License Plate Recognition: Localization
Final Project Report

Arthur CLAUDE
arthur.claude@supelec.fr

Mathieu BRUS
mathieu.brus@supelec.fr

Hugo GRANVILLE
hugo.granville@supelec.fr

Abstract

License plate recognition has an important role in traffic control, security systems and law enforcement. The first important part of this recognition is the localization of the plate on the image in order to extract it. In this report, we first present three algorithms made to realize this task that we implemented following their descriptions in articles. The algorithms are based on different approaches. The first one relies on morphological operations to extract important contrast features to identify plates. The second one is based on edge detection that is supposed to capture the rich edges and texture information that license plates can offer in their surroundings. The final one relies on blob detection in images and filtering using geometrical properties to directly detect the letters in the license plate. This simplifies the next phase of the recognition, which is identifying the text of the plate. Finally, we expose the evaluations we made to compare the performances of the algorithms. We compute the Intersection over Union score for each algorithm for 150 images of US plates and 20 images from Brazil. We also calculate the computation time by image. Thanks to this evaluations, we compare the algorithms and draw some conclusions on each one of them.

1. Introduction/Motivation

License plate recognition is an image-processing technique used to identify vehicles by their license plates. It is an interesting problem to be addressed in a technological point of view but it also has very useful applications in our every day lives. Indeed, it can play an important role in various fields such as traffic control and measurement, border control, electronic toll collection, parking automation and security, and law enforcement (stolen car detection, red-light enforcement, over-speed charging...).

For every vehicle plate number recognition model, the two main steps are locating the license plate in the image and identifying the text of the license plate. The first task, license plate localization, has been an active area of research lately. In fact, it is a challenging task due to the wide variation in license plates standards across countries (plate size, shape, color, texture, spatial orientation and position). The second task, text extraction, is related to optical character recognition (OCR), one of the earliest addressed computer vision tasks. It is very challenging as well because of the variations in text size, font, style, orientation and alignment. The complexity of the background of the images can also complicate the task in some cases.

In this project, we will only focus on the license plate localization task. Our goal is to provide an experimental evaluation of three different algorithms selected from the state of the art with the utmost care.

2. Problem Definition

In order to evaluate the algorithms, we will use a dataset [1] composed of 445 car images each of which is associated to a text file giving the ground truth, i.e. the position of the corresponding license plate. It also offers 22 additional car images without any text file. These pictures are separated into 3 folders, the cars registered in the United States, those in Brazil and those in Europe. Nevertheless, since two of our algorithms need the images to have the same sizes, we decided to select only the images of the dataset with the dimensions 720*1280. Thus, we only compared our algorithms on 150 images from the United States and 20 images from Brazil.

In order to carry out this project, we took two main steps. First, we implemented the three algorithms based on our understanding of their principles. We tried to be the most faithful possible and made sure the algorithms were compatible with our test data set. Second, we evaluated them with some specific metrics and compared them between each other to take some hindsight on their strengths and weaknesses .

We also implemented two simple functions that are quite similar and that allow to visualize the bounding box corresponding to the ground truth of an image and the ones representing the results of our algorithms.

We performed the evaluations of the three models on the two geographic areas separately (United States and Brazil) in order to highlight if this parameter had an impact on their performances.

At first, we wanted to compare the algorithms using the mean average precision score (mAP), which is widely used for object detection models as it considers both the classification and the localization aspects of the problem. We wanted to compute the mAP score using the Intersection over Union (IoU) to determine whether a bounding box is a True Positive or a False Positive. Unfortunately, since most of our IoU scores were too low, with a reasonable threshold some bounding boxes regions were considered false even though they were true. We therefore got terrible results using this metric, so we decided to simplify it. Instead, we chose to compute for each image the global IoU on the different geographic areas. Then we took the average score over all images.

As another evaluation technique, we also compared the computation time by image to see if those algorithms ran in a reasonable time and if they could effectively be used in the real-world applications we spoke about earlier.

3. Related Work

As we said earlier, vehicle plate recognition can find applications in a number of traffic control and surveillance systems. For this reason, this task has been an active area of research in the field of computer vision, and a large number of methods have been proposed.

Existing techniques use information from the car image to aid the identification of the potential license plate areas. This information can be categorized as either boundary or region based information. In fact, boundary information can be used to detect the various edges which are present in the license plate area [6] [9], while region based approaches are often based on the study of colors [7] [4] [8]. In addition, sometimes, a combination of two or more techniques which use boundary or region information have been proposed [5].

In our project, we decided to focus on three methods. These methods are presented in the following section. We will not look into the other methods in further details.

4. Methodology

4.1. Morphology-based method

The first method is a morphology-based technique. It consists in extracting important contrast features as guides in order to identify the license plates and get some potential candidates. Then, we choose the best ones knowing some characteristics of the license plates we need to detect. Finally, a recovery algorithm is applied to reconstruct a license plate in the case it is fragmented into several parts, but we did not implement it as it was hardly understandable and, in the tests we ran we never got several parts of the plate separated.[3].
In the following paragraphs, we will explain this algorithm and illustrate the steps thanks to the following car image:

Figure 1. Image of the car

4.1.1 Feature Extraction Using Morphological Operations

As the goal of this method is to extract important contrast feature, the first operation to do is to preprocess the image using an equalization of histogram in order to reduce the lighting variations. Then a series of morphological operations are applied to extract potential candidates. Before showing the pipeline of this series, we will first define the equations used to apply all this morphological operations. Here, $S_{m,n}$ represents a structure elements which is a matrix containing only ones and $I(x, y)$ describe the pixel in position (x, y) of the image:

Smoothing operation:

$$E_{S_{m,n}}(I(x, y)) = \frac{1}{mn} \sum_{i=-n/2}^{n/2} \sum_{j=-m/2}^{m/2} I(x+i, y+j) S_{m,n}(i, j)$$

Dilation operation;

$$I(x, y) \oplus S_{m,n} = \max_{i<=n/2, j<=n/2} I(x-i, y-j) S_{m,n}(i, j)$$

Erosion operation;

$$I(x,y)\square S_{m,n} = \min_{i<=n/2, j<=n/2} I(x-i, y-j)S_{m,n}(i,j)$$

Closing operation:

$$I(x,y) \bullet S_{m,n} = (I(x,y) \oplus S_{m,n})\square S_{m,n}$$

Opening operation:

$$I(x,y) \circ S_{m,n} = (I(x,y)\square S_{m,n}) \oplus S_{m,n}$$

Differencing operation:

$$D(I_1, I_2) = |I_1(x,y) - I_2(x,y)|$$

Thresholding operation:

$$T(I(x,y)) = \begin{cases} 255 & \text{if } I(x,y) > T \\ 0 & otherwise \end{cases}$$

The series of operation is illustrated by the following pipeline:

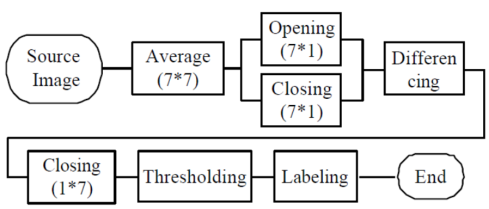

Figure 2. Pipeline of the morphological-based algorithm

As our images were bigger than the ones used in the article [3], we had to adapt the sizes of the structure elements shown in the pipeline.
First, in order to eliminate noises, a smoothing operation is applied to the pre-processed image. We did not change the size of the structure elements and let it be 7 by 7. The result of the preprocessing and the smoothing operation is shown on the following figure.

Figure 3. Image after preprocessing and smoothing

Then, closing and opening operations are applied to the smoothed image. But for these operations, we adapted the sizes of the structure elements used, we chose to use a structure element of size 18*1 as it allowed us to get the better results. Then, the two images obtained are differenced using the differencing operation which allow to detect vertical edges. The result of these operations is shown on the following picture.

Figure 4. Image after the differencing operation

As it is known that the vertical edges in a license plate are close and adjacent together, these adjacent edges are connected together through a closing operation. For this operation, we used a structure element of size 1*18. Finally, all possible vertical edges are extracted through a thresholding operation and all the areas are labelled thanks to connected components. The result of these operation is shown below.

Figure 5. Image containing only possible candidates

Thus, we obtained all possible candidates and we now need to select the best ones.

4.1.2 Candidate Selection

In order to select the best candidates, we will use some characteristics of the plates.

Thus, for each labelled area obtained, we first compute his width and height. We keep the areas that have a width greater than 60 and a height greater than 25.

Then, we compute the density of the areas remaining. The density is the ratio between the number of points in the labelled zone and the area of the rectangle surrounding it (the product between the width and the height). We only keep the zones that have a density greater than 0.25 and lower than 0.85.

Finally, we compute the ratio between the width and the height. As this ratio must be close to 2, we only keep the areas with a ratio greater than 1 and lower than 3.5.

Thus, we obtain the best candidates and we can draw them on the image to compare with the ground truth. These two images are shown below.

Figure 6. Ground truth

Figure 7. Result of the algorithm

As we can see, the algorithm worked quite well on this specific car as it detected a slightly smaller zone but which contains all the characters of the plate and this result led to an IoU score of 0.78.

4.2. Edges-based method

The second method is based on contour detection methods. This type of methods is effective thanks to the rich edge and texture information license plates can offer in their surrounding. We first enhance the original car image to boost up the plate area. Secondly, the vertical edges of the image are extracted, then, the background and noise edges are removed, and finally, the plate region is searched by in the residual edge image [11].

4.2.1 Image enhancement

On many car images, the gradients in the license plate area can be much lower than those in the other parts of the images. This can be caused by shadows, or by taking the photos during night or gloomy days. If we extract edges directly from these images, a few vertical edges will appear in the plate areas.

The solution to this issue is image enhancement. Our implementation is based on a method based on local standard deviation [10]. Here, we will use $I_{i,j}$ to denote the luminance of the pixel $P_{i,j}$ in the original image, and $I'_{i,j}$ to denote the luminance in the new image. $W_{i,j}$ is a window of size 40x80 (in order to represent the local information better, the size of this window should be smaller than the estimated size of the plate) centered on pixel $P_{i,j}$, $\overline{I_{W_{i,j}}}$ is the mean luminance of the pixels in this window, and $\sigma_{W_{i,j}}$ is the standard deviation of these pixels.

The enhancement equation is:

$$I'_{i,j} = f(\sigma_{W_{i,j}})(I_{i,j} - \overline{I}_{W_{i,j}}) + \overline{I}_{W_{i,j}}$$

where $f(\sigma_{W_{i,j}})$ is an enhancement coefficient defined by:

$$f(\sigma_{W_{i,j}}) = \begin{cases} \frac{3}{\frac{2}{400}(\sigma_{W_{i,j}} - 20)^2 + 1} & \text{if } 0 \leq \sigma_{W_{i,j}} < 20 \\ \frac{3}{\frac{2}{1600}(\sigma_{W_{i,j}} - 20)^2 + 1} & \text{if } 20 \leq \sigma_{W_{i,j}} < 60 \\ 1 & \text{if } \sigma_{W_{i,j}} 60 \end{cases}$$

Computing the values of $\overline{I}_{W_{i,j}}$ and $\sigma_{W_{i,j}}$ for each pixel is computationally expensive, and thus not advisable. We can use the bilinear interpolation algorithm to get them. The image is cut into 16x18 blocks equably, the values of $\overline{I}_{W_{i,j}}$ and $\sigma_{W_{i,j}}$ are computed at the vertexes of blocks, and we finally find every value thanks to bilinear interpolation:

$$\overline{I}_{W_{i,j}} = (1 - c_y)[(1 - c_x)\overline{I}_{W_A} + c_x\overline{I}_{W_B}] + c_y[(1 - c_x)\overline{I}_{W_C} + c_x\overline{I}_{W_D}] \quad (1)$$

$$\sigma_{W_{i,j}} = (1 - c_y)[(1 - c_x)\sigma_{W_A} + c_x\sigma_{W_B}] + c_y[(1 - c_x)\sigma_{W_C} + c_x\sigma_{W_D}] \quad (2)$$

Figure 8. Rectangle for bilinear interpolation

However, if the image is well illuminated and the image is in balance , the process will not change the contrast of the plate. The two images below show the result obtained for one of the photos of our dataset. Very few changes are made.

Figure 9. Original image (turned gray)

Figure 10. Enhanced image

As this enhancement operation is really computationally expensive (approximately 10 minutes for each image), and does not make any real improvements to our results, we have decided to leave it out.

4.2.2 Vertical edges extraction

The next step is the extraction of the vertical edges of the enhanced image. In order to accomplish this task, we use the vertical Sobel operator.

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

We convolve this operator with the car image to get the vertical gradient image.

Then, we compute the mean of the absolute gradient values in the image and multiply it by a coefficient in order to get a threshold. In our project, we used 4 as coefficient. We finally use this threshold to apply a non-maximum suppression in horizontal direction in the gradient image to obtain the vertical Sobel edge image.
The result obtained on our image is presented below.

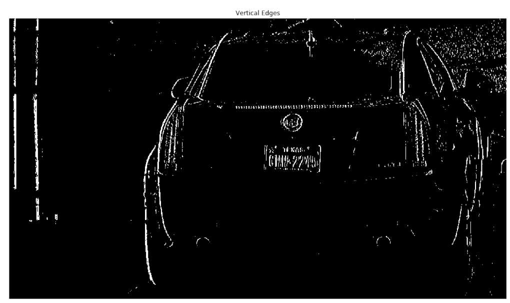

Figure 11. Vertical edges image

4.2.3 Background curve and noise removing

In the last image, we can notice that there are many long background curves and short random noise edges. These edges may interfere in the license plate localization.

In order to remove these edges, we scan the image three times. The first scan will record the edge lengths away from left to right and top to bottom, the second scan will record the edge lengths away from right to left and bottom to top, and the last scan will add up the two lengths to denote the actual lengths. Finally, if one edge has very long or very short length, it will be removed from the edge image. For our project, we used 20 and 150 as thresholds in order to remove these edges.
The result obtained on our image is presented below.

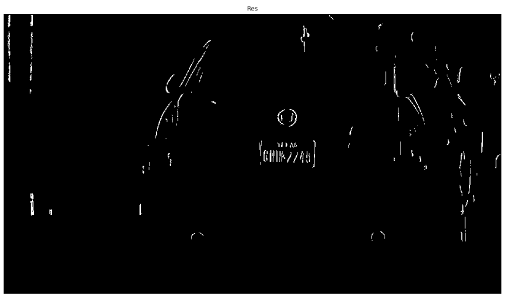

Figure 12. Vertical edges image without background noise and curve

4.2.4 License plate search and segmentation

The last step of our algorithm is the localization of the license plate area. To perform this task, we will shift a given-size rectangle window from left to right and top to bottom in the edge image. We will count the total number of the edges points in the window and if the number is above a certain value, there may be a license plate in the window.

As well, the size of the window has to be set at the beginning. This is one of the limitation of the algorithm. In our project we set a size of 80x150 for this window.

In order to make the search faster, the shift of the window is not performed pixel by pixel. As a first step, we cut our image into 16x18 blocks equably and count the number of edges points in each block. With these counts, we form a block image B of size 40x80. Then, we convolve the image B with a widow W of size $(150/16) * (80/18) = 9$x4 where each element is equal to 1. We obtain a new image B'.

The result obtained for B' on our image is presented below.

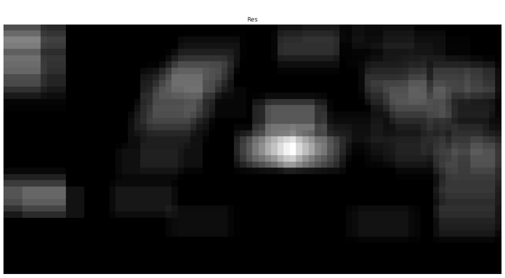

Figure 13. B' block image

Finally, in the B' image, we are looking for the points with a value higher than a threshold calculated as 80% of the maximum of the values of B' and which are a local maximum. These points are associated with a point in the original image, which is added to the candidates for the position of the center of the plate. In our algorithm, we only keep the one with the highest B' value.

So we have determined the central point of the plate, and the size of the window surrounding the plate has been predefined. We therefore have the result.

The result obtained on our image, and the ground truth are presented below.

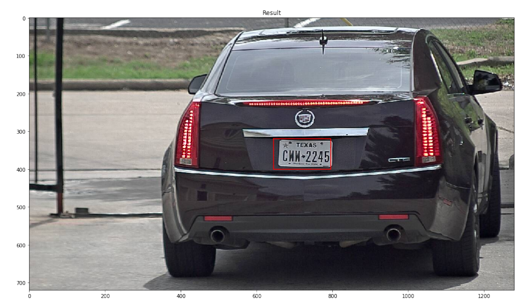

Figure 14. Result of the algorithm

Figure 15. Ground truth

In this example the IoU score obtained is 0.78.

4.3. Blob-based method

The last method we studied, described in paper [2], relies on blob detection in images. It was originally meant to directly detect the letters in the licence plate in order to run the text recognition phase. Since we are only focusing on the localisation of the licence plate here, we adapted the algorithm to get a unique box for the entire plate.

The general principle of this method is to first detect the text in the plates thanks to a maximally stable extremal regions algorithm (MSER) for blob detection. Second, the

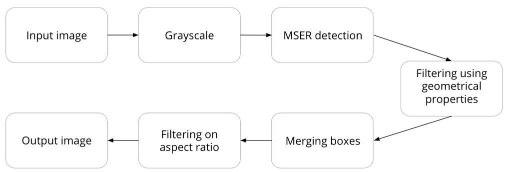

Figure 16. Pipeline of the method

blobs are filtered using geometric properties. Third, the overlapping boxes are merged into a bigger cluster. Finally, a last geometric filter is applied to select the best candidates.

4.3.1 MSER algorithm

The maximally stable extremal regions algorithm connects regions that have intensities of pixel values within a certain threshold. To be selected, a region must be both stable and extremal:

- a region is considered extremal when there is a drastic change in pixel intensities on the boarders.
- a region is considered stable when the area changes very slightly when the threshold varies.

To implement this first step we decided to use the black and white version of this algorithm, after converting the images to gray-scale. For the blob detection, we relied on the MSER library of Open CV.

Figure 17. Detection of blobs

4.3.2 Filtering using geometrical properties

The next step was to use region properties in order to filter all the selected areas. Unfortunately, it appeared very delicate to manipulate ellipses objects so we made the choice to convert the blobs into bounding boxes first, then filter on the aspect ratio of the rectangles, i.e. the height to width ratio. We decided to keep a quite wide range of boxes, the ones whose aspect ratio were between 1.3 and 4.5. Indeed, the rectangles we wanted to select could go from almost squares to very slim boxes for some characters such as ”1” or ”I”.

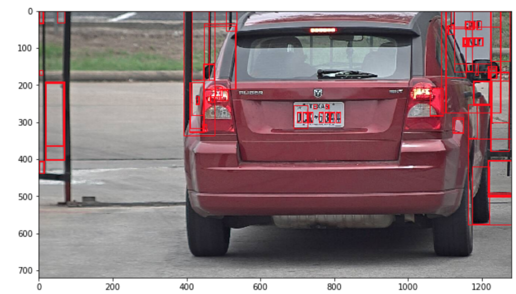

Figure 18. Bounding boxes after filtering on geometrical properties

4.3.3 Merging boxes and selecting candidates

As we can see, we managed to get rid of the regions that could not reasonably represent characters. The next step was to merge the overlapping boxes and group the closest ones to each other, in order to get the whole licence plate from the sole characters. This part was improvised as it was not a proper part of the paper. Thus, we implemented from scratch an algorithm that would group boxes close enough, with a slight preference on the abscissa axis. The general idea of the algorithm is to solve this with an iterative approach. At each iteration, we go through every bounding box and add it to a close cluster of boxes. At the end of an iteration, clusters are grouped into a new rectangle and become the new bounding boxes for next iteration. A relevant choice of our implementation was to try first the closeness to the latest created cluster of boxes, as boxes are arranged in such way that adjacent boxes in the list are likely to be close to each other in the image.

Figure 19. Merged bounding boxes

Once the merge was done, we filtered the remaining candidate bounding boxes on the aspect ratio. We empirically decided to keep the boxes whose aspect ratio were in the range $[0.15, 0.5]$, as licence plate shapes could change depending on the country and we wanted to keep a reasonable margin.

Figure 20. Result of the algorithm

In this example the IoU score obtained is 0.84.

5. Evaluation

5.1. Feedback on Morphology-based detection

This algorithm caused us a lot of trouble as it was very hard to adapt to our dataset. Indeed, by doing some morphological operations, such as opening or closing for example, we need the size of the structure element to be fixed. So, if we consider images of different sizes the algorithm will face difficulties to adapt. The size of the real license plate also needs to be fixed because of these operations. Indeed, while performing the closing operation to connect the adjacent vertical edges (i.e. to connect the letters of the plate) if the plate is big those edges will be far away and thus will not be connected by this operation. Furthermore, it is highly sensitive to the background of the image. For instance, if there are traffic or shop signs, it is likely that there will be detected by our algorithm and they can also influence the thresholding operation which will eventually erase the license plate if the signs have higher intensities. We have the same problem with trucks or commercial vehicles which can have text on them that will influence the detection. The next figure shows an example of one vehicle of the US dataset that has this problem and the license plate was never detected even when we changed the parameters of the algorithm.

However, this algorithm also has advantages. If the images are of high quality (which means a car centred on the image and the plate of a fixed size for all the images) the algorithm work quite well. It is the case for the Brazilian dataset where most images respect this and the average IoU score obtained is a lot better. Indeed, a score of 0.4 is

Figure 21. Problematic image for the morphology-based detection

quite good as many bounding boxes found by the algorithm are smaller than the ground truths but still contain all the characters of the plate which is enough for many real-world applications. Finally, the computation time is lower than 1 second which is very good for real-word applications which need a quick computation.

5.2. Feedback on edges-based detection

The method based on contour detection presents several limits.

Firstly, the fist step of this algorithm, the enhancement of the original image, is really computationally expensive. In fact, applying this step to an image of our dataset takes approximatively 10 minutes. Moreover, as we said earlier in this report, if the image is well illuminated and balanced, the process will not change the contrast of the plate. This operation is therefore not always required. We decided to leave it out for our tests. However, in some case, the results are strongly negatively impacted when this step is forgotten. This can be seen from the results obtained on the photos of Brazil from our dataset. In fact in many of these photos, there are reflections or shadows. The presence of this enhancement step can thus be seen as an advantage, even with its computational cost, in the case of images with contrast defects.

The computational cost is however always a shortcoming of this localization method. Indeed, this method is, even without the enhancement part, very long compared to the other methods studied. It takes about 3 seconds to process an image.

One other important limit of this algorithm is the fact that the size of the license plate on the image needs to be approximatively pre-determined. This can be a problem when the photos studied contain plates of different sizes (more or less far for example). Moreover, the shape of the plates may vary from one country to another, so the algorithm may be difficult to use in another country if it has been tuned for a particular shape. But if the images to be processed have plates of the same size and from the same country, the results will be good.

Moreover, our algorithm has been developed for a particular image size. If this size is changed, or if we want to make it work on multiple image sizes, adjustments will be necessary. Indeed, for example, the size of the cutting windows at the different stages will be modified.

We can also notice the lack of precision of this algorithm. This algorithm finds the center pixel of the detected license plate. Then, the window of given size is placed from this point. However, the detected point is often not exactly in the middle of the plate because of the approximations made when searching for it (block cutting). Thus it is very probable that there is a mismatch between the plate and the returned window.

Finally, our algorithm return only one solution for the license plate area. If it is wrong, it is actually possible that the right solution was found among the candidates in the last step, but not exploited. An adjustment could be considered in order to return all candidates.

5.3. Feedback on blob-based detection

As for this last algorithm, a lot of advantages can be noted. First, it requires no particular knowledge on the dataset and takes no particular input other than the images themselves (no estimation of the plate size for instance). Furthermore, contrarily to the two other algorithms, it does not require the images to have the same dimensions to be efficient. Moreover, regarding the structure of the algorithm, this one is the simplest to implement since it does not need the image to be prepossessed and does not rely on complex mathematical transformations.

A relevant downside is that it requires a very high quality of image to work properly, since small characters cannot be detected if the resolution is too poor. Another disadvantage is that any text on the car or on the background could fool the algorithm and would result in false positive detections.

Regarding our evaluations and in comparison to the others, the blob detection method performs well. It offers the best computation time (less than 0.2 second per frame) and gives overall quite satisfying results with a very good precision on detection. For instance, on the brasilian dataset, all the licence plates are detected and there is no false positives.

However, its scores are not incredible because this algorithm suffers a bias. As mentioned before, it was originally meant to directly detect the plate number, not the whole plate. As a consequence, we get very tight bounding boxes around the characters, which often result in IoU scores under 0.4.

Figure 22. Highlighting the bias

In this image the IoU score obtained is only 0.39 even though the interesting part of the plate was correctly detected. With a better image quality, we noticed that this algorithm could fit better the whole plate since small characters above the plate number could be detected as well (See the letters ”TEXAS” on the licence plate).

6. Conclusion

We managed to implement quite faithfully the algorithms described in the three papers. These do work well with very satisfying computation times. Time is a very important thing since in its real application, a licence plate detection algorithm must work within a second. We can easily imagine that drivers would be angry to wait an our at an electronic toll collection simply because the licence plate detection algorithm takes more than a minute to work for each car.

However, the mean scores of our algorithms on the test dataset are rather disappointing. This is mainly due to the fact that these algorithms are really efficient only in perfect conditions, on very well adapted pictures. In the real world, these could have a hard time to work properly since a lot of parameters depend on the context in which the picture is taken: homogeneity of the background, orientation of the car, distance of the car, quality of the image, shape and color of the plate, amongst other.

To go further and aim for better results a great idea would be to try to combine them into a single, more robust and precise one. We could even imagine applying adapting weights on each algorithm predictions depending on each algorithm average performance in the given context. Another solution to try to reach better results could have been to train a classifier on our dataset since we have access to the ground truth. A machine learning algorithm could indeed support the computer vision algorithm by classifying candidates bounding boxes in each image. Nevertheless, these suggested improvements fall outside of the framework of our project as our purpose was only to implement and compare the algorithms described in the three papers.

References

[1] Benchmark tests supporting the openalpr library. *Github [online], Available at* `https://github.com/openalpr/benchmarks`, 2020.

[2] B. A. J. R. B. Jin, R. Raj-Prasad. Vehicle license plate and color recognition using computer vision. *Department of Mechanical Engineering - Columbia University*, 2016.

[3] J.-W. Hsieh, S.-H. Yu, and Y.-S. Chen. Morphology-based license plate detection in images of differently illuminated and oriented cars. *Journal of Electronic Imaging*, 11(4):507–516, 2002.

[4] W. Jia, H. Zhang, and X. He. Region-based license plate detection. *J. Network Comput. Appl.*, 30:1324–1333, 2007.

[5] D. M. M.A. Lalimi, S. Ghofrani. A vehicle license plate detection method using region and edge based methods. *Comput. Electr. Eng.*, 39:834–845, 2013.

[6] S. Saha, S. Basu, M. Nasipuri, and D. Basu. A novel scheme for binarization of vehicle images using hierarchical histogram equalization technique. *International Conference on Computer, Communication, Control and Information Technology*, pages 270–274, 2009.

[7] W. Z. X. Shi and Y. Shen. Automatic license plate recognition system based on color image processing. *ICCSA 2005 LNCS*, 3483:1159–1168, 2005.

[8] X. Yan, X. Hao, and G. Zhao. License plate location based on trichromatic imaging and color-discrete characteristic. *Int. J. Light Electron Opt.*, 123:1486–1491, 2012.

[9] Y. Yanamura, D. Nishiyama, M. Soga, H. Nakatani, and H. Saji. Extraction and tracking of the license plate using hough transform and voted block matching. *Proceedings of IEEE intelligent vehicles symposium*, pages 243–246, 2003.

[10] D. Zheng, J. Wang, and Z. Xiao. Image enhancement based on local standard deviation. *Journal of Information Computational Science 2*, 2:429–437, 2005.

[11] D. Zheng, Y. Zhao, and J. Wang. An efficient method of license plate location. *Pattern Recognition Letters*, 26:2431–2438, 2005.