Tarea de Investigación 4: Reconocedor estadístico básico de dígitos escritos a mano con el uso de histogramas como vector de entrada

Ignacio Barquero García, Jose Daniel Camacho, Jose Pablo Feng, Paul Villafuerte Beita

This investigation's objective is to teach the reader one way to implement an Optic Character Recognition (OCR) system. Mind that there are a lot more alternatives to achieve the same and with much more precision and effectiveness, for instance: Machine Learning. The alternative presented here was chosen due to its relative "simplicity" because no complex probability and statistics analysis or methods are used. The algorithm processes beforehand handwritten characters in order to get the set of information to do the predictions (in this case, the histograms). The algorithm will "guess" a written character by calculating how much difference is there between the histogram kept by the algorithm as reference and the histogram generated from the written character.

***Index Terms*—Histogram, OCR, Optic Character Recognition**

I. Introduction

One of the most popular applications of machine learning algorithms is the recognition of characters. In order for these types of algorithms to be able to identify characters, probability and statistics analysis and techniques are required. Additionally, matrix and vectorial operations are also involved, leading to the inclusion of linear algebra. The idea of these types of algorithm is to look at an image and analyze its pixels' information. Each individual pixel from an image tells 3 things, the x, y coordinates plus its RGB channels' values or its luminosity level in the case of gray-scale images. Machine learning algorithms take this data and use a vector where its values change and "learn" by analyzing images, after "enough" learning, the algorithm should have the capacity of "guessing" whatever image is used as input. The former is called the training of the algorithm and the latter, the evaluation or testing of the algorithm. This project presents a way for "guessing" handwritten digits through a rather simple approach, using the histograms of images for the aforementioned objective.

II. Approach

A. Image Cropping

1) Finding horizontal limits for cropping

Numbers are placed in a (not perfect) grid, so, the first step to get all the digits of that grid as individual images, is finding the beginning and the end of each row of digits. The chosen approach, is iterate from the top to the bottom of the image in the 'Y' axis, and then, find the upper pixel that suggests the begin of the row. Then, the algorithm repeats a similar process, but now, it will find the end of the row. Both points (start and end of the row), are saved into a list. On 1 we can see in red the marks made by the algorithm, indicating the crop points on the 'Y' axis.

2) Finding vertical limits for cropping

The idea of finding vertical limits for the cropping it's similar and it's used after cropping the horizontal limits, so this function will get the rows that have the numbers cropped in the horizontal part, and then will look for the limits of pixels in the right and left sides for each number to crop the digits on those limits. After this process, the digits may have some white horizontal lines that need to be cropped again, leading to the execution of a horizontal direction snip.

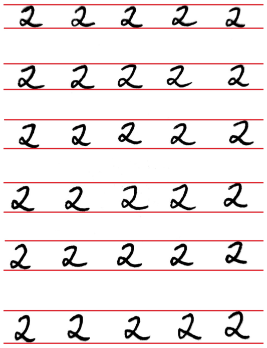


Fig. 1. 'Y' axis cropping marks

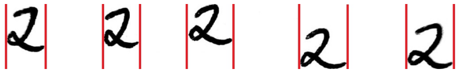


Fig. 2. 'X' axis cropping marks

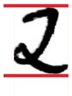


Fig. 3. Second 'Y' axis cropping marks

B. Training

1) Image resize and binarization

To create the histograms it's important to have in this case, all the arrays with the same size, so it can be modeled with pre-processed values that can be related between them. After this process, it's also necessary the binarization of the image so that the histogram creation could be easier and this will improve the prediction of the digits.

2) Training: Generating the histograms

Once the images have been cropped and resized to the standardized dimensions, the next step is to generate the algorithms "experience", the histograms. One of the project

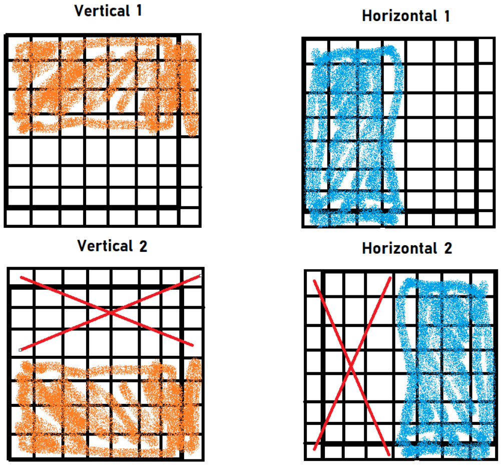


Fig. 4. 4 pixel wide/high

requirements was to build the histograms by horizontal and vertical orientations where columns and rows were of 4 pixels high/wide and adding them together after. We go by ”scanning” every row with a 4-pixel-high scanner and go 4 lines at a time without repeating. Column-wise, a 4-pixel-wide scanner is used, also without repeating columns. (Figure 4)

The ”scanning” process consists of traversing the rows and columns in the 4-pixel-wide-rectangle formation described previously, counting the amount of black pixels and keeping record of them in order to plot the histogram. Due to the process being executed for the horizontal and vertical orientations, to represent the entirety of the image we ”add” them together by appending the results (the resulting vectors) of one orientation with the other one. The idea can be depicted in 5. Mind that it can be done interchangeably the first time, but need to stick to the chosen order in order to calculate the difference with the input images. Finally, we got the vector with the values necessary to plot our histograms. This process is done for each number from zero to ten and for each training image supplied. The final vector used by the algorithm will be the average of all the histograms made for each number.

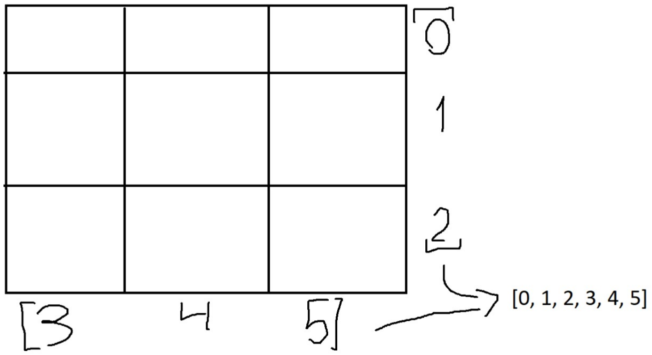


Fig. 5. Append of vectors

3) *Training: Mean and Variance*

After getting the histograms for each digit the idea is to get all the histograms for that number and create a mean matrix for each one of them and the variance of that matrix. This arrays will be saved in a list that will contain a list of the mean values of the image that will be use to compare the histogram of the images to be predicted.

III. Results

Looking at the charts and report logs, increasing the ”window” size for the ”scanning”, the average accuracy increases but presents some sort of *diminishing returns effect*. Analyzing the results, the accuracy score increases when the ”window” size is 1 until it reaches a size of 8, from there on, the score starts to decrease. The average accuracy for a ”window” of size 1 was 0.63333, from there on the average score increases until it reaches a ”window” of size 8 where the peak accuracy is reached with a value of 0.72499. From here onward the average score shows a negative trend. For example, the next three windows 9, 10, and 11 show values of 0.70555, 0.6556, 0.5389, respectively.

Now, looking at the plots [V-B1] of the comparison of histograms (with window size = 4), the input image’s histogram when compared to the reference histograms of the algorithmn differ slightly, for this reason the algorithm is able to predict most of the numbers. When the algorithm ”guesses” wrong, an observation can be done regarding the correct histogram compared to the one the algorithm chose as the ”correct” one, and it is that there are parts in the histogram that are of the same shape, for example, a curve or a ”plateau” but this one is lower, higher, or have a different scale in the same place between both histrograms causing the algorithm to ”confuse” itself and choose the wrong one.

IV. Conclusion

The hole idea that was implemented in this process to try to predict the digits that we write in the lab for this task, it’s very related to the process that is made in neural networks to predict different images, at least in the general idea in what it had to be made to ”train” the algorithm.

The idea of working in image processing and preprocessing of the image it’s heavy for the device that it’s executing the program, but it’s strictly necessary to facilitate the algorithm to predict in this case the digits. As we can see in the results, digits can be predicted with the idea that was implemented, but there is a big difference between a neural network training an the algorithm that was implemented in this paper, neural network may last longer in the training process but the algorithm for training will have a better accuracy, this said, it’s important to understand that the algorithm that it’s implemented in the paper would make the predictions in some cases but, the implementation of neural network despite the complexity that this may have would have a better result.

V. Other methods

A. *Character Classification using Logistic Regression*

1) ***Description***

The logistic regression algorithm is a very popular tool for dealing with binary classification problems. The algorithm estimates the correlation between multiple independent variables with a binary dependent variable. The sigmoid function is a synonym of logistic regression. When a value is strictly greater than 0.5 the value maps to 1, otherwise, it maps to 0. The values 1 and 0 can be equivalent to a ”yes” or ”no”, respectively. [2]

The sklearn library provides the Logistic Regression methods, plus has the capability of solving multivariate problems besides binary classification problems. [3]

The steps to follow in order to solve a supervised multivariate classification problem is as follows:

1) Load the training images and pre-process them as needed.
2) Reshape the image data as a one-dimensional array and adding an additionaly element which is the label or classification it belongs to. (A way of doing this is naming the training images with a certain format and parse them programatically to extract the label)
3) Once al specimens are preprocessed and loaded, separate the image data in an X array and parallely create another array Y with all labels without modifying the order. In other words, the indices in X and the labels in Y must match the original dataset.
4) Insert a 1 to every row in the data set, this will represent the intercept of the dataset, also called ”bias”.
5) Once the dataset is separated into the X and Y subsets, split the data in train data and test data. The amount of specimens for training and testing is arbitrary. A good proportion is 80% of the X dataset with its corresponding Y part as training and 20% as the testing subset.
6) Input the training dataset to the model object instance and test it with the testing data subset to measure its effectiveness.
7) If the model yields a good success rate/accuracy, the model is prepared to classify ”unseen” data

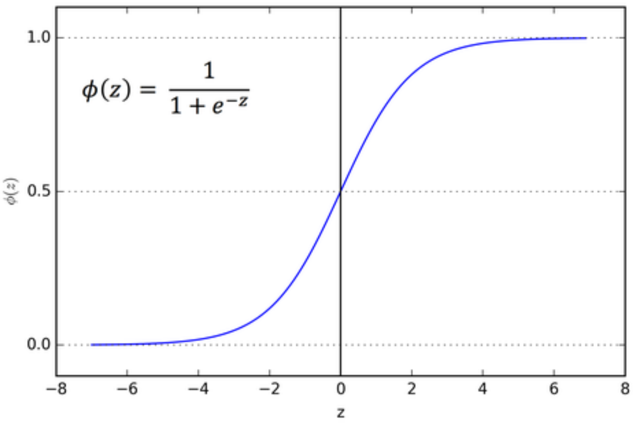


Fig. 6. Sigmoid function

2) *Hypotheses*

1) If the image are processed with Logistic Regression character classifier then the accuracy of the recognized characters will be better than the histogram-based approach.
2) If the image are processed with Logistic Regression character classifier then the time taken to detect and recognize the character is going to be significantly less than the time taken using the histogram approach.
3) If a ”noisy” image is input into Logistic Regression character classifier then this algorithm will have a higher success rate in the ”guessing” of characters than the histogram-based approach.

B. *Perceptron using Scikit-Learn*

Scikit-Learn is a really popular open-source software machine learning library and it is possible to build a multi-layer perceptron (feedforward artificial neural network) to recognize characters. [1]

Requirements

1) Import the image dataset and split into train and test set.
2) Flattening the images.
3) importing the required libraries (Numpy)
4) Initialize weights and bias.
5) Define an optimizer and learning rate for our network to optimize weights and biases on our given loss function.
6) Set a metric to evaluate model’s performance
7) Start training the network on train data and evaluate it on test dataset simultaneously.
8) Visualize test accuracy as a function of the number of epochs.

1) *Hypotheses*

1) Neural networks are more complex and complete systems and variating parameters in the network will result into more accurate results.
2) If you iterate multiple times through the training data and adjust the weights of the neural networks on each iteration, you get more accurate results just increasing the number of iterations.
3) An implementation of a characters recognizer using artificial neural networks is more efficient in terms of execution time and memory consumption.

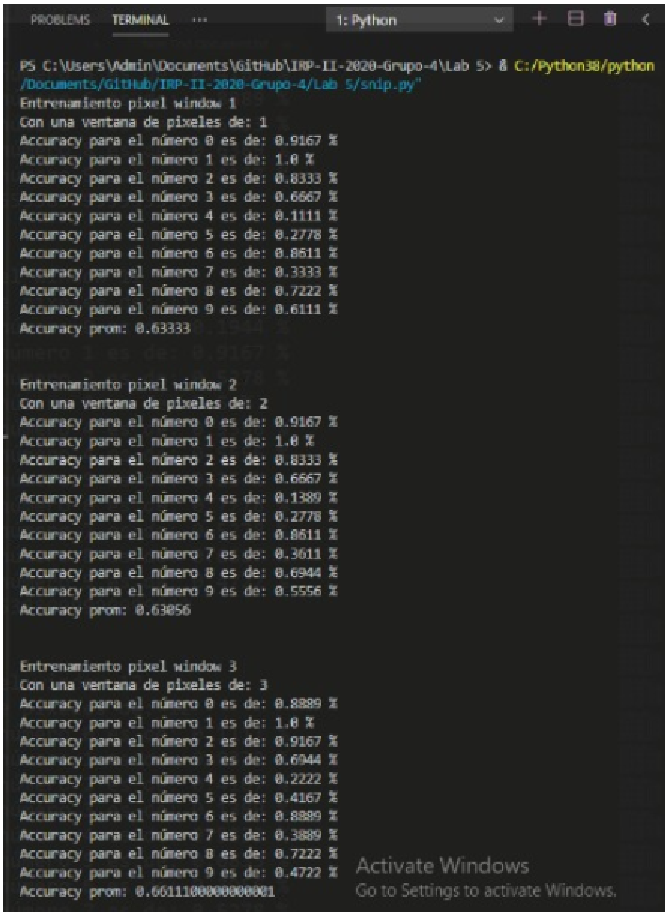
References

[1] Aayush Agrawal. *Multi-Layer perceptron using Tensorflow*. Sept. 2018. URL: https://towardsdatascience.com/multi-layer-perceptron-using-tensorflow-9f3e218a4809.

[2] Ligdi Gonzalez. *REGRESIÓN LOGÍSTICA - TEORÍA — 37 Curso Machine Learning con Python*. Youtube. 2019. URL: https://www.youtube.com/watch?v=KN167eUcvrs&feature=youtu.be&ab_channel=AprendeIAconLigdiGonzalez.

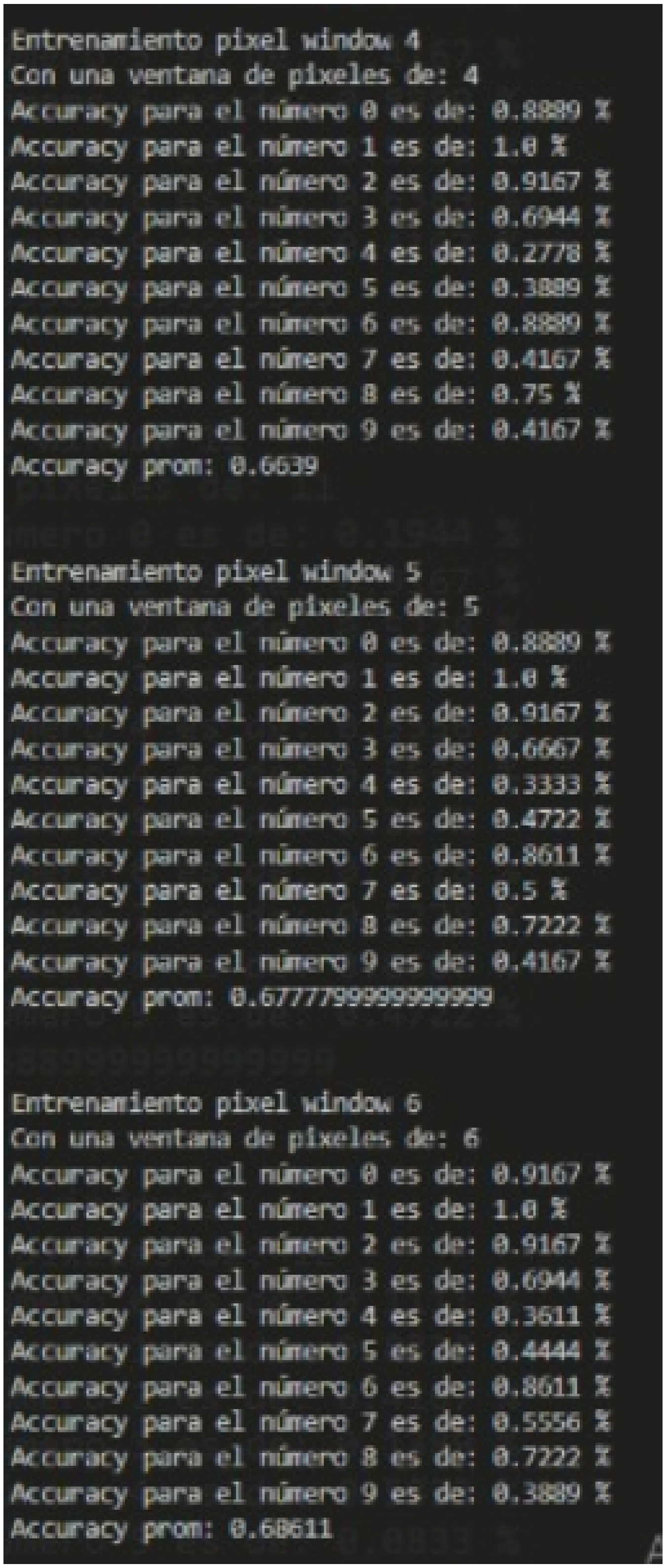
[3] $sklearn.linear_model.LogisticRegression$. scikit-learn. 2020. URL: https://scikit-learn.org/stable/modules/generated/sklearn.linear%5C_model.LogisticRegression.html.

Appendix A
Metrics

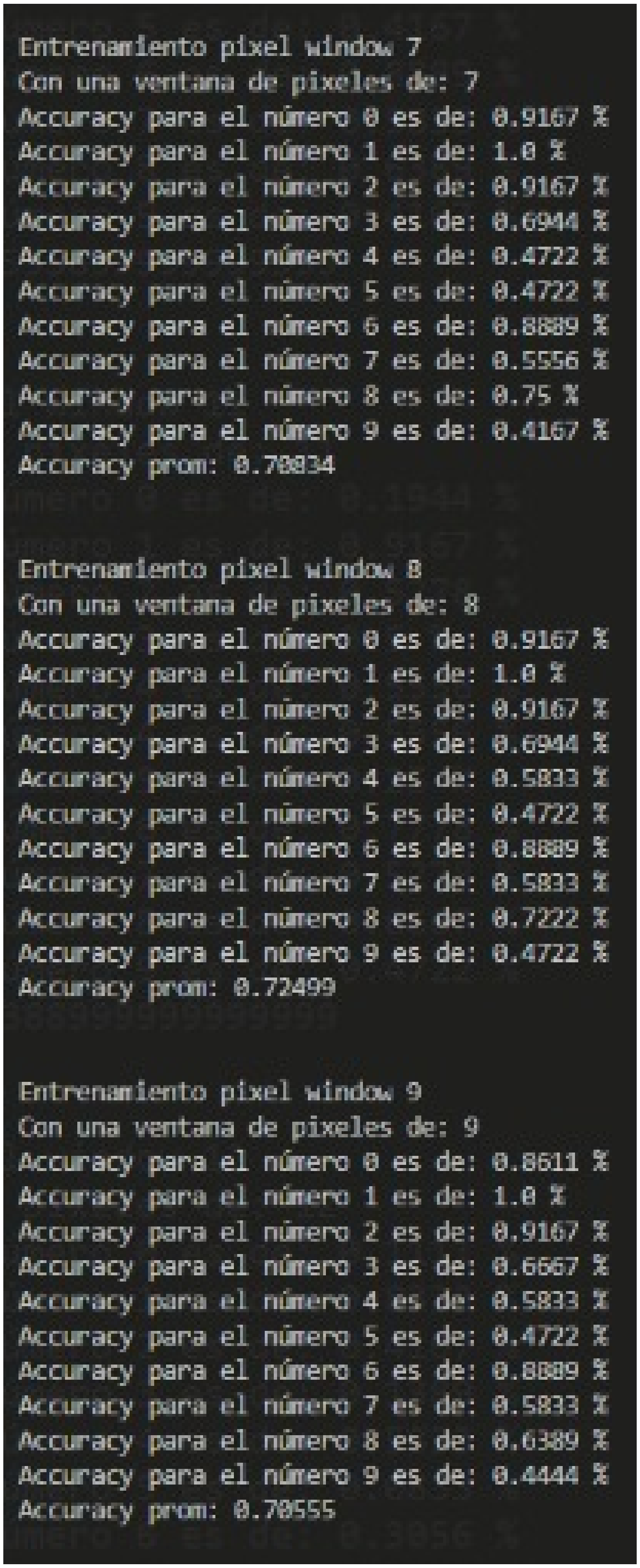


```
PS C:\Users\Admin\Documents\GitHub\IRP-II-2020-Grupo-4\Lab 5> & C:/Python38/python
/Documents/GitHub/IRP-II-2020-Grupo-4/Lab 5/snip.py"
Entrenamiento pixel window 1
Con una ventana de pixeles de: 1
Accuracy para el número 0 es de: 0.9167 %
Accuracy para el número 1 es de: 1.0 %
Accuracy para el número 2 es de: 0.8333 %
Accuracy para el número 3 es de: 0.6667 %
Accuracy para el número 4 es de: 0.1111 %
Accuracy para el número 5 es de: 0.2778 %
Accuracy para el número 6 es de: 0.8611 %
Accuracy para el número 7 es de: 0.3333 %
Accuracy para el número 8 es de: 0.7222 %
Accuracy para el número 9 es de: 0.6111 %
Accuracy prom: 0.63333

Entrenamiento pixel window 2
Con una ventana de pixeles de: 2
Accuracy para el número 0 es de: 0.9167 %
Accuracy para el número 1 es de: 1.0 %
Accuracy para el número 2 es de: 0.8333 %
Accuracy para el número 3 es de: 0.6667 %
Accuracy para el número 4 es de: 0.1389 %
Accuracy para el número 5 es de: 0.2778 %
Accuracy para el número 6 es de: 0.8611 %
Accuracy para el número 7 es de: 0.3611 %
Accuracy para el número 8 es de: 0.6944 %
Accuracy para el número 9 es de: 0.5556 %
Accuracy prom: 0.63056


Entrenamiento pixel window 3
Con una ventana de pixeles de: 3
Accuracy para el número 0 es de: 0.8889 %
Accuracy para el número 1 es de: 1.0 %
Accuracy para el número 2 es de: 0.9167 %
Accuracy para el número 3 es de: 0.6944 %
Accuracy para el número 4 es de: 0.2222 %
Accuracy para el número 5 es de: 0.4167 %
Accuracy para el número 6 es de: 0.8889 %
Accuracy para el número 7 es de: 0.3889 %
Accuracy para el número 8 es de: 0.7222 %
Accuracy para el número 9 es de: 0.4722 %
Accuracy prom: 0.6611100000000001
```

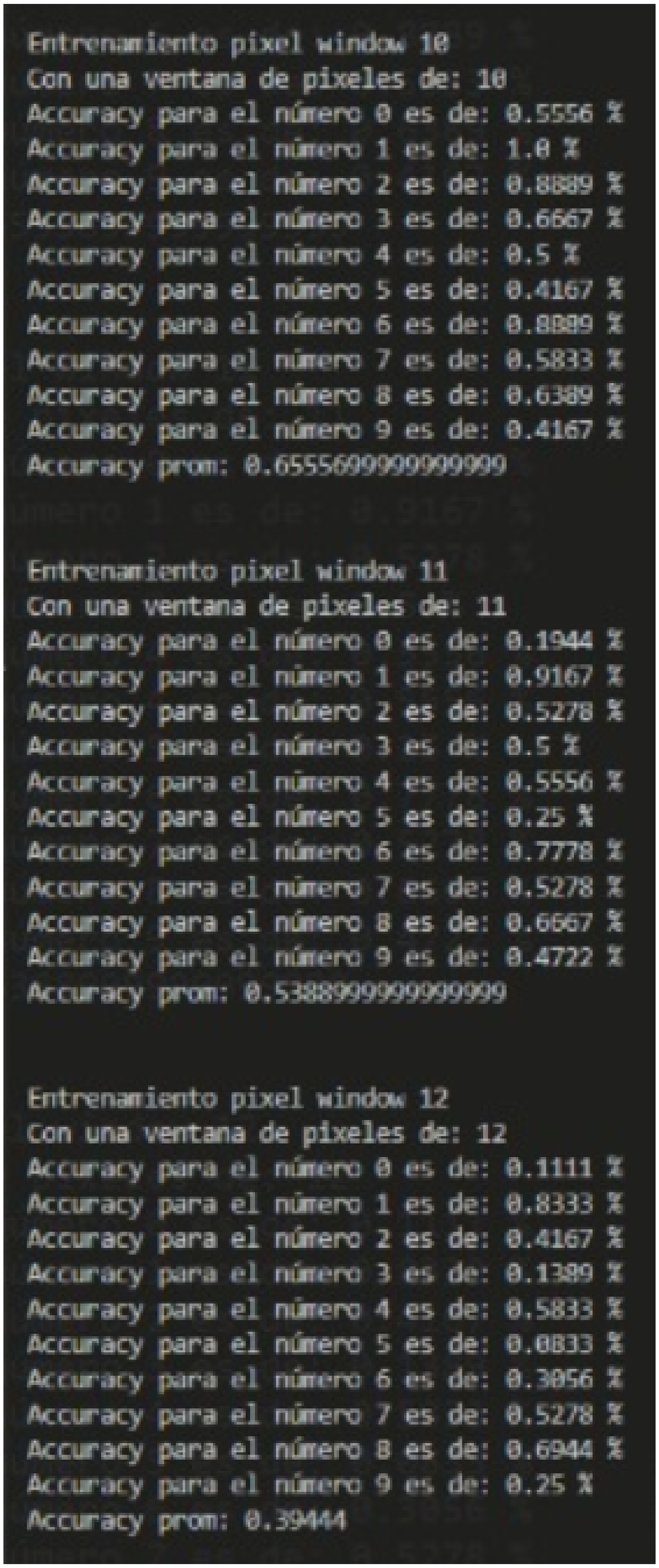
Activate Windows
Go to Settings to activate Windows.



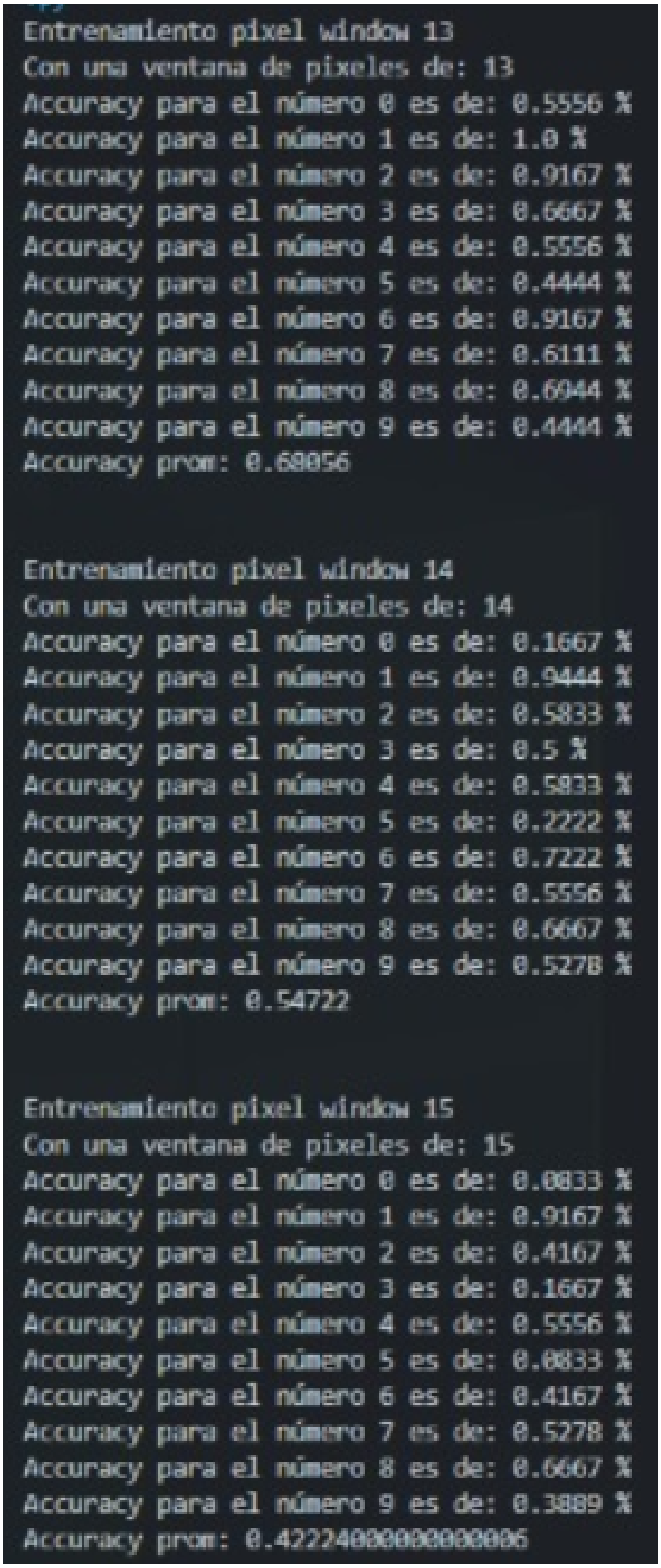
```
Entrenamiento pixel window 4
Con una ventana de pixeles de: 4
Accuracy para el número 0 es de: 0.8889 %
Accuracy para el número 1 es de: 1.0 %
Accuracy para el número 2 es de: 0.9167 %
Accuracy para el número 3 es de: 0.6944 %
Accuracy para el número 4 es de: 0.2778 %
Accuracy para el número 5 es de: 0.3889 %
Accuracy para el número 6 es de: 0.8889 %
Accuracy para el número 7 es de: 0.4167 %
Accuracy para el número 8 es de: 0.75 %
Accuracy para el número 9 es de: 0.4167 %
Accuracy prom: 0.6639


Entrenamiento pixel window 5
Con una ventana de pixeles de: 5
Accuracy para el número 0 es de: 0.8889 %
Accuracy para el número 1 es de: 1.0 %
Accuracy para el número 2 es de: 0.9167 %
Accuracy para el número 3 es de: 0.6667 %
Accuracy para el número 4 es de: 0.3333 %
Accuracy para el número 5 es de: 0.4722 %
Accuracy para el número 6 es de: 0.8611 %
Accuracy para el número 7 es de: 0.5 %
Accuracy para el número 8 es de: 0.7222 %
Accuracy para el número 9 es de: 0.4167 %
Accuracy prom: 0.6777799999999999


Entrenamiento pixel window 6
Con una ventana de pixeles de: 6
Accuracy para el número 0 es de: 0.9167 %
Accuracy para el número 1 es de: 1.0 %
Accuracy para el número 2 es de: 0.9167 %
Accuracy para el número 3 es de: 0.6944 %
Accuracy para el número 4 es de: 0.3611 %
Accuracy para el número 5 es de: 0.4444 %
Accuracy para el número 6 es de: 0.8611 %
Accuracy para el número 7 es de: 0.5556 %
Accuracy para el número 8 es de: 0.7222 %
Accuracy para el número 9 es de: 0.3889 %
Accuracy prom: 0.68611
```



```
Entrenamiento pixel window 7
Con una ventana de pixeles de: 7
Accuracy para el número 0 es de: 0.9167 %
Accuracy para el número 1 es de: 1.0 %
Accuracy para el número 2 es de: 0.9167 %
Accuracy para el número 3 es de: 0.6944 %
Accuracy para el número 4 es de: 0.4722 %
Accuracy para el número 5 es de: 0.4722 %
Accuracy para el número 6 es de: 0.8889 %
Accuracy para el número 7 es de: 0.5556 %
Accuracy para el número 8 es de: 0.75 %
Accuracy para el número 9 es de: 0.4167 %
Accuracy prom: 0.70834

Entrenamiento pixel window 8
Con una ventana de pixeles de: 8
Accuracy para el número 0 es de: 0.9167 %
Accuracy para el número 1 es de: 1.0 %
Accuracy para el número 2 es de: 0.9167 %
Accuracy para el número 3 es de: 0.6944 %
Accuracy para el número 4 es de: 0.5833 %
Accuracy para el número 5 es de: 0.4722 %
Accuracy para el número 6 es de: 0.8889 %
Accuracy para el número 7 es de: 0.5833 %
Accuracy para el número 8 es de: 0.7222 %
Accuracy para el número 9 es de: 0.4722 %
Accuracy prom: 0.72499

Entrenamiento pixel window 9
Con una ventana de pixeles de: 9
Accuracy para el número 0 es de: 0.8611 %
Accuracy para el número 1 es de: 1.0 %
Accuracy para el número 2 es de: 0.9167 %
Accuracy para el número 3 es de: 0.6667 %
Accuracy para el número 4 es de: 0.5833 %
Accuracy para el número 5 es de: 0.4722 %
Accuracy para el número 6 es de: 0.8889 %
Accuracy para el número 7 es de: 0.5833 %
Accuracy para el número 8 es de: 0.6389 %
Accuracy para el número 9 es de: 0.4444 %
Accuracy prom: 0.70555
```

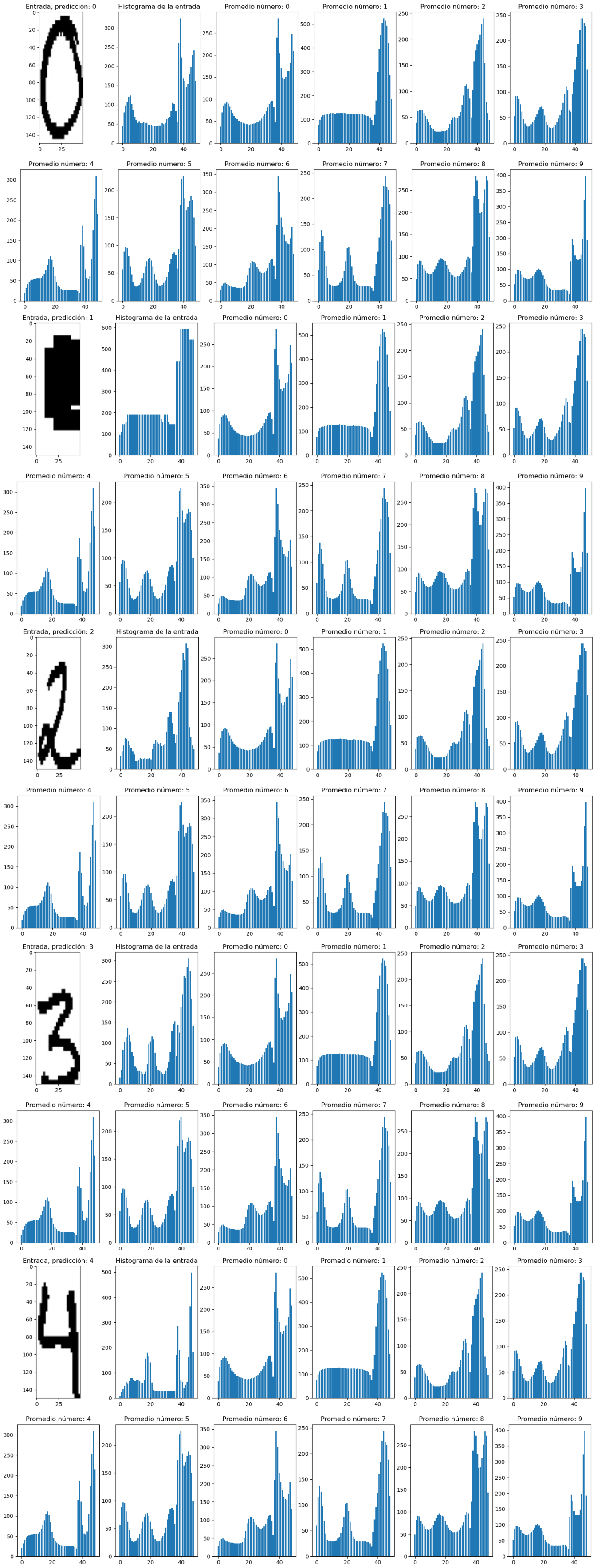


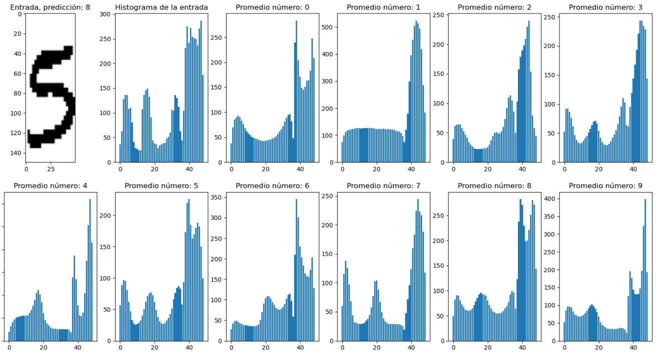
```
Entrenamiento pixel window 10
Con una ventana de pixeles de: 10
Accuracy para el número 0 es de: 0.5556 %
Accuracy para el número 1 es de: 1.0 %
Accuracy para el número 2 es de: 0.8889 %
Accuracy para el número 3 es de: 0.6667 %
Accuracy para el número 4 es de: 0.5 %
Accuracy para el número 5 es de: 0.4167 %
Accuracy para el número 6 es de: 0.8889 %
Accuracy para el número 7 es de: 0.5833 %
Accuracy para el número 8 es de: 0.6389 %
Accuracy para el número 9 es de: 0.4167 %
Accuracy prom: 0.6555699999999999

Entrenamiento pixel window 11
Con una ventana de pixeles de: 11
Accuracy para el número 0 es de: 0.1944 %
Accuracy para el número 1 es de: 0.9167 %
Accuracy para el número 2 es de: 0.5278 %
Accuracy para el número 3 es de: 0.5 %
Accuracy para el número 4 es de: 0.5556 %
Accuracy para el número 5 es de: 0.25 %
Accuracy para el número 6 es de: 0.7778 %
Accuracy para el número 7 es de: 0.5278 %
Accuracy para el número 8 es de: 0.6667 %
Accuracy para el número 9 es de: 0.4722 %
Accuracy prom: 0.5388999999999999

Entrenamiento pixel window 12
Con una ventana de pixeles de: 12
Accuracy para el número 0 es de: 0.1111 %
Accuracy para el número 1 es de: 0.8333 %
Accuracy para el número 2 es de: 0.4167 %
Accuracy para el número 3 es de: 0.1389 %
Accuracy para el número 4 es de: 0.5833 %
Accuracy para el número 5 es de: 0.0833 %
Accuracy para el número 6 es de: 0.3056 %
Accuracy para el número 7 es de: 0.5278 %
Accuracy para el número 8 es de: 0.6944 %
Accuracy para el número 9 es de: 0.25 %
Accuracy prom: 0.39444
```

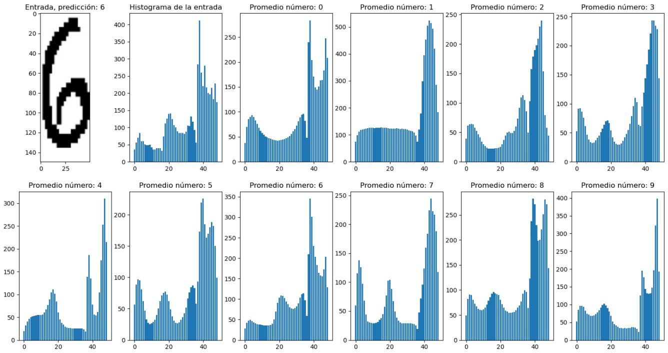


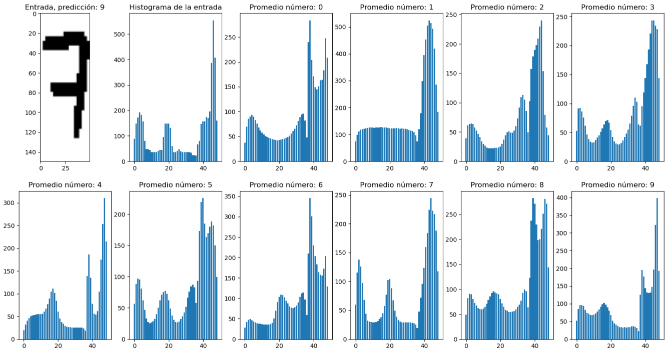
```
Entrenamiento pixel window 13
Con una ventana de pixeles de: 13
Accuracy para el número 0 es de: 0.5556 %
Accuracy para el número 1 es de: 1.0 %
Accuracy para el número 2 es de: 0.9167 %
Accuracy para el número 3 es de: 0.6667 %
Accuracy para el número 4 es de: 0.5556 %
Accuracy para el número 5 es de: 0.4444 %
Accuracy para el número 6 es de: 0.9167 %
Accuracy para el número 7 es de: 0.6111 %
Accuracy para el número 8 es de: 0.6944 %
Accuracy para el número 9 es de: 0.4444 %
Accuracy prom: 0.68056


Entrenamiento pixel window 14
Con una ventana de pixeles de: 14
Accuracy para el número 0 es de: 0.1667 %
Accuracy para el número 1 es de: 0.9444 %
Accuracy para el número 2 es de: 0.5833 %
Accuracy para el número 3 es de: 0.5 %
Accuracy para el número 4 es de: 0.5833 %
Accuracy para el número 5 es de: 0.2222 %
Accuracy para el número 6 es de: 0.7222 %
Accuracy para el número 7 es de: 0.5556 %
Accuracy para el número 8 es de: 0.6667 %
Accuracy para el número 9 es de: 0.5278 %
Accuracy prom: 0.54722


Entrenamiento pixel window 15
Con una ventana de pixeles de: 15
Accuracy para el número 0 es de: 0.0833 %
Accuracy para el número 1 es de: 0.9167 %
Accuracy para el número 2 es de: 0.4167 %
Accuracy para el número 3 es de: 0.1667 %
Accuracy para el número 4 es de: 0.5556 %
Accuracy para el número 5 es de: 0.0833 %
Accuracy para el número 6 es de: 0.4167 %
Accuracy para el número 7 es de: 0.5278 %
Accuracy para el número 8 es de: 0.6667 %
Accuracy para el número 9 es de: 0.3889 %
Accuracy prom: 0.42224000000000006
```

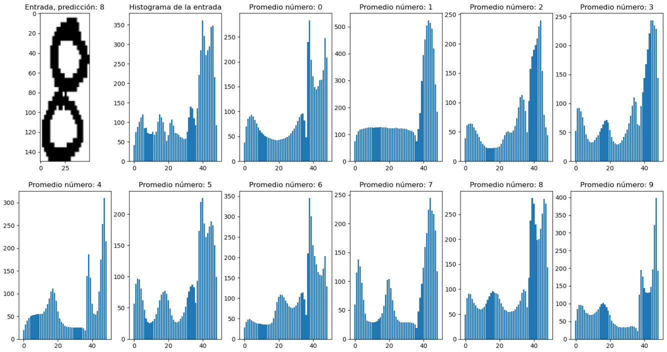
Appendix B
Number histogram comparison and prediction

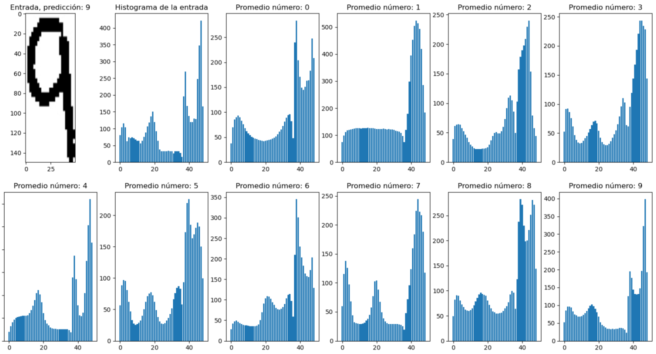



Entrada, predicción: 8
Histograma de la entrada
Promedio número: 0
Promedio número: 1
Promedio número: 2
Promedio número: 3
Promedio número: 4
Promedio número: 5
Promedio número: 6
Promedio número: 7
Promedio número: 8
Promedio número: 9


Entrada, predicción: 6
Histograma de la entrada
Promedio número: 0
Promedio número: 1
Promedio número: 2
Promedio número: 3
Promedio número: 4
Promedio número: 5
Promedio número: 6
Promedio número: 7
Promedio número: 8
Promedio número: 9


Entrada, predicción: 9
Histograma de la entrada
Promedio número: 0
Promedio número: 1
Promedio número: 2
Promedio número: 3
Promedio número: 4
Promedio número: 5
Promedio número: 6
Promedio número: 7
Promedio número: 8
Promedio número: 9


Entrada, predicción: 8
Histograma de la entrada
Promedio número: 0
Promedio número: 1
Promedio número: 2
Promedio número: 3
Promedio número: 4
Promedio número: 5
Promedio número: 6
Promedio número: 7
Promedio número: 8
Promedio número: 9


Entrada, predicción: 9
Histograma de la entrada
Promedio número: 0
Promedio número: 1
Promedio número: 2
Promedio número: 3
Promedio número: 4
Promedio número: 5
Promedio número: 6
Promedio número: 7
Promedio número: 8
Promedio número: 9