IMAGE CAPTURE SYSTEM IN CLASSROOM BOARDS A TOOL FOR THE LEARNING PROCESS OF PEOPLE WITH HEARING OR MOTOR IMPAIRMENT

Jessica V. Beltrán C., Joann S. Forero M. and Julián R. Camargo L.
Faculty of Engineering, Universidad Distrital Francisco José de Caldas, Bogotá D. C., Colombia
E-Mail: jcamargo@udistrital.edu.co

ABSTRACT

This article describes the implementation of an electronic device capable of taking images written on boards. These images are stored, processed, and edited. This process extracts the necessary information to reproduce digitally the text on the boards. A Single Board Computer (SBC) Raspberry Pi was used, allowing the capture, rendering, and editing of the images. The images were obtained using the micro camera made for Raspberry Pi, which uses serial communication to capture images.

Keywords: disabilities, image capture, inclusion, raspberry Pi, rendering images, tesseract.

1. INTRODUCTION

Population with disabilities are 6.4% of Colombia inhabitants, this means 2'632.255 people in Colombia have some kind of disability [1].

Even though there are specialized teaching institutes, and there are inclusion policies, disabled people have several recurrent problems with education access, at all education levels. These problems are related to the lack of either tools, platforms, or specialized educational programs. As a result of this, more than 48% of disabled people in Colombia haven't access to middle school. [1].

Nowadays, disability is seen as an integral condition, which is the main reason to create strategies that focus on increasing the abilities of each person. For this purpose, schooling needs to enhance education programs to become inclusive [2].

This investigation proposes the design and implementation of a device able to recognize characters written on a surface, to subsequently, process, digitalize, and compile those images in a digital format. This will allow people with hearing or motor impairment to have a record of their classes to complete their studies.

Technological advances represent a huge source of pedagogical strategies for disabled people, who need low-cost and easy access tools that contribute to their social development.

This investigation uses technological devices that will help with the academic interaction of people with hearing or motor impairment. The purpose is to obtain a higher percentage of children and young Colombians, with some kind of hearing or motor disability, get access to high-level education.

The device works on the recognition of written characters written on a board, and transforming them into digital text.

The student will be able to keep a record of his class by taking pictures of the text written on the board and being processed and converted to a .txt file afterward. This strategy will contribute to the academic inclusion of the disabled population.

All the said beforehand, it can be done using a low-cost, high performance, minicomputer called Raspberry Pi, which allows capture, edit, and process images [3].

Raspberry Pi comes with free software (Raspbian) [3], which is an advantage over cost. The code was made in Payton this platform was chosen due to its high performance.

The methodology used in this project consist of the following parts:

- Digitalization. Acquisition of the image taken from the board.
- Image processing
- Character recognition
- Words identification

2. DIGITALIZATION

The image acquisition was made by the Raspberry Pi camera module, which can record videos with high definition, and still photographs as well.

Among the main limitations of the photographs or a sequence of them, are the following: i) they only represent what can be recorded, given the light conditions and others; ii) they can only show what is within the range of vision and what is not restricted by objects in the foreground.

The Raspberry's camera module can be used to take high definition video as well as still pictures. It is easy to use for beginners but has a lot to offer to advanced users. You can also use the libraries that I pack with the camera to create effects. The code implemented helped the device to take pictures whenever the teacher wanted it.

It has a five-megapixel fixed-focus camera that supports 1080p30, 720p60 and VGA 90 video modes, as well as still image capture. It is attached via a 6-inch flat cable to the CSI port on the Raspberry Pi. It can be accessed through the MMAL and V4L APIs, and there are numerous third-party libraries built for this purpose, including the Picamera Python library [2].

The acquisition of images will be off-line, this acquisition model works with representations in luminance space, usually, the data are recorded by equipment capable of capturing the data of interest through images (cameras, scanners and others). This method requires a pre-processing of the image to eliminate the noise introduced by the medium and by the capture device. Being of great importance to binary the image and find the area of each character to be isolated and simplify the recognition process.

The acquisition of the image will always have problems of dependence on the style of writing, which varies according to the person, calligraphy and mood of the writer. These variations are of utmost importance since the characters from one individual to another vary significantly due to social or regional influences. Creating a writer's independence is an arduous task that requires a high degree of training as well as a robust database that stores the characteristic writing patterns of each of the alphabet characters of a large number of individuals.

3. IMAGE PROCESSING

Figure-1 shows the proposed methodology. This process offers satisfying results without sacrificing processing time. It should be mentioned, that there are some combinations in which similar results can be obtained, but the time taken by the Raspberry Pi in processing the image is higher than 10 minutes.

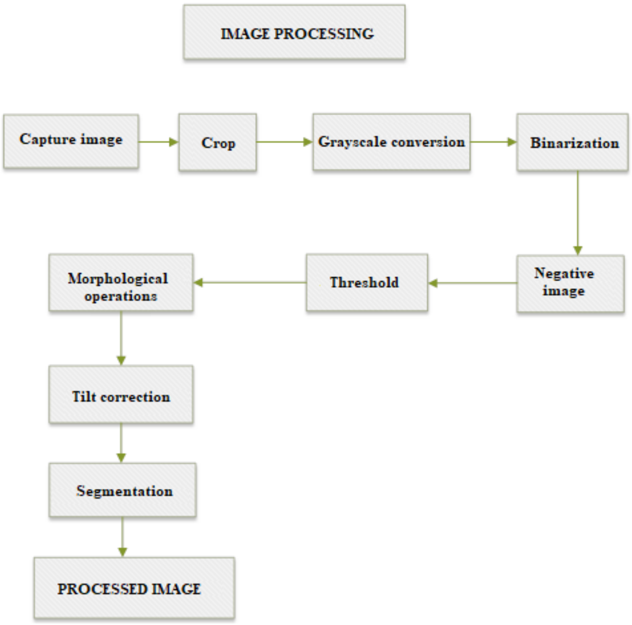


Figure-1. Image processing.

When these results are transferred to the software used, Tesseract- OCR, there is no significant difference and the results obtained have the same range of accuracy.

3.1 Image Processing

The image needs to be cropped, before the processing, to reduce the size of the image, and consequently obtain a reduction in the processing time.

Working with real-sized boards requires a predefined cut and the camera must be at a specific distance. Therefore, it is possible to define a cutting range of the size of the board. This step eliminates irrelevant information. This technique is simple and doesn’t add time or sources to the processing time [4-7].

3.2 Grayscale

The image is cropped and transformed into grayscale to reduce the information in it. This step is necessary to produce the threshold using the local characteristics method [8, 9].

3.3 Histogram Equalization

The histogram equalization method can be used in images that don’t have a uniform luminance value, or a reduced grayscale [8, 9].

3.4 Threshold and Negative Image

Before doing the pixel threshold, all the luminance values of the images are inverted, which means, replacing the actual value for the complement. Theoretically, applying the pixel threshold to an inverted image, or a grayscale image, should have the same results, but in practice, using the negative image before the threshold gives better results. [4, 10].

In this investigation, the method used was the pixel adaptative threshold, because the defined binarization threshold presented problems when it was used in images with low luminance.

Some problems seen using this method were the loss of information and differences in areas between characters [8, 10].

3.5 Image’s Inclination Correction

The image’s inclination correction is done after the pixel adaptative threshold.

Even though the Raspberry Pi camera is placed in a horizontal position, some boards have an inclination that needs to be fixed. Some classrooms where the test was taken, at the Universidad Distrital Francisco José de Caldas, had a considerable inclination. Additionally, some people tend to write in a non-straight line.

The deskew filter is used to fix the inclination. This filter does a horizontal line that is compared with the shape of the image. When the inclination is found, the image is cropped and rotated automatically.

3.6 Object Segmentation by Area

As a final step, it is necessary to eliminate figures or noise that passed through in the threshold process. Segmentation is the process of selection if a character is part of the image or just noise, according to the area of each object [6, 11, 12].

This technique offers better results to the process because it eliminates the characters that are not necessary and leaves the ones that are. On the other hand, this technique has some limitations. Sometimes the noise has the same area of the characters, and noises with an area bigger than the set one can't be eliminated.

4. CHARACTERS RECOGNITION

Different tests were made to find all the possible errors in the process. Each test was done and each error found helped to create a satisfactory process language for the character's recognition. The process is described down below.

4.1 Tesseract OCR

The Optic Character Recognition (OCR) Tesseract was used to recognize the written characters [13, 14]. Tesseract is a free software, developed by Hewlett Packard, and subsequently, bought and managed by Google. This software is based on neuronal networks, and it is considered one of the most accurate free software.

The software has an acceptable execution time, but it presents problems with the hand-writing text. It can be used in Linux, Windows, and Mac OS. Version 3 of the software supports formatted text and character analysis on the page.

Tesseract can process characters in English, French, Italian, Dutch, Portuguese, German, and can be trained to work with other languages [13, 14]. This software uses OCR's technology to identify characters in a printed text.

Figure-2 shows the operating steps of the Tesseract, which are listed below:

- Analyze and save shapes
- Gathering contours in Blobs forms
- Blobs are organized in text lines
- The text lines are divided into words.
- Recognition of each word
- Recognized words pass to the adaptation trainer.
- The lessons learned by the adaptative trainer are used in the following step.
- Try to recognize the words that weren't recognized in the previous steps.
- Recognize and solved small-caps and blurry spaces.
- Produce the digital text

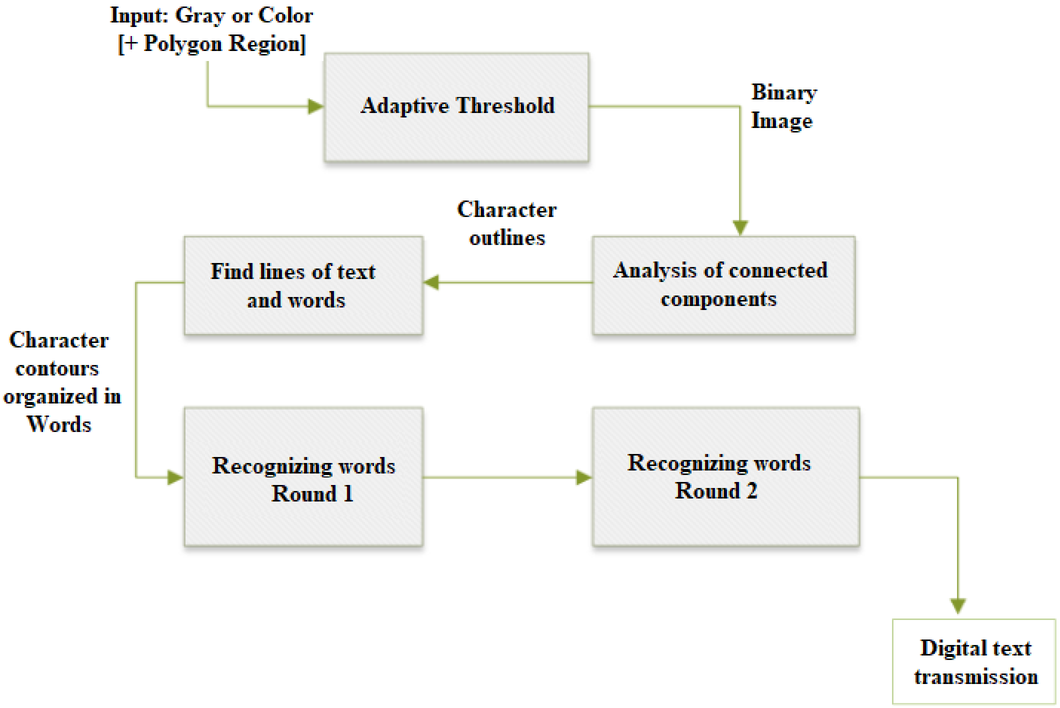


Figure-2. Topology of TESSERACT - OCR.

The first step of the process is the analysis of the device where the information is going to be safe. Tesseract was the first software in used white over black text. In this stage of the process, all the information is saved by nesting patterns in Blobs (Binary Large Objects) [12].

Blobs are organized in text lines, that are going to be analyzed by similar size and text. The text lines are divided into word shapes depending on the characters' spaces, using defined spaces or diffused spaces.

The recognition of the text is done in two steps. First, each word is recognized individually. If all the words were recognized, these words pass to an adaptative classifier. This classifier recognizes the text with higher accuracy [4].

As a subsequence step, the previous process is done again, so, if any word wasn't recognized previously, it can be recognized in this new analysis

In the final step of the process, all the blurry spaces and x-high positions are analyzed, just in case, there's a capital letter with a lower case size [15, 16].

4.2 Training

Tesseract OCR is a software that recognizes digital text from an image, so, to be able to recognize written characters, it was necessary to train the software and create a database with the information needed.

5. RESULTS

Five thousand characters and four different trainers were analyzed in this investigation. At the beginning of the process, the investigation used the trainer by default of the Tesseract, but this tool only was able to recognize 32% of the images.

Table-1 shows the results obtained using the Tesseract tool.

Table-1. Recognizing the percentage of the Tesseract tool without training.

% Recognized Lowercase	**56,33**
% Recognized Capitals	78,98
% Total	58,88
% recognized Total	63,62
No. of characters	591

The recognition percentage improved to 58.88% after using algorithms to improve the image quality, eliminate the noise, and not-desired information.

Table-2 shows the recognition percentage of the Tesseract tool after training.

Table-2. Recognition percentage of the Tesseract tool after training.

% Recognized Lowercase	97,16
% Recognized Capitals	99,22
% Total	95,94
% recognized Total	98,05
No. of characters	1207

This investigation found that Tesseract has problems recognizing the letters g,j,l,q,r,s. Without training the percentage of recognition of those characters is just 40%. To avoid mistakes in the segmentation and recognition of those characters it was necessary to define special characteristics for each letter.

Figure-3 and Figure-4 show the characteristics above mentioned.

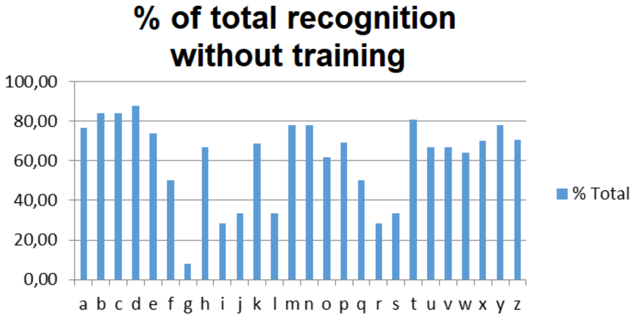


Figure-3. Total recognition without training.

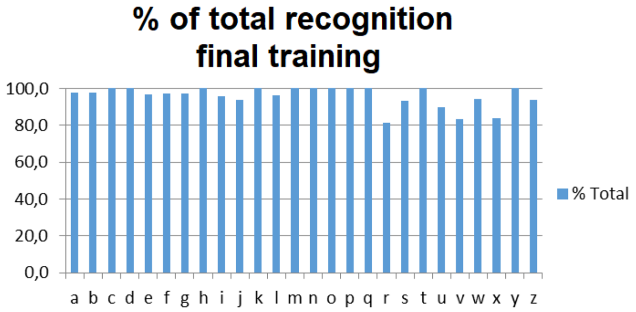


Figure-4. Total recognition with training.

Using final training, with the parameters mentioned before, the recognition percentage improved to 95.94%. The device enhanced the recognition accuracy by 37% after training. One thousand and seven characters were used in this test.

Figure-5, Figure-6, and Figure-7 show a recognition problem with the letter t. In the first segmentation, this letter was changed by "1*".

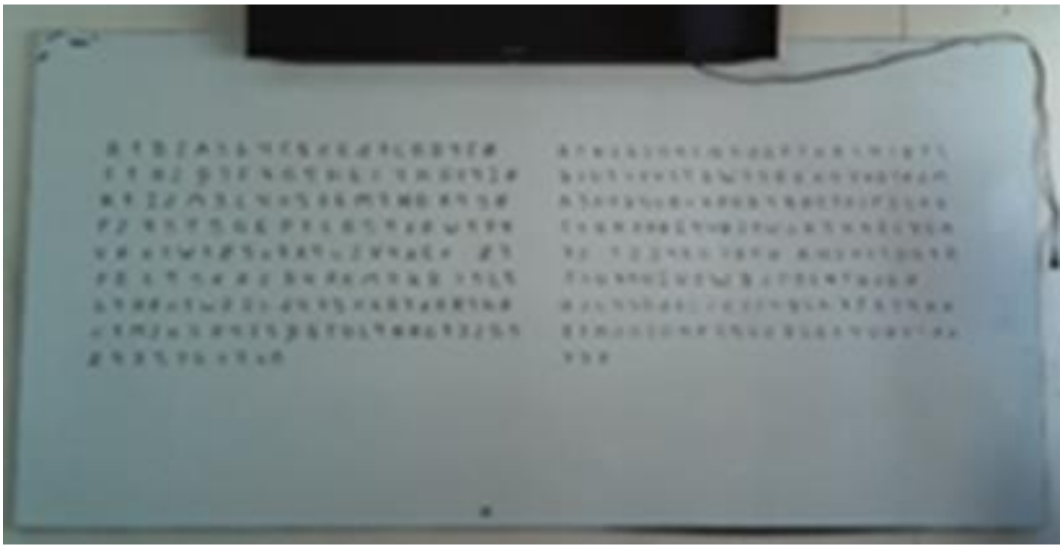


Figure-5. Capture 1, Original image.

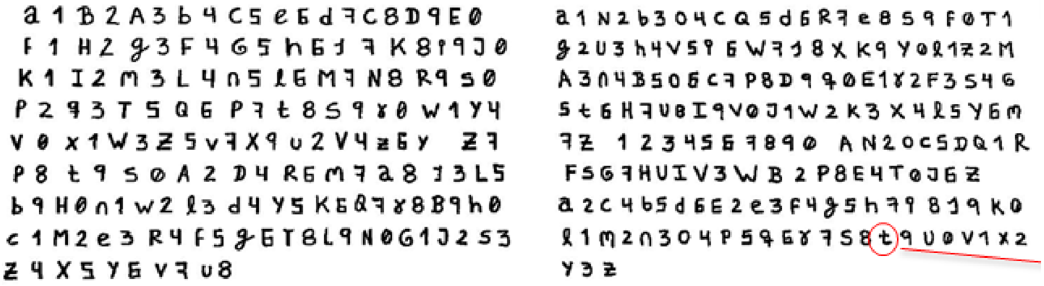
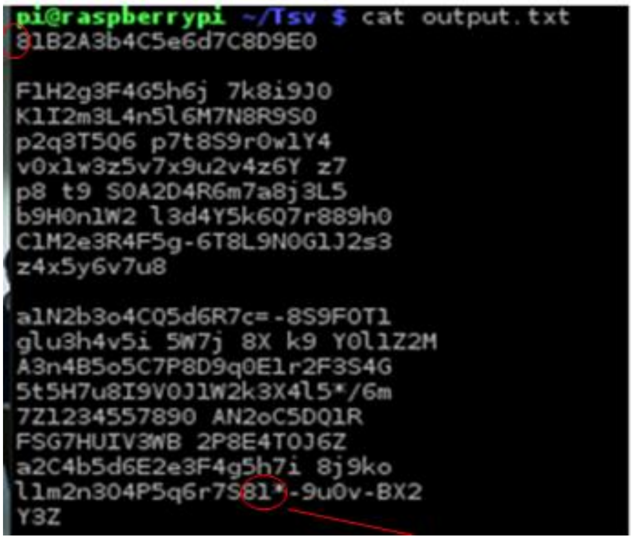


Figure-6. Processed image.



```
pi@raspberrypi ~/Tsv $ cat output.txt
81B2A3b4C5e6d7C8D9E0

F1H2g3F4G5h6j 7k8i9J0
K1I2m3L4n5l6M7N8R9S0
p2q3T5Q6 p7t8S9r0w1Y4
v0x1w3z5v7x9u2v4z6Y z7
p8 t9 S0A2D4R6m7a8j3L5
b9H0n1W2 l3d4Y5k6Q7r889h0
C1M2e3R4F5g-6T8L9N0G1J2s3
z4x5y6v7u8

a1N2b3o4CQ5d6R7c=-8S9F0T1
glu3h4v5i 5W7j 8X k9 Y0l1Z2M
A3n4B5o5C7P8D9q0E1r2F3S4G
5t5H7u8I9V0J1W2k3X4l5*/6m
7Z1234557890 AN2oC5DQ1R
FSG7HUIV3WB 2P8E4T0J6Z
a2C4b5d6E2e3F4g5h7i 8j9ko
l1m2n304P5q6r7S81*-9u0v-BX2
Y3Z
```

Figure-7. Final character recognition.

This investigation also found that if the text has a significant space between words, the recognition software is going to place that word in a new column.

6. CONCLUSIONS

Tesseract is a specialized software, that is used on the Linux platform and it presents better processing times than other OCR software.

Different images processing technics exist nowadays, each one with different kinds of quality, resolutions, and processing times, so it is important to try different combinations of these techniques and find which combination presents better results. The Raspberry Pi doesn't present the highest processing capacity.

To eliminate noise and too soft the shapes, the opening and closing technique presents suitable results. However, this technique presents a loss of information that can be relevant in the final result.

It isn't possible to achieve 100% accuracy in image recognition. Things like the writer's calligraphy, basic characteristics of the character, the database of the trainer, etc. can affect the final results.

ACKNOWLEDGMENTS

The authors would like to thank the Universidad Distrital Francisco José de Caldas and the LASER research group that supported the development and testing of the project.

REFERENCES

[1] Departamento Administrativo Nacional de Estadística (DANE). 2004. Información estadística de la discapacidad. Bogotá - Colombia, Julio de 2004.

[2] P. Sarmiento. 2008. Personas con discapacidad y acceso a servicios educativos en Latinoamérica. CERMI.ES Comité Español de representantes de personas con discapacidad. Quito- Ecuador, Análisis de situación 2008.

[3] Raspberry Pi. 2020. Raspi-config. [Online]. Available: http://www.raspberrypi.org/documentation/configuration/raspi-config.md. [Accessed December 2020].

[4] V. Papavassiliou, T. Stafylakis, V. Katsouros and G. Carayannis. 2010. Handwritten document image segmentation into text lines and words. Pattern Recognition. 43: 369 - 377.

[5] R. Bindal, P. Kumar, G. Kaur and G. Dhiman. 2019. An Approach for Automatic Recognition System for Indian Vehicles Numbers Using K-Nearest Neighbours and Decision Tree Classifier. International Journal of Advanced Science and Technology. 28(9): 477-492.

[6] S. Du, M. Ibrahim, M. Shehata and W. Badawy. 2013. “Automatic License Plate Recognition (ALPR): A State-of-the-Art Review. IEEE Trans. Circuits Syst. VideoTechnol. 23(2): 311-325.

[7] K. Kousalya, S. Kapilan, K.Karthik, M. P. Benny and S. Santhiya. 2020. Number plate detection using deep neural network. International Journal of Advanced Science and Technology. 29(3s): 937-954.

[8] J. M. Castillo. 2012. La composición de la imagen del renacimiento al 3D. España: Paraninfo.

[9] V. Gopan, K., L. D. Wins and A. Surendran. 2018. Image Processing Based Tailor-Made Software Package for the Condition Monitoring of Grinding Wheel. International Journal of Mechanical and Production Engineering Research and Development (IJMPERD). 8(3): 825-834.

[10] R. Khamdamov and E. A. Saliyev. 2018. Approaches to Image Processing Using the Tools of Fuzzy Sets. International Journal of Computer Science Engineering and Information Technology Research (IJCSEITR). 8(1): 1-12.

[11] Universidad Nacional de Quilmes. 2005. Segmentación por Umbralización. Universidad Nacional de Quilmes, Ingeniería en Automatización y Control Industrial.

[12] L. Likforman-Sulem, A. Zahour and B. Taconet. 2006. Text Line Segmentation of Historical Documents: a Survey. International Journal on Document Analysis and Recognition. 9(5): 1-25.

[13] Git Hub. 2020. Tesseract-ocr. [Online]. Available: https://github.com/tesseract-ocr/. [Accessed December 2020].

[14] G. Abhishek, M. Srihari, P. Sudheer and M. Saravanan. 2019. Hybrid Colour Infrared Image Edge detection using RGB-YCbCr image Fusion. International Journal of Advanced Science and Technology. 28(19): 101-108.

[15] Bangor University. 2008. An overview of the Tesseract OCR (optical character recognition) engine, and its possible enhancement for use in Wales in a pre-competitive research stage. Language Technologies Unit (Canolfan Bedwyr), Bangor University, 2008.

[16] R. Smith. 2007. An Overview of the Tesseract OCR Engine. Ninth International Conference on Document Analysis and Recognition (ICDAR 2007). 2: 629-633.