in figure 6.



(a) (b)

Figure 6: (a) Grayscale image, (b) Thresholded image [33]

The thresholding performed by the Premier application is global thresholding according to equation 2, where b(r,c) is the intensity and t_0, and t_1 are the two possible gray levels of the thresholded pixel when the value t_0=0 and the value t_1=1 the image is said to be binarized.

$$b(r,c) = \begin{Bmatrix} t_0 sil(r,c) < T \\ t_1 sil(r,c) \geq T \end{Bmatrix} \quad (2)$$

Once the application has identified the edges and has performed the thresholding of the image, it performs a process to remove any noise that may contain the sheet and carefully adjust the edges, making trimming of the corners that do not fit.

2.2.4 Character detection and transcription

The Premier application implements a character detection process to guarantee that the Braille dots detected in the image acquisition and preprocessing process do not correspond to any noise to proceed with the transcription process of the Braille text into Spanish, avoiding possible errors.

2.2.4.1 Braille dot detection - Hough transform

Through a native function of OpenCV called HoughCircles, the Premier application detects or recognizes the Braille dots[34] that resemble in some percentage to a circumference. These circles are drawn on the original image in another color. Thus, the user can verify possible circles that do not correspond to the Braille text, a product of noise defects by light or problems in the preprocessing of the image. In this way, possible errors are removed. An example of the above is shown in figure 7.

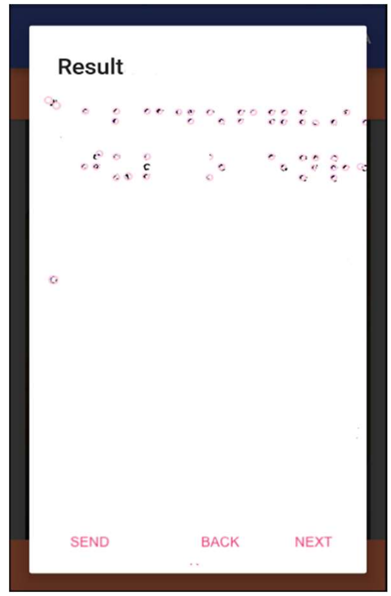


Figure 7: Point Recognition with Hough [33]

It is essential to clarify that, when working on a mobile device, it can only have an image in latent memory within the bitmap; for this reason, every time a process is performed, it must be stored in the device's internal memory.

2.2.4.2 Image segmentation by workspaces (Braille slate)

Subsequently, the Premier application establishes the limits of the workspace in the image, which is achieved by identifying four points that correspond to the guides generated by the Braille slate; these points or guides are designed to locate and spatially guide the person writes a text in Braille.

Once these points have been identified, the application displays them on the screen. Then the user can verify that the area used by the slate has been correctly located and identify the workspace by making a cutout of the image, as shown in figure 8.

Through the guide points and considering the dimensions of the slate, the application allows to identify the following parameters:

- Number of slates per page (each slate has specific dimensions and a certain number of Braille characters)
- Maximum number of lines per sheet (24 for letter size sheet)
- Number of characters per line (28)

Figure 8: Work area [33]

Up to this point in the image processing, the bitmap has been converted to a binary matrix. The Premier application uses that matrix later to count the number of pixels in black, which corresponds to the number of Braille dots on the sheet. Moreover, thus compare them with the four dots of the guide to verify the degree of occupancy of writing on the sheet; this occupancy degree is measured in used slates by the visually impaired person.

Once the Premier application identifies how many slates are used and the guide points (the guide points define the working space), it segments the original image into smaller spaces that enclose the pixels of each character; this is called "working cell," as shown in figure 9.

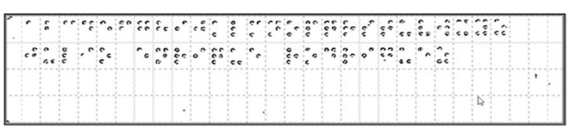


Figure 9: Identification of each of the work cells [33]

The development of this application detected that the Braille cells have an inherent error due to their position when performing the exact arithmetic division of a work cell in the six spaces. Because some characters are out of phase and the fractionation generates errors in identifying the Braille dots, as shown in figure 10. That is the reason the application detects the contours and counts the positions of the dots. Besides, it performs the image segmentation according to the detected contour (image "b") if the image has more than 35% of its area corresponding to dots; otherwise, the segmentation of the image is left according to the original cell (image "a").

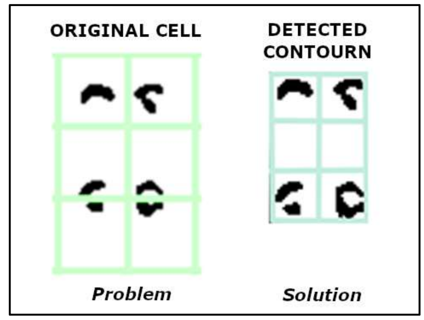


Figure 10: Image segmentation. Image "a" (left), Image "b" (right) [33]

After the Premier application analyzes each of the cells, a test series are run to verify the operation in the most complex letters of the Braille alphabet to Spanish. Figure 11 illustrates that transcription using the dotted outline criterion.

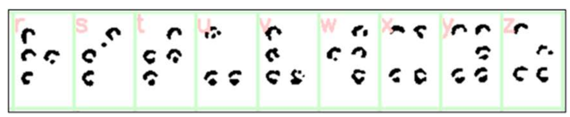


Figure 11: Text transcribed into Spanish over image of Braille symbols [33]

Finally, when the application has the information of each of the cells, it makes a tour of the slates to generate a string of characters which can be sent in a simple text string. Considering that the application is developed on a mobile device, it generates the text string with the option to be sent via SMS or email.

2.3 Personal Computer Application

The computer software application called "Hiperión" is a tool that transcribes a digitized image of a Braille text into Spanish text [35], using different image filtering techniques, according to the OBR techniques used in the application. This application allows people without visual impairment (teachers) to immediately verify the texts written in Braille by people with visual impairment (students) in different pedagogical activities, contributing to the educational process of reading and writing for the teaching-learning of the Braille system.

2.3.1 Initial technical considerations

When developing software or applications for PC, it is necessary to consider RAM size and the required disk space.

Journal of Theoretical and Applied Information Technology
15th December 2021. Vol.99. No 23

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

In the case of the Hiperión software application, RAM size is less, as it only requires 4 gigabytes and a minimum available disk space of 1 gigabyte to install the necessary libraries. It was developed in Matlab using an academic campus license of the Universidad Distrital Francisco José de Caldas.

The application has a "Front End" designed with the help of the Matlab Guide that offers a series of tools that favor the design of the graphical interface and a "Back End" that relies on the libraries for image processing and computational power for matrix work that Matlab offers.

2.3.2 Software requirements

The development of the Hiperión application requires a set of tools that cover all the image processing and contribute to satisfactory and high-performance developments; for this, it uses the Imtool library of Matlab, which is compatible with Linux and Windows, making this software application multiplatform.

2.3.3 Image acquisition and preprocessing

The image acquisition process of a Braille text in the Hiperión application is carried out through a digital scanner configured for a specific resolution (300ppm). That allows a more sophisticated image preprocessing, admitting texts coming from different sources (sheets of different thickness) with different conditions (sheets of various relief and environmental noise).

This stage incorporated results from previous studies on preprocessing, using a digital scanner as an acquisition tool [34] [36]. Because of that, the Hiperion application supports writing on white sheets of the recommended thickness for Braille writing and thinner sheets like those of a notebook with a graph paper, offering a feature of versatility to this software application.

In the preprocessing of the digitized image, the Hiperion application performs filtering processes, which contribute to removing environmental noise due to the illumination applied by the scanner in the acquisition process and removing possible defects of the sheet, dirt, or according to the type of sheet, its grid. All this is in order to facilitate a correct detection of the Braille characters.

Then, it executes computational techniques for image processing, such as the logarithmic function that helps lighten the dark image and increase its contrast. Also, it uses techniques to rotate and align the image correctly to facilitate accurate Braille lines detection.

2.3.4 Character detection and transcription

2.3.4.1 Image segmentation

The Hiperion application performs a symmetrical tracing on the previously processed image to establish the lines of each of the slates and to separate the work cells. Thus, reach the correct segmentation of a character and the space of each of them, as shown in figure 12.

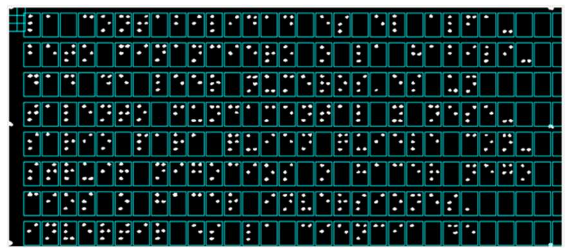


Figure 12: Segmented white sheet[35]

2.3.4.2 Transcript

Once the image has been segmented, the Hiperión application reads each of the Braille characters and transcribes them into Spanish, and then performs a visualization process in the graphic interface so that the reader of the text in Spanish (teacher) can verify what the visually impaired person wrote in the beginning. Figure 13 shows a flow chart that summarizes each of the stages executed by the application.

In figure 14, it can see the result of the transcription of a Braille text, where a format is showed with the number of Braille characters detected and the number of dots.

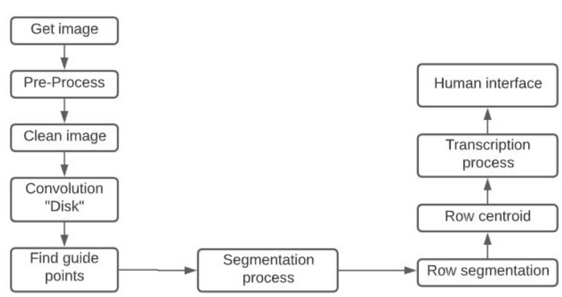


Figure 13: Flowchart

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

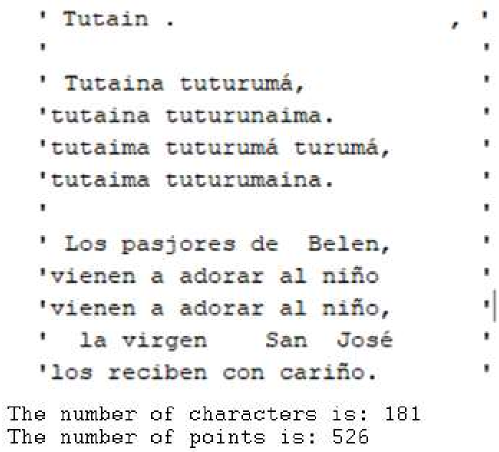


Figure 14: Braille Graph Paper Sheet Test [35]

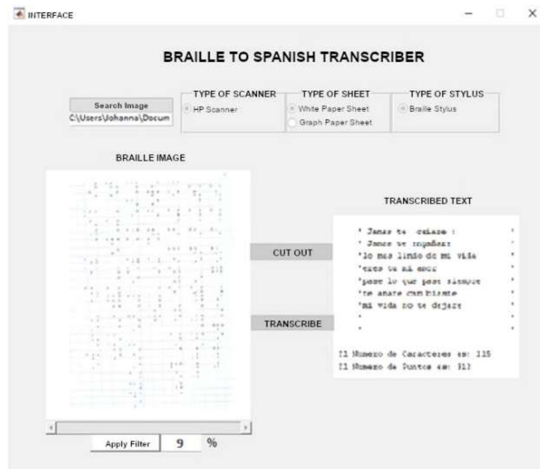


Figure 16: Interface in operation [35]

2.3.5 Graphical user interface

The design of the graphic interface of the Hiperión software application allows the user to easily and quickly search for the image and use the options corresponding to the type of scanner used, the type of stylus, and the sheet on which the Braille text is written, as shown in figure 15.

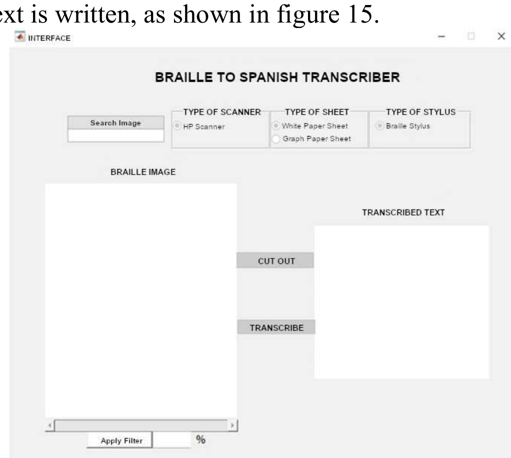


Figure 15: Graphical interface [35]

When loading a Braille text sheet through the scanner, the graphical interface will look like figure 16.

3. PERFORMANCE ANALYSIS

The performance analysis of each software application for the transcription of literary Braille to Spanish is based on the specific characteristics described in the previous section for mobile devices and personal computers.

3.1 Mobile Device Application

In the case of the Premier application developed for mobile devices, tests were performed with the Android Profiler tool. It determines the minimum requirements for the exact CPU load, internal memory, and RAM size when the application runs on a mobile device with an Android operating system. These tests were performed on mobile devices with different hardware features and camera resolution, as specified in table 1.

According to the results obtained, it was determined that a smartphone-type mobile device only requires 190 Mbytes of internal memory available for the installation of the application and only 300 Mbytes of RAM for its correct operation. In terms of processing, the application does not generate an excessive load on the CPU.

On the other hand, it should be noted that the success of transcription using a mobile device such as a smartphone depends mainly on the resolution and lighting conditions with which the photograph of the Braille text is taken. Thus, preliminary tests were carried out, using only the flash of the cameras of the devices described in table 1, showing that the light is insufficient.

Table 1: Characteristics of Mobile Devices Used in Testing

Device	Model	Processor	RAM	Operating System	Rear Camera Resolution
1	SONY F8331	Snapdragon 820, Quad-core 2.2GHz, 64-bit	3GB	Android 7.1.1	23 MP
2	HUAWEI Y5 (MYA-L23)	MediaTek MT6737T, Quad-core 1.4GHz, ARM-A53 64-bit	2GB	Android 6.0	8 MP
3	LG G3 BEAT D722P	Snapdragon 400, Quad-core 1.2GHz, 32-bit	1GB	Android 5.1	13 MP
4	HUAWEI Y360-U03	MediaTek MT6582, Quad-core 1.2GHz, Cortex-A7 32-bit	500MB	Android 4.3.1	5 MP

Therefore, it is necessary to generate an environment with controlled light to acquire the image of the Braille text to be transcribed through the Premier application, using at least a couple of LED lamps, so that no shadows are generated on the sheet of paper containing the text.

Additionally, the effectiveness of the application in transcription was evaluated, taking into account the number of Braille characters that a sheet could have, measuring the time it takes the application to process a line, a character, and even a single dot.

Table 2 summarizes the processing times measured according to the mobile devices used in the tests described above.

According to the results obtained, it is evident that the processing time per page does not exceed two seconds because the OpenCV libraries used by the Premier application are optimized for the Android operating system because they are native to Linux. In addition, this style of architecture allows running software with acceptable performance and low power consumption.

Table 2: Measuring Mobile Devices Processing Times

Device	2896 Points	1 Character	1 Line	662 Characters
1	170 ms	1,922 ms	61,792 ms	1487 ms
2	410 ms	2,285 ms	73,264 ms	1752 ms
3	650 ms	2,481 ms	79,392 ms	1902 ms
4	720 ms	2,644 ms	84,292 ms	2026 ms

3.2 Personal Computer Application

The case of the Hiperión application developed for personal computers allows the handling of images and matrices easily because it had been implemented using the licensed software Matlab. That makes it possible to perform in a simple way tasks such as rotate and align the image acquired by a scanner; it also allows the user to choose the type of slate used and not only a slate of 4 lines. On the other hand, the software also detects subtle changes in the holes, so it can be adjusted to detect Braille characters for any of the two types of the stylus with which a visually impaired person can write a text.

On the other hand, a series of tests were performed on the application in order to test the effectiveness of the image preprocessing algorithm, varying the acquisition resolution of the Braille text images to be transcribed, using different scanners, so that the algorithm adapts to the different resolutions and sizes acquired. Therefore, the algorithm has the robustness to handle different variations: possibilities of the stylus, size of the slate, and its ability to support various resolutions of image acquisition devices. However, this feature increases the application's response time, which, in the case of a sheet filled with 16 lines, can take more than a minute.

Additionally, it can be pointed out that one of the most relevant characteristics of the Hiperion application is its capacity to carry out the transcription process on different types of sheets in terms of their texture, thickness, and background (graph paper sheets and white sheets). Thus, a series of tests are carried out to demonstrate the effectiveness of the application according to sheet type. It takes as a reference parameter the percentage of detection of dots in the Braille characters (accuracy) or percentage of error for each type of sheet processed.

Table 3 summarizes the tests performed, specifying the maximum and average error percentage obtained per sheet type. Note that at least 15 tests with each type of sheet are necessary to tabulate the error percentages.

All tests were performed on a personal computer with the following specifications: Core I5-6200U@2.3 GHz processor, 8 Gigabytes of RAM, and a 1 Terabyte mechanical disk.

Table 3: Percentage error or accuracy by sheet type

	White Sheets	Graph Paper Sheets
Maximum Error	4%	4%
Average Error	2.55%	1.61%

According to the above, the following average processing times were obtained:

- 6,2 seconds for image insertion and preprocessing
- 35,4 seconds in image rotation
- 10,1 seconds in segmentation and transcription of the sheet

4. CONCLUSIONS

The performance of the two applications that execute transcription from Braille to Spanish text was evaluated. It was found that the minimum requirements for proper operation of the application for mobile devices (smartphone) are 190 Mbytes of memory available for installation and only 300 Mbytes of RAM, and in the case of the application for PC are 300 Mbytes of memory space on a hard disk and 500 MBytes of RAM. Additionally, satisfactory results and processing times were obtained without excessively loading the CPU when testing the two tools on mid-range computers.

It was evidenced that the two applications have similar characteristics in their performance. However, the application for mobile devices has been developed using OpenCV and free software, while the application for PC has been developed with the licensed software Matlab(R).

It was found that the most important parameter in the evaluation of the performance of the two applications is the accuracy, that is, the percentage of detection of dots in the Braille characters or percentage of error, since this defines the correct functioning of the transcription from Braille to Spanish text. In the case of the "Premier" and "Hiperión" applications, at least 30 tests were executed, with changes in lighting, type of stylus, type of sheet, among other aspects. In both cases, the accuracy exceeds 96% effectiveness, is comparable to those presented in another research on the same topic [26]; that means for a typical Braille writing sheet with an average of 110 words, the transcription only would have three words wrong.

One of the significant contributions of this research was the work done with the OpenCV libraries for the Android operating system used in the "Premier" application. It was found that this application for mobile devices supports only white sheets, and it has a response time in the transcription of a few seconds, resulting in a high impact for any class of image processing application on mobile devices. In addition, this application facilitates its possible use by teachers of primary and secondary education in Colombia, who have easy access to such devices to support the development of their classes with students with visual impairment.

On the other hand, another contribution of the research is that the PC application lets choose some parameters to image acquisition of Braille text in terms of materials used by visual impairment. The application supports different Braille slates, stylus, and, most importantly, white sheets and gridded and lined sheets, the most used by people with visual impairment in Colombia. The use of a different class of sheets, especially gridded or lined sheets, is because of the low economic capacity of visual impairment, which forces them to acquire low-budget items for their daily use.

5. LIMITATIONS AND FUTURE RESEARCH

Taking into account the information presented by Colombian governmental entities in [4,5,14,15], it can be evidenced that people with visual impairment have difficulties in access to education, leading in many cases to low academic performance, in addition to not having access to adequate study materials, a situation that pushes these people to use in the case of Braille writing sheets of any type, of any thickness, even with imperfections and/or elements of dirt, or possibly previously printed, which in the case of the applications developed for the transcription of Braille text to Spanish becomes a technical limitation that makes it necessary to make a hardware development to improve the process of acquiring information from Braille text, which is why future research is proposed where the design of a series of sensors or methodologies to capture the Braille dots regardless of the type of sheet, punch or that the sheet has any class of imperfection, and that can be used for both mobile devices and personal computers.

On the other hand, as a next step in the research, it is proposed to conduct field tests with visual impairment of different educational levels and rural and urban sectors. Those tests will continue to contribute to the improvement of literacy and teaching-learning processes through technological tools that facilitate the work of teachers and timely feedback to students in Colombia.

6. ACKNOWLEDGEMENTS

This work was supported by the Universidad Distrital Francisco José de Caldas, in part through CIDC, and partly by the Facultad Tecnológica. The views expressed in this paper are not necessarily endorsed by the Universidad Distrital. The authors would like to thank the research group Embedded Computer Security -SIE-, Lina M. Noguera R., Julio E. Rodríguez P., Jhon J. Benítez N., and Sandra J. Vargas C., for the evaluation of the prototype ideas and strategies.

REFERENCES

[1] C. M. P. la Educación and J. Walker, “La Educación Inclusiva para niños con Discapacidad,” *Campaña Mundial por la educación*. pp. 1–36, 2013. [Online]. Available: http://www.campaignforeducation.org/docs/reports/Equal

[2] A. Sarmiento Gomez, “Situacion de la educacion en Colombia Preescolar, basica, media y superior,” Educacion Compromiso de todos, Bogota D.C., 2011.

[3] M. de Salud de Colombia, “Boletín Observatorio Nacional De Discapacidad.” pp. 26–56, 2014. [Online]. Available: http://discapacidadcolombia.com/index.php/inclusion-educativa/196-orientaciones-generales-para-la-atencion-educativa-de-las-poblaciones-con-discapacidad-en-el-marco-del-derecho-a-la-educacion

[4] M. de Educación Nacional de Colombia, “Orientaciones Generales para la atención educativa de las poblaciones con discapacidad en el marco del derecho a la educación.” p. 80, 2012. [Online]. Available: http://www.insor.gov.co/home/wp-content/uploads/filebase/orientaciones_poblacion_discapacidad_2012.pdf

[5] P. de la República, “Discapacidad en la primera infancia: una realidad incierta en Colombia,” *De Cero a Siempre*, vol. 5. p. 32, 2013.

[6] M. de Salud y Proteccion Social, “Sala situacional de las personas con discapacidad,” *Ministerio de salud y proteccion social*. pp. 28–30, 2019. [Online]. Available: https://www.minsalud.gov.co/sites/rid/Lists/BibliotecaDigital/RIDE/VS/MET/sala-situacional-discapacidad2019-2-vf.pdf

[7] DANE, “Funcionamiento humano- Censo Nacional de Población y Vivienda,” *Departamento Administrativo Nacional de Estadistica*. 2019. [Online]. Available: https://www.dane.gov.co/files/censo2018/informacion-tecnica/cnpv-2018-presentacion-funcionamiento-humano.pdf

[8] Minsalud, “Sala situacional de las personas con discapacidad,” *Ministerio de salud y proteccion social*, 2020, [Online]. Available: https://www.minsalud.gov.co/sites/rid/Lists/BibliotecaDigital/RIDE/VS/MET/sala-situacional-discapacidad2019-2-vf.pdf

[9] M. de Educación Nacional de Colombia, “Educación inclusiva.” p. 1, 2018. [Online]. Available: https://www.mineducacion.gov.co/1759/w3-article-374740.html?_noredirect=1

[10] L. Vélez-Latorre and D. Manjarrés-Carrizalez, “La educación de los sujetos con discapacidad en Colombia: abordajes históricos, teóricos e investigativos en el contexto mundial y latinoamericano,” *Revista Colombiana de Educacion*, vol. 1, no. 78, pp. 253–298, 2020, doi: 10.17227/rce.num78-9902.

[11] M. de Educación Nacional de Colombia and C. P. Dussan, “Los ciegos en el Censo 2018,” *106*.

p. 1, 2018. [Online]. Available: https://www.inci.gov.co/blog/los-ciegos-en-el-censo-2018

[12] P. Y. Chaves Leguízamo, "Diseño de un Programa Piloto de Formación a Maestros para la Educación Inclusiva de Población con Discapacidad Visual: Una Experiencia desde la Perspectiva Fonoaudiológica," 2018.

[13] S. M. Sánchez-Medina and E. Díaz-Sánchez, "Discapacidad Visual, Sistema Braille e Inclusión Educativa desde la Perspectiva Interseccional," *Revista Gestión I+D*, vol. 5, no. 2, pp. 33–59, Feb. 2020.

[14] A. Marcet, M. Jiménez, and M. Perea, "Why braille reading is important and how to study it," *Cultura y Educacion*, vol. 28, no. 4, pp. 811–825, 2016, doi: 10.1080/11356405.2016.1230295.

[15] L. Nahar, R. Sulaiman, and A. Jaafar, "'Bangla Braille learning application' in smartphones for visually impaired students in Bangladesh," *Interactive Learning Environments*, vol. 0, no. 0, pp. 1–14, 2019, doi: 10.1080/10494820.2019.1619588.

[16] R. H. Zatserkovnyi, V. Z. Mayik, R. S. Zatserkovna, and L. Ya. Mayik, "Analysis of Braille translation software," *Printing and Publishing*, vol. 2, no. 78, pp. 36–44, 2019, doi: 10.32403/0554-4866-2019-2-78-36-44.

[17] S. Kumari, A. Akole, P. Angnani, Y. Bhamare, and Z. Naikwadi, "Enhanced braille display use of OCR and solenoid to improve text to braille conversion," *2020 International Conference for Emerging Technology, INCET 2020*, no. June, pp. 0–5, 2020, doi: 10.1109/INCET49848.2020.9153996.

[18] N. R. J. Hynes, D. J. J. Jebaraj, J. S. Kumar, S. Immanuvel, and R. Sankaranarayanan, "Portable electronic braille devices - An overview," *AIP Conference Proceedings*, vol. 2142, no. August, 2019, doi: 10.1063/1.5122531.

[19] K. Kulkarni, A. Mahajan, Y. Zambre, F. Belwadi, S. Killedar, and A. Marathe, "Text Detection and Communicator Using Braille for Assistance to Visually Impaired," *2019 IEEE Pune Section International Conference, PuneCon 2019*, pp. 1–5, 2019, doi: 10.1109/PuneCon46936.2019.9105829.

[20] S. Hossain *et al.*, "Text to Braille Scanner with Ultra Low Cost Refreshable Braille Display," *GHTC 2018 - IEEE Global Humanitarian Technology Conference, Proceedings*, pp. 1–6, 2019, doi: 10.1109/GHTC.2018.8601552.

[21] J. L. dela Cruz, J. A. D. Ebreo, and R. A. J. Inovejas, "Development of a Text to Braille Interpreter for Printed Documents through Optical Image Processing," *IEEE Access*, pp. 1–6, 2017, doi: 978-1-5386-0912-5/17/31.00.

[22] A. Moise, G. Bucur, and C. Popescu, "Automatic system for text to braille conversion," *Proceedings of the 9th International Conference on Electronics, Computers and Artificial Intelligence, ECAI 2017*, vol. 2017-Janua, pp. 1–6, 2017, doi: 10.1109/ECAI.2017.8166391.

[23] O. D. S. Fonseka and N. Wedasinghe, "-Blind Draw-A software solution for image identification and artistic skills for visually impaired people using braille," *2018 3rd International Conference on Information Technology Research, ICITR 2018*, pp. 1–6, 2018, doi: 10.1109/ICITR.2018.8736137.

[24] S. Sultana, A. Rahman, F. H. Chowdhury, and H. U. Zaman, "A novel Braille pad with dual text-to-Braille and Braille-to-text capabilities with an integrated LCD display," *2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies, ICICICT 2017*, vol. 2018-Janua, pp. 195–200, 2018, doi: 10.1109/ICICICT1.2017.8342559.

[25] J. Siqueira *et al.*, "BrailleÉcran: A Braille Approach to Text Entry on Smartphones," in *Proceedings - International Computer Software and Applications Conference*, Aug. 2016, vol. 2, pp. 608–609. doi: 10.1109/COMPSAC.2016.5.

[26] K. Smelyakov, A. Chupryna, D. Yeremenko, A. Sakhon, and V. Polezhai, "Braille Character Recognition Based on Neural Networks," *Proceedings of the 2018 IEEE 2nd International Conference on Data Stream Mining and Processing, DSMP 2018*, pp. 509–513, 2018, doi: 10.1109/DSMP.2018.8478615.

[27] H. Montiel A., F. H. Martínez S., and E. Jacinto G., "Training Strategy for a Neural Network Using a Modified Genetic Algorithm," *JATIT & LLS*, vol. 98, no. 13, pp. 2606–2615, Jul. 2020.

[28] K. S. Wibawa and K. D. Rushjayanti, "Implementation of Electronic Braille Alphabet Connected with Android Smartphone Device," vol. 11, no. 12, pp. 268–272, 2019.

[29] C. Avendaño Mejia and E. Villa Freddy, "Diseño e implementación de un dispositivo

electrónico interactivo para aprendizaje de Pre-Braille orientado a personas con discapacidad visual," p. 18, 2019.

[30] Ismael. Martínez Liébana and Delfina. Polo Chacón, *Guía didáctica para la lectoescritura braille*. ONCE, Dirección de Educación, 2004.

[31] INCI, "INTERACTUANDO CON EL BRAILLE Orientaciones Generales para la Enseñanza del Sistema de Lectoescritura Braille." pp. 1–39, Mar. 2020.

[32] O. P. Loza Peñaloza, "Sistema electrónico braille para la ayuda en el aprendizaje de personas no videntes," *Universitas, Revista de Ciencias Sociales y Humanas*, vol. 7, pp. 209–225, 2006.

[33] Li. M. Noguera Rojas and J. E. Rodriguez Prieto, "Aplicación móvil para Transcribir Lenguaje Braille Literario a Texto En Español," 2018.

[34] E. Jacinto Gómez, H. Montiel Ariza, and F. H. Martínez Sarmiento, "Circle Hough transform implementation for dots recognition in braille cells," *Eighth International Conference on Graphic and Image Processing (ICGIP 2016)*, vol. 10225, no. Icgip 2016, p. 102250N, 2017, doi: 10.1117/12.2266991.

[35] J. J. Benítez Niño and S. J. Vargas Caro, "Transcriptor de Sistema Braille a Texto Español con Técnicas Computacionales Estándar," 2020.

[36] E. M. Suárez Molina, M. Á. Barrera Pérez, and E. Jacinto Gómez, "Literary Braille language translator to Spanish text," *2016 IEEE International Conference on Automatica, ICA-ACCA 2016*, 2016, doi: 10.1109/ICA-ACCA.2016.7778514.