Research Homework 4

1st Ericka Céspedes Moya
Tecnológico de Costa Rica
San José, Costa Rica
ericka.cespedes@gmail.com

2nd Esteban Alonso González Matamoros
Tecnológico de Costa Rica
San José, Costa Rica
esteb.gonza29@gmail.com

***Abstract*—The following document contains the first and second part of the Research Homework 4 for the course of Introduction to Pattern Recognition. The first part of this homework required the students to process handwritten vowels in order to analyze it in the second part. The second part involved the making of histograms based off of the latter pre-processed handwritten vowels to recognize each vowel.**

***Index Terms*—image processing, pattern, recognition, Python, OpenCV**

I. Results

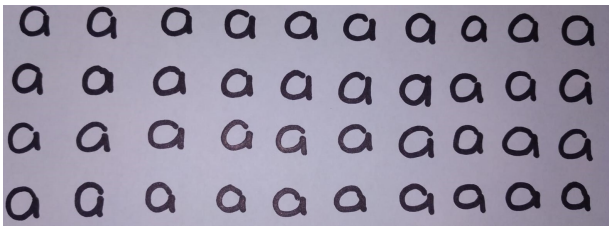

Fig. 1. Original image

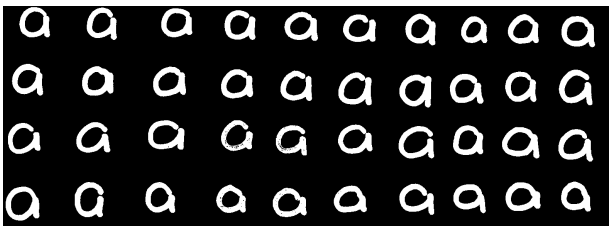

Fig. 2. Tresholding

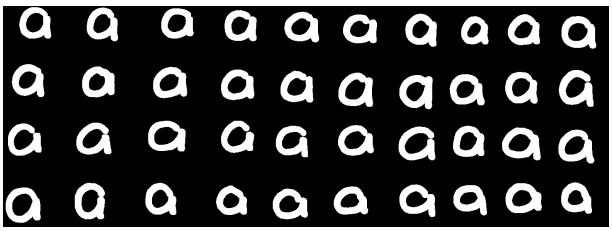

Fig. 3. Dilation

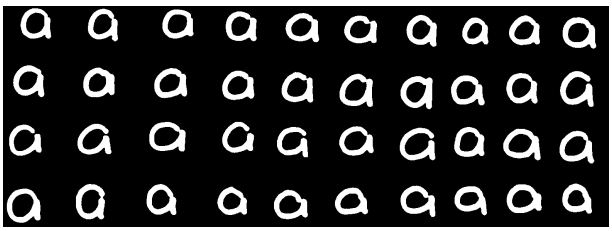

Fig. 4. Erosion

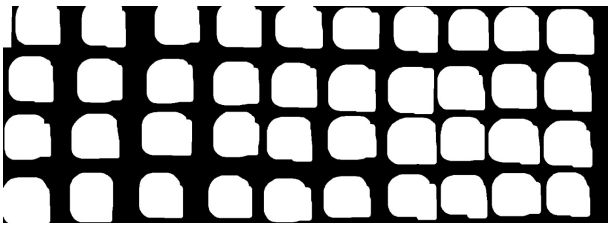

Fig. 5. Bad Dilation

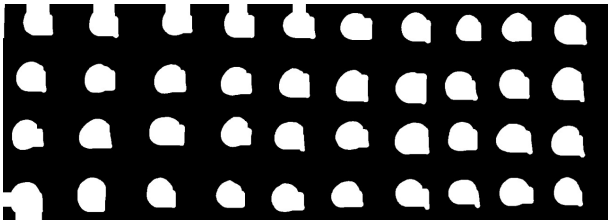

Fig. 6. Bad Erosion

Fig. 7. Cropped a by Daniel

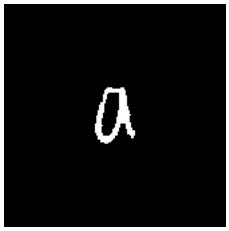

Fig. 8. Cropped a by Darío

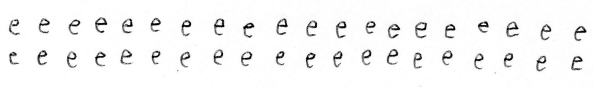

Fig. 9. Original es of Darío

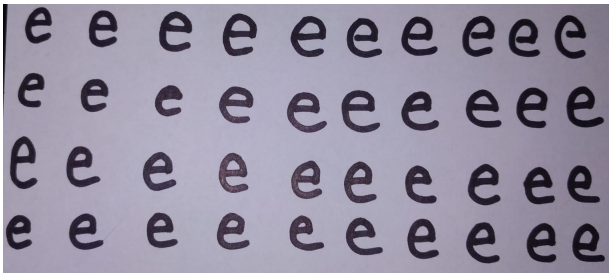

Fig. 10. Original es of Daniel

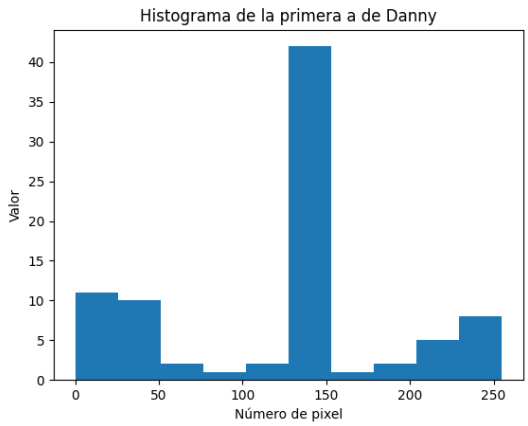

Fig. 11. Histogram of one of Daniel's As

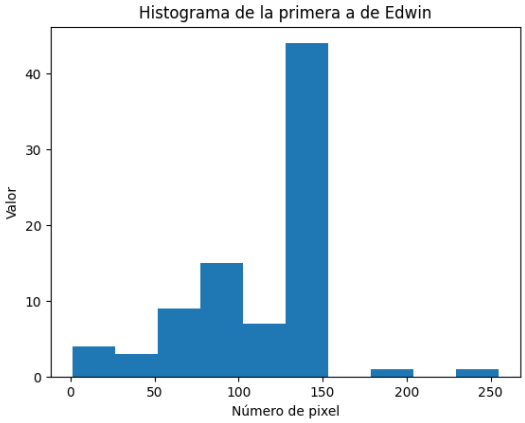

Fig. 12. Histogram of one of Edwin's As

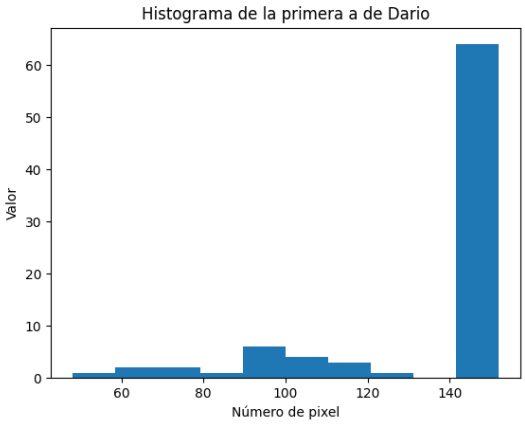

Fig. 13. Histogram of one of Dario's As

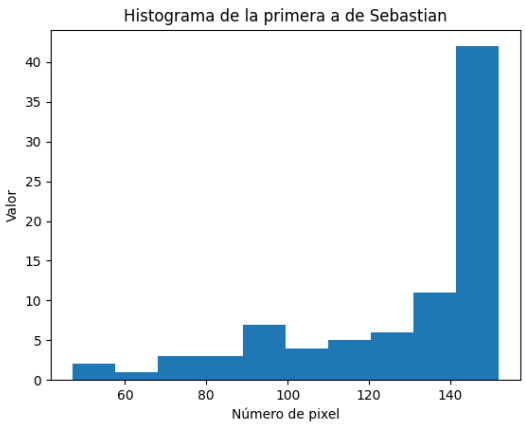

Fig. 14. Histogram of one of Sebastian's As

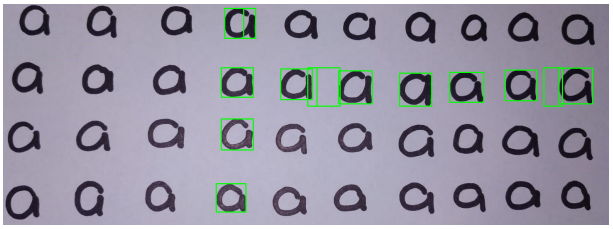

Fig. 15. PyTesseract Recognizing Handwritten Letters

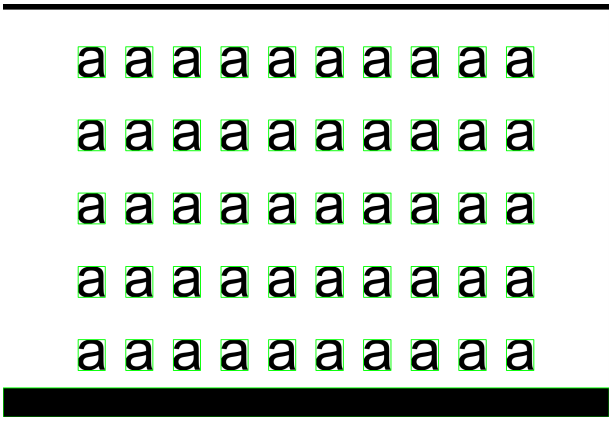

Fig. 16. PyTesseract Recognizing Computed Letters

TABLE I
HIT PERCENTAGE BY VOWEL AND PATTERN RECOGNIZER

Vowel	Hit Percentage	
	Self-made Recognizer	Already Existing Recognizer
a	42.7083	0.0
e	40.9091	0.0
i	60.0000	1.2780
o	55.6701	0.0
u	31.2500	0.0

II. ANALYSIS

Figure 1 is the original image taken by Daniel of handwritten as. First, the image is subjected to thresholding [1]–[4] to remove noise and make it easier to work with: figure 2. Then, since a black background and white borders were chosen, the image is first dilated (figure 3) and then eroded (figure 4). If the background were white and the letter borders black, then it would first be eroded and later dilated. A kernel of 2x2 was chosen in this case, if a bigger kernel is used, the image borders will get extremely thick and distorted, which is not the expected result. For instance, a kernel of 20x20 was used first, but the results were not as expected: figure 5and figure 6.

After the image is dilated and eroded correctly to suppress noise, we look for the contours of the image. When said contours are found, the image is cropped according to the parameters of the contours. Then, each cropped image is added to a black background with a certain height and width for all letters, the chosen size was 166x166 for all since this was the maximum for all. In this black background, the cropped image is centered so all are placed adequately: figure 7 and figure 8. For example, in the latter figures, Danny's a is very large while Darío's is very small, but they are set to the same size in order to get an average height and width for each vowel and person.

The most difficult part about recognizing handwritten letters is that each person writes in a different average size, and if the images are not taken with the adequate lightning and in a standard environment, then it is going to be extremely complicated to set a standard for each letter. We decided to set a minimum height and width for each person in order to not get noise when an image was cropped. For example, figure 9 has a tiny dark line in the left border. This is very difficult to cut given that it is slant and not straight. Then, the character recognizer would recognize that straight line as a border and letter. Then, figure 10, which are Darío's vowels, are very small compared to Daniel's. This was not the only case, but it is a very noticeable one. Therefore, average minimum heights and widths for each person were set.

Optical character recognition systems have been widely used to provide automated text entry into computerized systems. Yet in all this time, conventional OCR systems (like zonal OCR) have never overcome their inability to read more than a handful of type fonts and page formats. [5] As seen in figure 15, recognizes some but not all of the handwritten letters while it recognizes all of figure 16.

Handwritten letters histogram are fortuitous. They follow a certain pattern, but according to the letter and person. For example, figures 11 and 12 are very similar while figures 13 and 14 are different than the latter but similar to each other. Due to these differences and the large amount of data, the variance and mean measurements are generated very similar to those of other vowels. Due to these differences and the large amount of data, the variance and mean measurements are generated very similar to those of other vowels. This causes problems to identify each vowel, making all the vowels have the same probability by small differences.

The results of each recognizer were as shown in Table I. The self-made recognizer had a higher hit percentage than the already existing one. The already existing one had trouble recognizing the handwritten letters and then identifying them as the correct vowel. The only vowel identified was the i with 1.2780% of hits. In all the recognizers, the i was the easiest vowel to identify due to its difference in shape from the others. The second most-recognized vowel was the o. In third place, was the a while fourth place was the e. Finally, the last-recognized vowel was the u. The self-made recognizer had a hit rate above 30%, if the vowel u is ignored, 40%. The highest percentage was the i with 60% correct answers.

III. CONCLUSIONS

An image must be first applied thresholding, then dilation and erosion in order to suppress noise. If the background is black and the borders white then dilation first and erosion second. If the opposite, then vise versa.

It is crucial that each image's photo is taken in similar environments in order to get the best results. Otherwise, it is extremely difficult to suppress noise and set a standard for each vowel.

The most difficult part about recognizing handwritten letters is that each person writes in a different average size, and if the images are not taken with the adequate lightning and in a standard environment, then it is going to be extremely complicated to set a standard for each letter.

The existing written character recognizers in the free software community work very well for letters written by a computer, but not by a person.

Better photographs must be taken in order to identify the vowels more accurately. Now, it is easier to identify which vowel belongs to whom because each person writes in their own accord. Some vowels could be categorized as the same like figures 11 and 12, which are similar to each other being the same vowel and others such as figures 13 and 14 put in a different category to get better results.

REFERENCES

[1] A. Rosebrock, "Thresholding: Simple image segmentation using opencv," *Py Image Search*, 2014. [Online]. Available: https://www.pyimagesearch.com/2014/09/08/thresholding-simple-image-segmentation-using-opencv/

[2] K. Hong, "Image thresholding and segmentation," *BogoToBogo*. [Online]. Available: https://bit.ly/3DzDXHv

[3] AnandhJagadeesan, "Text detection and extraction using opencv and ocr," *Geeks for Geeks*, 2021. [Online]. Available: https://www.geeksforgeeks.org/python-thresholding-techniques-using-opencv-set-1-simple-thresholding/

[4] rishabhsingh1304, "Python — thresholding techniques using opencv — set-1 (simple thresholding)," *Geeks for Geeks*, 2019. [Online]. Available: https://www.geeksforgeeks.org/python-thresholding-techniques-using-opencv-set-1-simple-thresholding/

[5] F. Zelic and A. Sable, "A comprehensive guide to ocr with tesseract, opencv and python," *Nanonets*, 2021. [Online]. Available: https://nanonets.com/blog/ocr-with-tesseract/