Final Project

Cristian Camilo Escobar Rojas - 1455778, Daniel Alejandro Rivera Zapata - 1455817,

Faculty of Engineering, Universidad del valle

Tuluá, Colombia

escobar.cristian@correounivalle.edu.co

rivera.daniel@correounivalle.edu.co

ABSTRACT

"Digital image processing, in general terms, involves the recognition of 2D images, 3D images and sequences of images, analysis, manipulation, transmission and other related areas. Parts covered by this area are: intensity transformations and spatial filters, processing in the frequency domain, image restoration, color processing, wavelets, digital image compression, morphological processing, segmentation, representation and description, shape recognition and objects, interpretation" [1]. This, opens doors to a new world of possibilities, on this occasion, recognizing the vehicle registration number allows us to take a step further, thus having important advances in fields such as security. These algorithms are usually used in surveillance cameras, be it in a shopping mall or the famous photo fines.The use of openCV, with the help of a support vector machine, allows the implementation of the algorithm proposed in this project.

I. INTRODUCTION

THE PROCESSING of images, aims to improve the appearance of images and make more evident in them certain details that you want to note [2]. On this occasion, identifies the characters that lie in the vehicle registration number is the task performed.

OpenCV (Open Source Computer Vision) is an open-source BSD-licensed library that includes hundreds of computer vision algorithms [3]. The latter has an important task in the development of the algorithm, since it is what allows to perform a preprocessing, so afterwards, with the help of a support vector machine, a solution is reached that allows the identification of characters in an enrollment vehicular, main objective of the present project.

The article is composed of the present introduction, PRIOR AND RELATED WORK where context of past works or related to what was taken into account for this project is given, DESCRIPTION OF THE ALGORITHM which explains how it was reached solution, algorithm results, project conclusions and finally the bibliography.

II. PRIOR AND RELATED WORK

Previously we worked with OpenCV, where algorithms of both filters and contours were performed. The latter manage to be of great help in the development process of the project, in addition, having witnessed the course of Artificial Intelligence that allowed to complement a good development, since, during the analysis of a possible solution, it was reached that this includes both the topics seen in the course, such as Artificial Intelligence topics. The latter is because, once the problem is analyzed it is understood that it is necessary to use a support vector machine, in order to give an optimal classification of the letters and numbers found in the vehicle registration number.

In conclusion, as previous work, we have the algorithms made in OpenCV throughout the course "Introduction to digital image processing", as the knowledge acquired in "Artificial Intelligence".

III. DESCRIPTION OF THE ALGORITHM

CLASSIFICATION AND DETECTION OF CHARACTER

The algorithm designed is based on the main examples seen in the course (image reading, color transformation, thresholding, morphological operations and edge detection).

Initially, a gray-scale color transformation and a Gaussian filter are used for the elimination of possible noise in the image. The image passes through a binary thresholding which allows to differentiate more easily the characters and the background of the image, then a closure is made applying two morphological transformations, two dilations and an erosion in the image, these transformations are made to separate the characters of other figures present in the image.

For the individual detection of each character, several alternatives were searched for an efficient detection, it was concluded that the area of the largest contours belong to the characters.

Once all the contours have been found in the imgen, an order is made and the 6 major contours are obtained (these correspond to the characters), each detected character is resized to a certain size and transformed into an array to be sent to previously trained Support Vector Machine (SVM)

TRAINING

In the training, different stages of complexity were performed using the Python sklearn library. Initially trained only one character of each class. By using so little data, accuracy is not enough to obtain reliable results. One way to get better results in this training was to decrease the size of the pixel array to only a few (from 6 to 10), giving very good results in the same training set and some successes with new images.

Subsequently, the size of the training set is increased to 1017 characters per class, that is, approximately 36000 training characters which could lead to a memory error which could not be identified.

IV. EXPERIMENTAL RESULTS

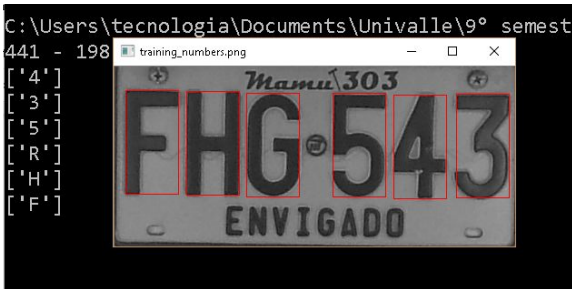

figure 1: result with good approximation

With an image size of 7px by 7px we obtained results very approximate to the desired, increasing the size of the imgagen increases the details of the image and decreases the approximation to the result given as a consequence that all characters are identified as the same class.

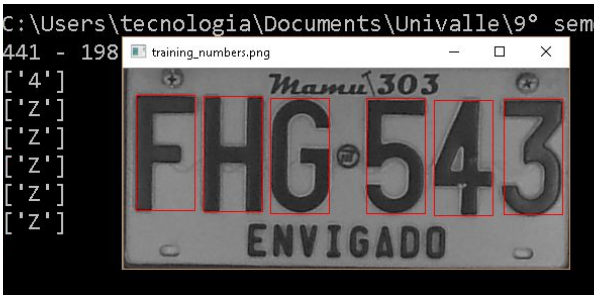

figure 2: result with bad approximation

OTHERWISE

In order to improve the results obtained in the implementation described above, it was decided to make use of a dataset called 74kcharts, which has a set of 66 folders, each containing 1016 images of the letters with a size of 128x128. Now, the training was divided into a separate file, called training.py, which is responsible for reading the dataset and saving a .pkl file, which contains all the training done, the latter was done with a view to not perform more once the training, but, it will be done once and saved for later tests.

```python
def read():
    imgs = np.zeros((36576, 16384), np.bool_) #BINARI IMAGE
    z = np.zeros((36576, 1), np.uint8) #CLASSES
    indice = 0
    for i in range(1, 1016): #1016 IS THE NUMBER OF IMAGES FOR FOLDER
        for x in range(1, 37):
            folder_name = "Sample{0:03d}".format(x)
            file_name_prefix = "img{0:03d}-".format(x)

            file_name = "{0}{1:05d}.png".format(file_name_prefix, i)
            img_path = "Fnt\\{}\\{}".format(folder_name, file_name)

            this_img = cv2.imread(img_path, 0)
            this_data_img = this_img
            this_target_label = x-1

            imgs[indice, :] = this_img.reshape(1, 16384)
            z[indice, 0] = this_target_label
            indice += 1
    print(indice)
    return imgs, z
```

figure 3: reading the dataset

As in the procedure described above, an SVM was used, with the difference that on this occasion there was a large dataset. However, it should be noted that every image was passed to binary, this is that although it was previously represented with 0-255, we opted for

a representation of 0-1, this in order to optimize memory.

```
images, classes = read() #RESULT OF READ FUNCTION
imagesX = images
mask = imagesX == 255
imagesX[mask] = 1 #BINARI MASK FOR IMAGES
clf = svm.SVC(gamma=0.001, C=100)

clf.fit(imagesX, classes)
joblib.dump(clf, 'filename.pkl') #GENERATING THE PKL FILE
```

figure 4:Generating the pkl file

Finally, once the .pkl file is generated, the same recognition of letters is implemented, the contours are generated, it is done to resize and, before passing to the prediction, it is passed to binary. The latter has been done to generate a bad prediction, as much as the training and the test data must be on the same scale. It is worth mentioning that they are tested with the same license plates used in the previous approach, in addition to the use of the Python sklearn library.

Once the results are obtained, it is worth mentioning that the prediction thrown is not the best, in fact, it classifies all the letters in the same class. Obviously something is wrong, and as future work it is proposed to correct the error to obtain a prediction above 70%.

V. CONCLUSIONS

- Because a VMS is based on supervised learning, having a large data set, this for the training phase, will yield a more favorable result.
- In small training sets, to achieve a closer approximation you can decrease the size of the image.
- In order to advance, only ingenuity is needed, since, as could be observed in this project, a basic function such as the contour, added with artificial intelligence, can generate a very useful application such as the recognition of a vehicular registration.

VI. BIBLIOGRAPHY

[2] Digital image processing, taken from: http://alojamientos.us.es/gtocoma/pid/introduccion.html

[2] The processing of images, taken from: http://bibliotecadigital.ilce.edu.mx/sites/ciencia/volumen2/ciencia3/084/htm/sec_9.htm

[3] OpenCV, taken from: https://docs.opencv.org/2.4/modules/core/doc/intro.html