CONVOLUTIONAL NEURAL NETWORKS FOR LICENSE PLATE DETECTION IN IMAGES

Francisco Delmar Kurpiel, Rodrigo Minetto, Bogdan Tomoyuki Nassu

Federal University of Technology — Parana, Brazil
francisco@kurpiel.eng.br, rminetto@utfpr.edu.br, btnassu@utfpr.edu.br

ABSTRACT

License plate detection is a challenging task when dealing with open environments and images captured from a certain distance by low-cost cameras. In this paper, we propose an approach for detecting license plates based on a convolutional neural network which models a function that produces a score for each image sub-region, allowing us to estimate the locations of the detected license plates by combining the results obtained from sparse overlapping regions. Experiments were performed on a challenging benchmark, containing 4,070 license plates in 1,829 images, captured under several weather conditions. The proposed approach achieved a precision of 0.87 and recall of 0.83, outperforming a state-of-the-art detector — a promising result, given that the experiments were performed on single images, without any kind of preprocessing or temporal integration.

Index Terms— license plate location, convolutional neural networks, object detection, traffic surveillance.

1. INTRODUCTION

Surveillance cameras are widely spread in most cities, being an important tool for monitoring urban mobility. However, as pointed out by Zheng [1] in 2014, it is still a challenging task to automatically turn this huge volume of videos into useful data for urban computing applications. The difficulties include cluttered backgrounds, motion blur, proper camera setting and positioning, shadows, and variations in weather and illumination. License plate detection is an essential component for many applications such as speed measurement, vehicle recognition, traffic flow estimation, security control, automatic vehicle ticketing, etc. This problem has been previously addressed in several ways [2, 3, 4, 5, 6, 7, 8]. However, there is a lack of research on license plate detection in complex scenes captured in a open environment by a single, low-cost overhead camera, positioned so as to record multiple vehicles in several lanes, as exemplified in Fig 1.

Fig. 1. Example of image captured by a traffic surveillance system.

In this paper, we propose an approach for detecting license plates based on convolutional neural networks (CNNs). CNNs are a type of connectionist system inspired by the mechanism of the animal visual cortex, implemented as a series of layers which perform convolutions [9]. In recent years, research on CNNs has been progressing quickly, leading to considerable advancements, especially for object detection, recognition, and classification [10, 11, 12, 13].

The approach introduced in this paper is summarized in Fig. 2[1]. It takes as input an image, which may or may not contain vehicles. The image is partitioned in sub-regions, which form an overlapping grid. Each sub-region is independently fed to a CNN, which produces, for each sub-region, a single value in the $[0, 1]$ range. Ideally, the output is 1 when the sub-region contains a nearly centered license plate, and 0 when it contains less than half of a license plate, with other situations producing intermediate values. By analyzing the outputs of neighboring sub-regions, it is possible to estimate the location of each detected license plate. The training samples for the CNN were extracted from video sequences.

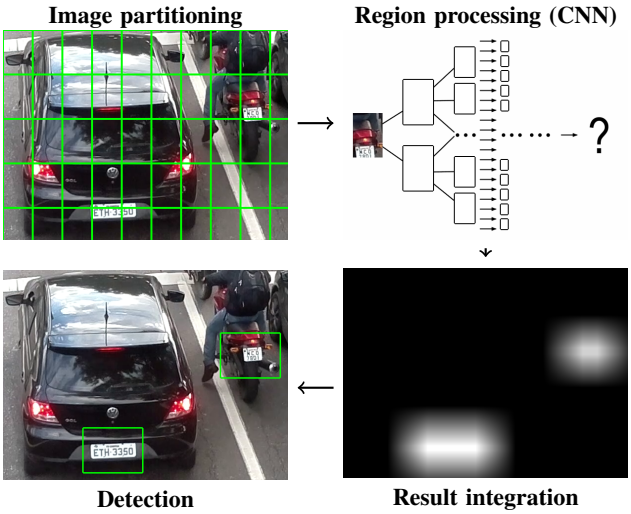


Fig. 2. Steps of the proposed approach.

The main contributions of this paper are: (1) the design of a robust CNN-based license plate detector; (2) the creation of an output function that allows combining the results obtained from a subset of image sub-regions, and which can be employed for other object detection tasks; and (3) the development of a challenging image benchmark, freely available for research purposes[2]. The dataset was

[1] A demonstration video can be watched on https://goo.gl/drd8W7
[2] http://www.dainf.ct.utfpr.edu.br/%7erminetto/projects/license-plate/

captured under various weather and recording conditions by a fixed overhead camera, positioned so that the rear license plates of vehicles in three adjacent lanes are clearly visible. It contains 1,829 full-HD images with 4,070 license plates associated with *ground truth files*, in a simple XML format.

The rest of this paper is organized as follows. In Sec. 2 we discuss some related work. The proposed approach is detailed in Sec. 3. Experiments and their results are discussed in Sec. 4. Finally, Sec. 5 concludes the paper and points to future work.

2. RELATED WORK

License plate detection is an active field of research with many algorithms and extensive bibliography. The surveys from Anagnostopoulos *et al.* [2] and Du *et al.* [3] cover some recent advances up to 2013. Algorithms for text detection in urban scenes [14] can also be used, although they are designed to solve a more general problem. License plates have their design standardized and regulated, so they have well known sizes, shapes, and color schemes. Moreover, the characters must have a strong contrast with the background. That leads to distinctive image features that can be explored by detection algorithms, such as edges, texture, color, and geometry.

Algorithms for license plate detection can be classified into two categories. A bottom-up algorithm first attempts to identify numbers/characters, which are then grouped to produce license plate candidates. A top-down algorithm first attempts to find image regions that are *license plate-like*, and then refine those regions by removing unnecessary background. Bottom-up algorithms generally use image segmentation techniques such as edge detection [15], stroke width [16], maximally stable extremal regions (MSER) [17, 5], saliency features [18] and mathematical morphology [15] to recover the license plate characters. Top-down algorithms typically use a sliding-window strategy, and rely on texture features such as haar wavelet coefficients [19], Fourier spectrum [20], or the scale-invariant feature transform (SIFT) [19]. Both strategies can explore properties such as region size, shape, aspect ratio, and geometric organization. Classifiers such as Support Vector Machines (SVM) or Artificial Neural Networks are also widely used to classify the extracted features. One must note that some features (especially color) are sensitive to illumination variations, so some approaches also require special lighting [2, 3].

In recent years, convolutional neural networks (CNN) [9] have been successfully used in many challenging tasks of object detection and recognition [10, 11, 12, 13]. However, CNN-based approaches for license plate detection are still uncommon. One approach is described by Chen *et al.* [21]. Their method employs a CNN with a single convolutional layer, applied in a sliding window fashion over small image sub-regions. The CNN was trained to classify individual characters as text/non-text, with the outputs being merged and then tested based on aspect ratio and size. Unfortunately, the authors do not provide enough details about the algorithm and the evaluation, so it is not possible to reproduce their method for comparison purposes.

As detailed in Section 4, we have compared our approach with the license plate detector proposed by Luvizon *et al.* [8]. This detector has a bottom-up phase for edge extraction, edge filtering and region grouping to provide coarse candidate regions based on the edge attribute that makes up the license plate. Then, the authors used a specialized gradient-based text descriptor [22, 23] with a SVM classifier to filter false candidate regions.

3. PROPOSED APPROACH

In this section, we detail the proposed approach for license plate detection. Due to space constraints, we do not discuss CNNs in detail — for an overview, please refer to [24]. The sequence of steps was shown in Fig. 2. The following sub-sections describe each step, as well as a method for obtaining training samples from a video.

3.1. Image Partitioning

The detector takes as input an RGB image without any type of normalization or smoothing, other than those automatically executed by the camera. License plates are detected in image sub-regions. The sub-regions must be at least large enough to contain the whole license plate, as well as additional pixels around it, but without more than one license plate appearing in the same sub-region. That way, the CNN will have enough information to learn the difference between the license plate texture and the texture around it, becoming robust against misclassification of scene text as a license plate.

In our implementation, the sub-regions are 120 pixels wide by 180 pixels high. They are evenly spaced, with the space between the centers of neighboring sub-regions being called the *stride*. A *sliding window* approach would use a stride of 1 in both directions, resulting in a very large number of regions. Our detector uses a horizontal stride of 60 pixels and vertical stride of 90 pixels, corresponding to half the sub-region width and height, respectively. That way, the sub-regions form an overlapping grid.

3.2. Region Processing

The image sub-regions are independently fed to a CNN, which was designed to produce outputs that, when taken together, allow estimating the position of a detected license plate. A single value in the $[0, 1]$ range is produced for each sub-region. If the sub-region contains a centered license plate, the CNN output is 1, remaining with this value if the image is translated while keeping the license plate center inside a rectangle with half the width and height of the sub-region. The value then smoothly decreases from 1 to 0 as the plate center moves further outside, reaching 0 when it leaves the sub-region. That way, the CNN models a function whose behavior is shown in Fig. 3.

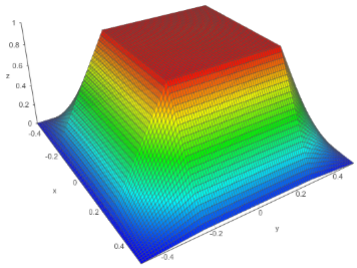


Fig. 3. 3D plot of the function the CNN must implement, showing how the translation of the plate affects its numeric output.

The precise definition of the function is obtained by considering each direction separately. Taking the translation in the x axis while

keeping the license plate centered in y, the function is:

$$f_x(\Delta x) = \begin{cases} 1 & \text{if} \quad |\Delta x| \leq 0.25w \\ (0.5w - |\Delta x|)/0.25w & \text{if} \quad 0.25w < |\Delta x| < 0.5w \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where w is the region width, and Δx is the distance in pixels between the center of the license plate and the center of the region. An equivalent equation is applicable when translating in the y axis while keeping the plate centered in x. The combined function is then:

$$f(\Delta x, \Delta y) = f_x(\Delta x) \cdot f_y(\Delta y) \tag{2}$$

The topology of our CNN is shown in Table 1. The input is a 3-channel sub-region with 180x120 pixels. There are 9 parametric layers. The first two layers use a larger stride to reduce the size of the image and reduce computation time, the other layers use stride 1×1. Layers M3 through M7 contain most of the trainable parameters. The last two layers are fully-connected. All layers use rectified linear unit (ReLu) activation, except the last layer, which uses identity.

Table 1. Convolutional Neural Network: [CV] convolution, [DOF] Degrees Of Freedom, [MP] MaxPool 2×2, [FC]: fully-connected neuron, [LIN] linear neuron.

ID	Input	Operation	DOF
M1	$180 \times 120 \times 3$	$2 \times$ CV 3×3, stride 3×3	56
M2	$60 \times 40 \times 2$	$12 \times$ CV 3×3, stride 3×3	228
M3	$20 \times 14 \times 12$	$16 \times$ CV 3×3	1,744
M4	$20 \times 14 \times 16$	$16 \times$ CV 3×3	2,320
M5	$20 \times 14 \times 16$	$24 \times$ CV 3×3, MP 2×2	3,480
M6	$10 \times 7 \times 24$	$16 \times$ CV 3×3	3,472
M7	$10 \times 7 \times 16$	$2 \times$ CV 3×3, MP 2×2	290
-	$5 \times 4 \times 2$	*flatten*	-
FC1	40	$32 \times$ FC	1,312
FC2	32	$1 \times$ FC LIN	33
OUT	1		-
Total			12,935

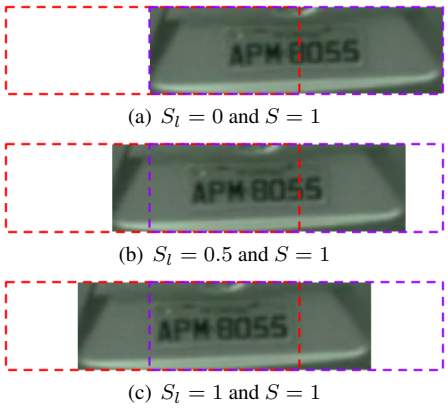
3.3. Result Integration

Once the CNN outputs are computed for all image sub-regions, we combine the results to estimate the location of each license plate. Sub-regions whose outputs are local maxima and greater than a given threshold are taken as candidate license plate regions. For each selected candidate sub-region with score S, we also take its left or right neighbor — whichever had the largest score. The largest the score for the neighbor, the furthest the plate center is displaced to the direction of that neighbor. We move the candidate region in the direction of its neighbor. The offset in the x axis is computed by:

$$\text{offset}_x = \begin{cases} -S_l/4 \cdot w & \text{if} \quad S_l > S_r \\ +S_r/4 \cdot w & \text{otherwise} \end{cases} \tag{3}$$

where S_l and S_r are the output scores for the left and right neighbor, respectively, and w is the sub-region width. If both S_l and S_r are 0, the plate is already centered, and $\text{offset}_x = 0$. The offset in the y direction is computed in an equivalent manner, but considering the neighbors above and below the sub-region, and the region height h.

Figure 4 shows the CNN outputs for two neighboring sub-regions as an image with a license plate is translated from being centered on one of the sub-regions up to the point where it is centered on the other. The offset is computed based on this behavior.



(a) $S_l = 0$ and $S = 1$

(b) $S_l = 0.5$ and $S = 1$

(c) $S_l = 1$ and $S = 1$

Fig. 4. CNN output scores for two neighboring partitions as an image with a plate is translated to the left.

3.4. Obtaining Training Samples

The CNN is trained using supervised learning, and requires many labeled samples. We obtain a balanced set of training samples from video sequences by manually labeling selected frames. Positive samples with centered license plates are directly obtained from the manual annotations. Samples containing off-center license plates are obtained by shifting the manually selected regions by small random amounts. Negative samples are extracted from regions not marked as license plates. We took care not to select negative samples from similar regions (e.g. showing only asphalt), by avoiding taking samples too close from each other, as well as samples that are too similar, given a similarity metric. To add variety to the samples, they are also shuffled and distorted by noise and random brightness and saturation changes. For each training sample, the corresponding CNN output was computed as discussed in Sec. 3.2.

4. EXPERIMENTS

The experiments were carried out on an Intel Core i5 machine (2.4 GHz) with 8 GiB of RAM running Linux with a GT-740M GPU. The implementation was in Python, using the TensorFlow library.

4.1. Dataset

Our dataset, summarized in Table 2, contains 1,829 images with 1920×1080 pixels, captured from 5 videos, recorded by a low-cost 5-megapixel CMOS sensor, under different weather conditions. Each image has a *ground truth* obtained by manual labeling. The training samples were taken from a disjoint subset of the dataset, with 380 images, resulting in a total of 294,602 examples.

4.2. Metrics

To evaluate the algorithm's performance we compared the detected license plates with those in the ground truth. Objectively,

Table 2. Dataset information. The quality options are: [H] high-quality, [N] frames affected by natural or artificial noise, [L] frames affected by severe lighting conditions, [B] motion blur, and [R] rain.

Set	Number of images	Number of plates	Quality
1	119	254	[H]
2	209	437	[L]
3	435	875	[N]
4	331	658	[N,R]
5	735	1,846	[L,B]
Tot.	1,829	4,070	

we compute precision $p = \left(\sum_{i=1}^{|D|} m(d_i, G)\right)/D$ and recall $r = \left(\sum_{i=1}^{|G|} m(g_i, D)\right)/|G|$ metrics, where $G = \{g_1, g_2, \ldots, g_{|G|}\}$ is the set of ground truth license plate regions, $D = \{d_1, d_2, \ldots, d_{|D|}\}$ is the set of detected license plates, and m is defined by

$$m(a, S) = \max_{i=\{0,\ldots,|S|\}} m'(a, s_i) \tag{4}$$

m' is a function that compares two rectangular regions a and b, and is defined as in the PASCAL Visual Object Detection Challenge [25]

$$m'(a, b) = \begin{cases} 1 & \text{if } \dfrac{\text{area}(a \cap b)}{\text{area}(a \cup b)} > 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

We also consider the *F-measure*: $F = 2 \cdot p \cdot r/(p + r)$.

4.3. Evaluation

We compared our approach with a state-of-the-art license plate detector [8], described in Sec. 2. The performance and average execution time of both algorithms are shown in Table 3. Our experiments indicate that the proposed CNN license plate detector is faster and more robust. The algorithm of Luvizon *et al.*missed a lot of small plates because the Sobel edge detector did not detect sufficient image edges over those regions. We show in Fig. 5 samples of license plates found by our algorithm — many of such license plate regions have unreadable characters and very poor image quality.

We noted that the bounding boxes of some CNN detections are not tight enough, as shown in two examples of Fig. 5. For this reason, if we consider an overlap of 0.3 (instead of 0.5) in Eq. 5, we increase our average F-measure to **0.91**.

Table 3. License plate detection performance evaluation, based on precision (p), recall (r), F-measure and time (t) in seconds.

	PROPOSED				LUVIZON *et al.* [8]			
Set	p	r	F	t(s)	p	r	F	t(s)
1	**0.86**	**0.87**	**0.87**	**0.23**	0.74	0.68	0.71	2.3
2	**0.86**	**0.82**	**0.84**	**0.23**	0.73	0.52	0.61	2.2
3	**0.87**	**0.83**	**0.85**	**0.23**	0.74	0.66	0.70	2.4
4	**0.89**	**0.84**	**0.86**	**0.23**	0.84	0.61	0.71	2.3
5	**0.86**	**0.78**	**0.82**	**0.23**	0.76	0.53	0.63	2.4
Total	**0.87**	**0.83**	**0.85**	**0.23**	0.76	0.60	0.67	2.3

Figure 6 shows the result of the proposed CNN classifier, by using a 180×120 window, sliding over the image from Fig. 1. Note the high selectivity of our approach.

Fig. 5. Representative samples of license plates detected by our CNN algorithm — images with good quality, different color schemes, motorcycles, noise, irregular illumination, motion blur, occlusion, and poor license maintenance.

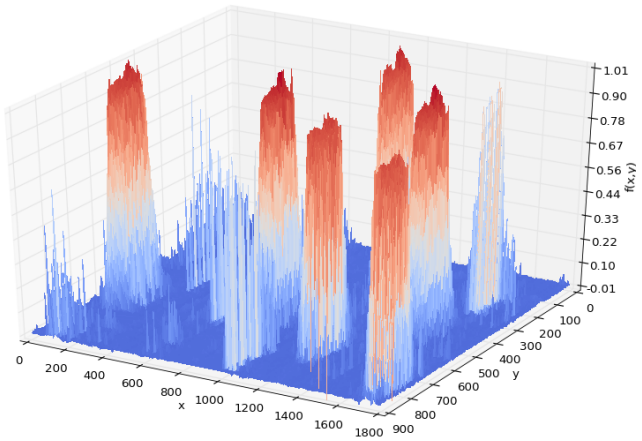


Fig. 6. The output of proposed CNN sliding window classifier (stride = 4×4), over the image shown in Fig. 1, produces a heat map where warm tones denote positive output and cold tones negative outputs.

5. CONCLUSION AND FUTURE WORK

In this paper, we proposed a CNN-based approach for detecting license plates, which uses a novel output function that allows us to estimate the locations of the detected plates by combining the results obtained from sparse overlapping regions, reducing computation time. The proposed approach achieved promising results when compared to a state-of-the-art detector. Besides the license plate detection approach itself, we highlight as contributions the CNN output function, which can be employed for other object detection tasks, and the image benchmark, which is freely available for research.

Future work include exploring the temporal aspect of videos to detect regions containing motion and to track the detected license plates, increasing robustness to false positives and negatives.

6. ACKNOWLEDGMENT

This work was partially supported by the Brazilian National Council for Scientific and Technological Development (CNPq), the Coordination for the Improvement of Higher Education Personnel (CAPES) and/or the Araucaria Foundation.

7. REFERENCES

[1] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang, "Urban computing: Concepts, methodologies, and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 3, pp. 38:1–38:55, Sept. 2014.

[2] C.E. Anagnostopoulos, I.E. Anagnostopoulos, I.D. Psoroulas, V. Loumos, and E. Kayafas, "License plate recognition from still images and video sequences: A survey," *IEEE Transactions on Intelligent Transportation Systems (T-ITS)*, vol. 9, no. 3, pp. 377–391, 2008.

[3] S. Du, M. Ibrahim, M. Shehata, and W. Badawy, "Automatic license plate recognition (ALPR): A state-of-the-art review," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 2, pp. 311–325, 2013.

[4] M. Garg and S. Goel, "Real-time license plate recognition and speed estimation from video sequences," *Transactions on Electrical and Electronics Engineering*, vol. 1, no. 5, pp. 1–4, 2013.

[5] Bo Li, Bin Tian, Ye Li, and Ding Wen, "Component-based license plate detection using conditional random field model," *IEEE Transactions on Intelligent Transportation Systems (T-ITS)*, vol. 14, no. 4, pp. 1690–1699, 2013.

[6] A.H. Ashtari, M.J. Nordin, and M. Fathy, "An iranian license plate recognition system based on color features," *IEEE Transactions on Intelligent Transportation Systems (T-ITS)*, vol. 15, no. 4, pp. 1690–1705, 2014.

[7] W. Zhou, Houqiang Li, Yijuan Lu, and Qi Tian, "Principal visual word discovery for automatic license plate detection," *IEEE Transactions on Image Processing (TIP)*, vol. 21, no. 9, pp. 4269–4279, 2012.

[8] D. C. Luvizon, B. T. Nassu, and R. Minetto, "A video-based system for vehicle speed measurement in urban roadways," *IEEE Transactions on Intelligent Transportation Systems (T-ITS)*, vol. PP, no. 99 (Early Acesss), pp. 1–12, 2016.

[9] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[10] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei, "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[11] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick, "Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2874–2883.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[13] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

[14] Q. Ye and D. Doermann, "Text detection and recognition in imagery: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 37, no. 7, pp. 1480–1500, July 2015.

[15] J. L. Tan, S. A. R. Abu-Bakar, and M. M. Mokji, "License plate localization based on edge-geometrical features using morphological approach," in *IEEE International Conference on Image Processing (ICIP)*, Sept 2013, pp. 4549–4553.

[16] Jobin K. V., Jiji C. V., and Anurenjan P. R., "Automatic number plate recognition system using modified stroke width transform," in *National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics*, Dec 2013, pp. 1–4.

[17] J. Matas and K. Zimmermann, "Unconstrained licence plate and text localization and recognition," in *IEEE Intelligent Transportation Systems*, Sept 2005, pp. 225–230.

[18] K. H. Lin, H. Tang, and T. S. Huang, "Robust license plate detection using image saliency," in *IEEE International Conference on Image Processing (ICIP)*, Sept 2010, pp. 3945–3948.

[19] W. T. Ho, H. W. Lim, and Y. H. Tay, "Two-stage license plate detection using gentle adaboost and sift-svm," in *2009 First Asian Conference on Intelligent Information and Database Systems*, April 2009, pp. 109–114.

[20] R. Al-Hmouz and K. Aboura, "License plate localization using a statistical analysis of discrete fourier transform signal," *Computers and Electrical Engineering - Elsevier*, vol. 40, no. 3, pp. 982 – 992, 2014.

[21] Ying-Nong Chen, Chin-CHuan Han, Cheng-Tzu Wang, Bor-Shenn Jeng, and Kuo-Chin Fan, "The application of a convolution neural network on face and license plate detection," in *International Conference on Pattern Recognition (ICPR)*, 2006, pp. 552–555.

[22] R. Minetto, N. Thome, M. Cord, N.J. Leite, and J. Stolfi, "T-hog: An effective gradient-based descriptor for single line text regions," *Pattern Recognition - Elsevier*, vol. 46, no. 3, pp. 1078–1090, 2013.

[23] R. Minetto, N. Thome, M. Cord, J. Fabrizio, and B. Marcotegui, "Snoopertext: A multiresolution system for text detection in complex visual scenes," in *IEEE Int. Conf. on Image Processing (ICIP)*, 2010, pp. 3861–3864.

[24] Yann Lecun, Yoshua Bengio, and Geoffrey Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 5 2015.

[25] Mark Everingham, Luc Van Gool, C. K. I. Williams, J. Winn, and Andrew Zisserman, "The PASCAL Visual Object Classes (VOC) challenge," 2009.