Training-Free License Plate Detection Using Vehicle Symmetry and Simple Features

Sheng WANG, Qiang WU, Wenjing JIA, and Xiangjian HE
School of Computing and Communications
University of Technology, Sydney
Sydney, NSW 2007, Australia

Abstract—**In this paper, we propose a training free license plate detection method. We use a challenging benchmark dataset for license plate detection. Unlike many existing approaches, the proposed approach is a training free method, which does not require supervised training procedure and yet can achieve a reasonably good performance. Our motivation comes from the fact that, although license plates are largely variant in color, size, aspect ratio, illumination condition and so on, the rear view of vehicles is mostly symmetric with regard to the vehicle's central axis. In addition, license plates for most vehicles are usually located on or close to the vertical axis of the vehicle body along which the vehicle is nearly symmetric. Taking advantage of such prior knowledge, the license plate detection problem is made simpler compared to the conventional scanning window approach which not only requires a large number of scanning window locations, but also requires different parameter settings such as scanning window sizes, aspect ratios and so on.**

I. Introduction

License Plate Detection (LPD) in object detection research is highly application oriented, and is widely used for various purposes. In this paper, we propose a novel license plate detection framework. Different from many previous methods for license plate detection, our approach does not require supervised learning. Moreover, compared with those approaches which need to classify a scanning detection window as either a license plate or non-license plate, our approach only requires a fairly small amount of scanning detection windows. Specifically, we have reduced the computational complexity from $O(M \times N)$ to $O(M + N)$, where M and N are the height and width of the input image.

There have been many approaches published for license plate detection. In general, those approaches can be divided into two main categories, i.e., learning based approaches and non-learning based approaches. The method proposed in this paper belongs to the latter category, i.e., non-learning based approaches. However, the difference between our approach and many other non-learning based approaches is that we have embedded high level semantic information into the low level feature extraction algorithm. In this way, not only the feature extraction process is made faster, but also the process of parameter tuning is simplified.

In [2], the authors proposed a segmentation based license plate detection method using the Mean Shift algorithm for image segmentation. Three features were defined, i.e., rectangularity, aspect ratio, and edge density. Then, a classifier based on Mahalanobis distance was used for classification. The edge density features are also used in [3] and [4] as a simple thresholded filter in order to further speed-up the detection process which is based on a supervised learning framework.

In [5], the authors proposed a non-learning based license plate detection framework using the TO-MACH filter, a powerful correlation filter algorithm which has been extensively used to recognize distorted objects.

In [6], the authors used the Maximally Stable Extremal Region (MSER) for license plate detection. They also utilized some prior knowledge for the detection and localization, such as the constraints on the gray level intensity changes.

In [7], the authors proposed a system based on DSP-platform, which is capable of performing both license plate detection and license plate recognition.

In [8], the authors used a multi-stage approach for license plate detection. Edge images were first extracted. Then, histogram analysis together with compact factor measurement was used on the extracted edge images to determine potential candidate license plates. Finally, morphological operations were performed for final detection.

In [9], the author presented a new license plate detection method based on information in the *Lab* color space. The proposed method first categorized the candidate region into blue plates or yellow plates (two major categories of license plates) in order to select the corresponding threshold matrix for optimal parametrization. Then, based on certain prior knowledge, the *Lab* space was used instead of gray level in order to better locate potential license plate regions.

In [10], the authors reported a license plate detection algorithm using the image saliency. They performed experiments on a mixed dataset from four different databases (including the Caltech Car markus dataset).

Through this paper, our contributions are summarized as follows.

Firstly, we propose a novel training-free license plate detection method which aims at detecting license plates from natural scene images. The novelty of our method, however, is not only the training-free nature of the algorithm, but also those assumptions which are related to the particular practical problem of license plate detection. Those assumptions can be considered as prior knowledge, and the connection between this method and other supervised learning methods is that, in supervised learning methods, the prior knowledge is obtained from training samples through machine learning algorithms in an (usually) iterative manner, whilst in our work, the prior knowledge is well known and manually fitted into the framework.

Secondly, although our approach aims at detecting license plates, it is not limited to license plate detection, because object symmetry is widely observed on most artificial (human made) objects, such as vehicles and buildings. Moreover, many natural objects, such as butterfly, tree leaf, and flower, are also symmetric. Hence, it is reasonable to explore object symmetry in order to develop object detection algorithms with better performance. From another point of view, we formulate the high level semantic information, i.e., the object symmetry, into an object detection framework. Without using such high level semantic information, the object detection framework can only rely on a strong learning algorithm (e.g., AdaBoost algorithm) to perform selection of low level image features for object detection [11]. With this high level semantic information (latent features derived from object symmetry), a well trained strong classifier is no longer needed and the computational cost can be significantly reduced.

Thirdly, we propose a novel modification to the conventional Edgelet feature extraction algorithm. Instead of predefining edgelets manually, we use a (left to right) flipped part of the input image to form an edgelet. Such an edgelet is compared with the other part of the image in order to calculate an affinity value. The affinity value is later used to determine the degree of symmetry of the input image on the compared pair of image regions (parts).

The rest of this paper is organized as follows. Section II illustrates the framework of the proposed license plate detection algorithm. Section III provides the experimental results and some discussions. Section IV concludes this paper.

II. The Proposed Framework

As mentioned earlier, our motivation comes from two facts. The first fact is that the rear views of most vehicles are horizontally symmetrical. This implies that we may be able to find a symmetric axis in the vertical direction for most of rear view vehicle images, under the assumption that rear views of those vehicles are well aligned. The second fact is that most license plates are mounted at a predictable location of the vehicle body. For instance, almost all license plates are located at the central or lower central part of a vehicle body.

The first fact gives us the pixel location of a license plate along the horizontal axis and the second fact tells the pixel location of a license plate along the vertical axis. Together, the pixel coordinate of a license plate can be determined. In our framework, we set the top left corner of an image as the origin, and all pixel coordinates are positive integers.

Given an uprightly aligned rear-view vehicle image, a simple method to detect a symmetric axis is to perform subtraction over the image (i.e., to subtract the left half from the right half), which is similar to fold up a paper when we try to find the symmetric axis of the paper. However, in our case, it is not guaranteed that the horizontal edges of a vehicle are accurately aligned with the horizontal axis of an input image. Moreover, different input images may have different levels of subtle left-right rotations. As a result, using a simple subtraction technique may fail to detect the symmetric axis.

In our approach, we use the low level feature (Edgelet) proposed by [1] for symmetric axis detection. Although the Edgelet feature is initially proposed for human detection, we find that it can serve our purpose well. In [1], different types of Edgelet templates (line, curve and hyperbola) were manually defined with prior knowledge of the shape of different human body parts, and then a supervised learning method was used for human detection. In this paper, we do not manually define any Edgelet template, because the pattern of vehicles' rear views are largely unpredictable.

In order to proceed, we introduce the following symbols and definitions.

The index i_{max} for the symmetric axis $\mathbf{V}$, which is parallel to the vertical direction, can be represented by

$$i_{max} = \arg\max_{i} f(\mathbf{I}(:, i-k:i), \mathbf{I}(:, i:i+k)). \tag{1}$$

In (1), i represents the index (pixel location) for the symmetric axis, which is parallel to the vertical (Y) axis, and k represents the width of the 'folded' image region, which should be identical to the other part (image region which lays flat on the desk with width k). The function $f(\cdot,\cdot)$ calculates the affinity value between two image regions. Please refer to [1] for the definition of affinity value. $\mathbf{I}(:, i-k:i)$ represents a sub-image of image $\mathbf{I}$, with all the rows and column $i-k$ to column i of $\mathbf{I}$. $\mathbf{I}(:, i:i+k)$ represents a sub-image of image $\mathbf{I}$, with all the rows and column i to column $i+k$ of $\mathbf{I}$. This definition is consistent with the Matlab syntax.

Intuitively, the value of k should be large enough so that at least the left half of the vehicle can be 'flipped' to match against the right half in order to calculate an affinity value. However, if k is too large, some of the background will also get 'flipped' and cause undesirable effects. Moreover, some vehicles may be too close to the image boundary for a larger k which likely results in the algorithm not working well. In our experiments, k is determined empirically.

For notation simplicity, denote the left part of such an image region as $\mathbf{I}_1$ and its right part as $\mathbf{I}_2$, i.e.,

$$\mathbf{I}_1 = \mathbf{I}(:, i-\frac{m}{2}:i) \tag{2}$$

and

$$\mathbf{I}_2 = \mathbf{I}(:, i:i+\frac{m}{2}) \tag{3}$$

where m represents the width of the candidate plate region.

When $\mathbf{V}$ is detected, the next step is to determine potential license plate region(s) among all those image regions along $\mathbf{V}$ with a width of m. In our approach, we use a modified version of the Vertical Edge Density Variance (VEDV) feature described in [3] for this task. Instead of considering the density variance, we sum up all the density values for different sub-blocks and use the total value as a feature, referred to as Summation of Vertical Edge Density (VED^{Σ}). The image region which gives maximum response of VED^{Σ} is considered as a candidate license plate region. Please refer to [3] for more details about the Vertical Edge Density Variance feature and the definition of a sub-block.

As its precedent, the proposed VED^{Σ} feature is also a weak feature. In our experiments, we have found that the VED^{Σ} feature is sensitive to plant textures. In order to obtain better detection performance, we again make use of high level semantic information. In the Caltech Car markus dataset, all vehicles' rear views are complete. This implies that all license plates can be located at the middle or lower part of the image (top part of the image are either vehicle rear window or cluttered background). In order to make use of this information, we introduce one more parameter u, which represents the smallest possible index value for a potential license plate (recall that we are using the top-left corner as the origin). A larger u can remove more potential false positive regions. However, if u is too large, there will be missing detection of plates. On the other hand, a smaller u may not be sufficient to reduce the sensitivity of the VED^{Σ} feature. We demonstrate through our experiments that, introducing u can improve the detector's performance. This again proves the validity of using prior knowledge to complement low level image features for better performance.

Similar to (1), define j_{max} as

$$j_{max} = \arg\max_{j} VED^{\Sigma}(\mathbf{I}_1(j:j+l,:), \mathbf{I}_2(j:j+l,:)). \quad (4)$$

where j_{max} is the index of the horizontal line $\mathbf{H}$, which overlaps with the top boundary of a potential license plate.

In (4), j represents the pixel location on Y axis where the maximum VED^{Σ} value is obtained, l determines the height of the image patch which is considered as a potential license plate, $\mathbf{I}_1(j:j+l,:)$ is a sub-image of $\mathbf{I}_1$ with row j to row $j+l$ and all columns, and $\mathbf{I}_2(j:j+l,:)$ is a sub-image of $\mathbf{I}_2$ with row j to row $j+l$ and all the columns. In our experiments, similar to k, l and m are also empirically determined.

Finally, given i_{max} and j_{max}, representing the horizontal and vertical indices of a potential license plate, the pixel coordinates of the license plate's top middle point can be written as:

$$Intersection(\mathbf{V}, \mathbf{H}) = (i_{max}, j_{max}). \quad (5)$$

Details of our symmetry detection algorithm with Edgelet feature is illustrated in Fig. 1. Note that the algorithm in Fig. 1 only describes how to detect the symmetric vertical axis in an image (example).

In Fig. 1, function $flip_{(left\ to\ right)}(\mathbf{I})$ flips an image $\mathbf{I}$ horizontally. Function $max(\mathbf{A})$ gives the exact value and index of the maximum element of matrix $\mathbf{A}$. In the algorithm, the exact value is represented by $maxval_A$ and the index is represented by $maxind_A$. The function $f(I_2, I_1')$ calculates an affinity value that describes how similar two image patches of the same size I_2 and I_1' are. Detailed algorithm for function $f(\cdot,\cdot)$ is given in Fig. 2.

$Input : k, \mathbf{I}.$
$Output : \mathbf{A} = [a_1, a_2, ...a_N], maxval_A, maxind_A.$
$[Height, Width] \leftarrow size(\mathbf{I})$
$N \leftarrow Width - 2 * k$
for $i = k+1 \rightarrow Width - k$ **do**
  $\mathbf{I_1} \leftarrow \mathbf{I}(:, i-k : i)$
  $\mathbf{I_2} \leftarrow \mathbf{I}(:, i : i+k)$
  $\mathbf{I_1'} \leftarrow flip_{(left\ to\ right)}(\mathbf{I_1})$
  $\mathbf{A}(i) \leftarrow f(\mathbf{I_2}, \mathbf{I_1'})$
end for
$[maxval_A, maxind_A] \leftarrow max(\mathbf{A})$
Return $maxind_A$ as the index for the symmetric axis

Fig. 1: The algorithm for symmetry axis detection

$Input : \mathbf{I}, \mathbf{J}.$
$Output : a.$
$[Height_I, Width_I] \leftarrow size(\mathbf{I})$
$[Height_J, Width_J] \leftarrow size(\mathbf{J})$
if $(Height_I == Height_J)$ AND $(Width_I == Width_J)$ **then**
  $[\mathbf{Gx_I}, \mathbf{Gy_I}] \leftarrow gradient(\mathbf{I})$
  $[\mathbf{Gx_J}, \mathbf{Gy_J}] \leftarrow gradient(\mathbf{J})$
  $\mathbf{E_I} \leftarrow edge(\mathbf{I})$
  $mx \leftarrow$ max value of $\mathbf{E_I}$
  $\mathbf{E_I'} \leftarrow \frac{\mathbf{E_I}}{mx}$
  $\mathbf{C_I} \leftarrow arctan(\frac{\mathbf{Gy_I}}{\mathbf{Gx_I}})$
  $\mathbf{M_J} \leftarrow \sqrt{\mathbf{Gx_J} * \mathbf{Gx_J} + \mathbf{Gy_J} * \mathbf{Gy_J}}$
  $\mathbf{C_J} \leftarrow arctan(\frac{\mathbf{Gy_J}}{\mathbf{Gx_J}})$
  $\mathbf{aff} \leftarrow \mathbf{E_I'} * \mathbf{M_J} * cos(\mathbf{C_I} - \mathbf{C_J})$
  $a \leftarrow \frac{sum\ of\ all\ elements\ in\ \mathbf{aff}}{sum\ of\ all\ elements\ in\ \mathbf{E_I'}}$
else
  error: **I** and **J** must be equally sized
end if
Return a as the affinity value.

Fig. 2: The algorithm for the affinity value calculation

The $'*'$ symbol in Fig. 2 represents matrix convolution operation.

Our license plate detection result is indicated by the intersection of two lines. Also, in our current framework, each failed detection (missed example) corresponds to one false positive.

III. Experimental Results

We test our framework on the Caltech Car markus dataset, which is composed of 126 images. All 126 images are color images of 896 × 592 pixels in resolution. Among those 126 images, there are 124 images each containing one license plate, and two images having name plates instead of license plates. But for our method, we consider those two images as license plates, because we are not performing License Plate Recognition (LPR) at this stage. All vehicle images are from the rear view. In addition to wild illumination changes and different size of license plates, these vehicle images are very

TABLE I: Parameters used in our experiments

Name	k	l	m	u
Value	300	40	100	100

complex natural images in severely cluttered environments, from plants and sky to buildings and road surface.

Table I lists the detailed parameters used in our experiments. In Table I, k represents the width of the image region which will be compared with another image region of the same size using Edgelet affinity values to measure their similarity. l and m represents the height and width of an image region which will be measured by the VED^{Σ} features to determine if it is a potential plate region. u represents the lower bound for the index of a potential plate region along the vertical direction.

In our experiments, the algorithm in Fig. 1 can successfully detect the symmetric axis for 89.68% of the images (113 out of 126). For all the test images, if we scan the symmetric axis from the top to the bottom, the proposed algorithm achieves a detection rate of 80.16% (101 out of 126) with 25 false positive regions. Under the assumption that all plates are having a vertical index value of greater than or equal to 100 (*i.e.* $u = 100$), the proposed algorithm can achieve a detection rate of 84.92% (107 out of 126) with 19 false positve regions.

All the parameters listed in Table I are determined by experimental results to be the optimal values. We argue that such parameter estimation is inevitable, because in scanning window approach such as [3], one still has to determine the aspect ratio, scaling factor, and scanning frequency for the detection window. However, this approach is not directly comparable with [3], because the dataset being evaluated is different from that of [3]. Although [10] also used the Caltech Car markus dataset for their mixed evaluation dataset, it did not report the performance of the algorithm on each individual dataset. So, direct comparison of our approach with that in [10] is not available.

For those failed examples, in Fig. 3, we can see that the backgrounds of those vehicles are either buildings which also exhibit very high symmetric properties or the vehicles carrying spare tyre which severely disturbs the detector. However, from Fig. 3(l), we can also see that our algorithm can detect a potential license plate even when the characters of the license plate are hardly visible. Fig. 3(l) also indicates that the proposed method have resistance to disturbances caused by non-object of interest, i.e., the spare tyre mounted at the rear of the jeep.

Fig. 4 gives the detailed Edgelet affinity value and VED^{Σ} value for examples (l), (j), (f), and (e) in Fig. 3. In Fig. 4, the dashed red vertical line indicates the actual pixel locations, and the data labels indicate the detection results given by the algorithm. Note that in the Edgelet Affinity value plot (the second column), the index needs to be increased by 300 in order to obtain the correct license plate coordinate, because in our experiments, we empirically set $k = 300$ and $l = 40$. Also, in Fig. 4(l), the horizontal line **H** coincides with the ground truth because there are characters along the detected vertical line. However, the detected vertical line **V** is not the ground truth. Hence, Fig. 4(l) is an example of a false positive.

One may argue that this method assumes that the vehicle itself is symmetric in the captured images, and that the license plate is located along the vertical axis by which vehicle is symmetric. Indeed, in reality, it is not always the case. In the Caltech Car markus dataset, we have found that 123 out of 126 (*i.e.* 97.62%) of the vehicles abide the rules of symmetry (both vehicle symmetry and license plate location symmetry). In reality, it is not practical to figure out exactly how many vehicles are not symmetric and how many license plates are located at asymmetric locations. Nevertheless, we can conclude that in the Caltech Car markus dataset, 97.62% of the vehicles abide the rule of plate symmetry. Another implication is that, given a data set of 126 samples, we can predicate under certain accuracy (probability), that in the Caltech Car markus data acquisization environment, 97.62% of vehicles are symmetric (both vehicle symmetry and license plate location symmetry). The practical applicability of our approach can be assessed once we have corresponding census data available. Such census data can be collected from highway entries or road authorities.

IV. Conclusion

In conclusion, we have proposed a novel training-free license plate detection framework based on high level semantic information, which can be interpreted by two symmetry assumptions, i.e., most of the vehicles are symmetric, and the license plates of most vehicles are located along the vertical symmetric axis.

Given these two assumptions, our aim is to first extract the Edgelet features, then find the maximum affinity value (Edgelet feature value as defined by [1]) in order to locate the vertical symmetry axis. After that, a modified Vertical Edge Density Variance feature, VED^{Σ}, is used to determine the coordinate of a potential license plate.

We have tested our method on a benchmark dataset and the experimental results indicate that our training-free method is reasonably good for a challenging public dataset with complex background in a cluttered environment.

References

[1] B.Wu and R.Nevatia. Detection and Tracking of Multiple, Partially Occluded Humans by Bayesian Combination of Edgelet based Part Detectors. *International Journal of Computer Vision*, 75(2):247–266, 2007.

[2] W.Jia, H.Zhang, X.He, and M.Piccardi. Mean Shift for Accurate License Plate Localization. In:*Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems*, pp. 566–571, 2005.

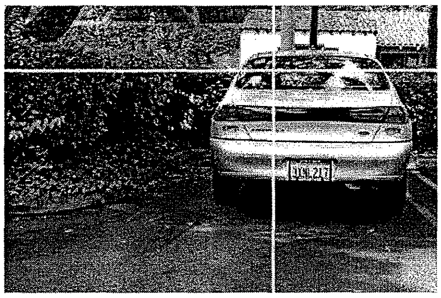
[3] H.Zhang, W.Jia, X.He, and Q.Wu. Learning-Based License Plate Detection Using Global and Local Features. In:*Proc. ICPR*, pp. 1102–1105, 2006.

[4] H.Zhang, W.Jia, X.He, and Q.Wu. A Fast Algorithm for License Plate Detection in Various Conditions. In:*Proc. SMC*, pp. 2420–2425, 2006.

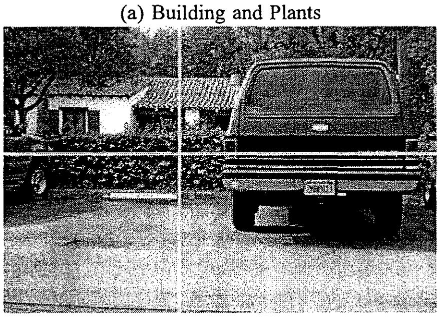
[5] J.Liu, J.Liu, and W.Jin. Robust Vehicle License Plate Localization Using TO-MACH Filter. In:*Proceedings of the 5th International Conference on Computer Science and Education*, pp. 1106–1109, 2010.

(a) Building and Plants

(b) Tree

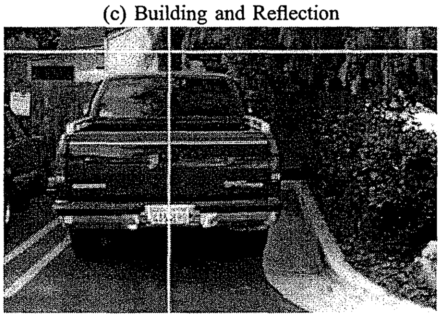


(c) Building and Reflection

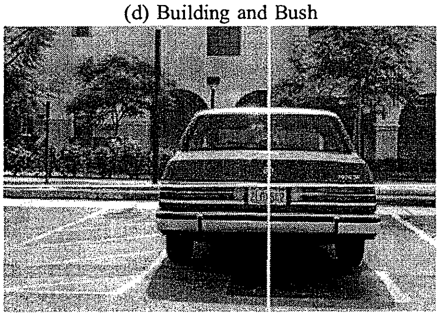


(d) Building and Bush

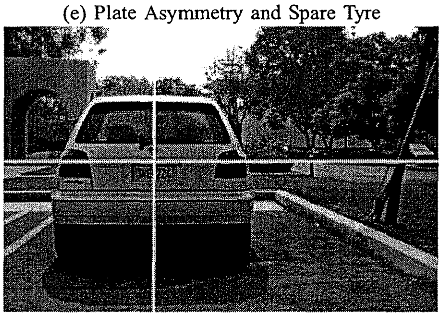
(e) Plate Asymmetry and Spare Tyre



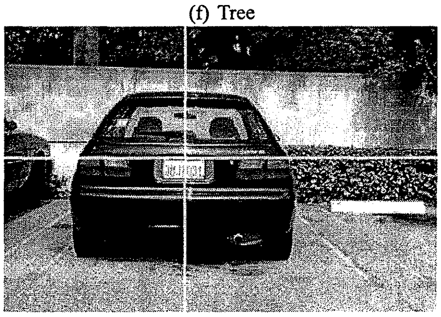
(f) Tree



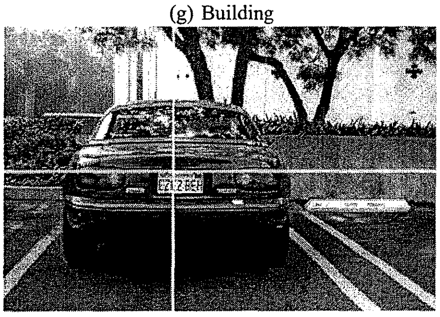
(g) Building



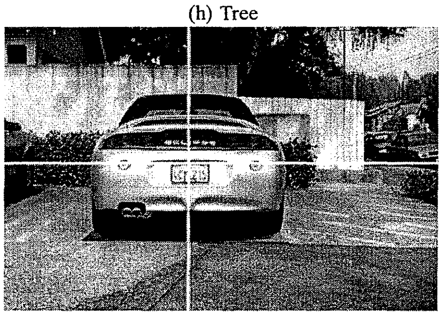
(h) Tree



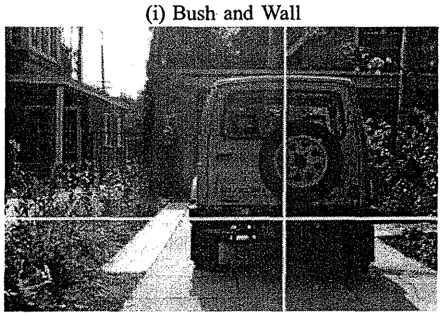
(i) Bush and Wall



(j) Tree



(k) Characters on Car



(l) Illumination and Spare Tyre

Fig. 3: Some examples of detection results. Sub-figure (a) to (f) are failed examples, where the caption in each sub-figure indicates the object or factor which disturbs the detector. Sub-figure (g) to (l) are successful detections, where the caption in each sub-figure indicates the challenging objects or factors posed to the detector.

[6] W.Wang, Q.Jiang, X.Zhou, and W.Wan. Car License Plate Detection Based on MSER. In:*Proceedings of the 2011 International Conference on Consumer Electronics, Communications and Networks*, pp. 3973–3976, 2011.

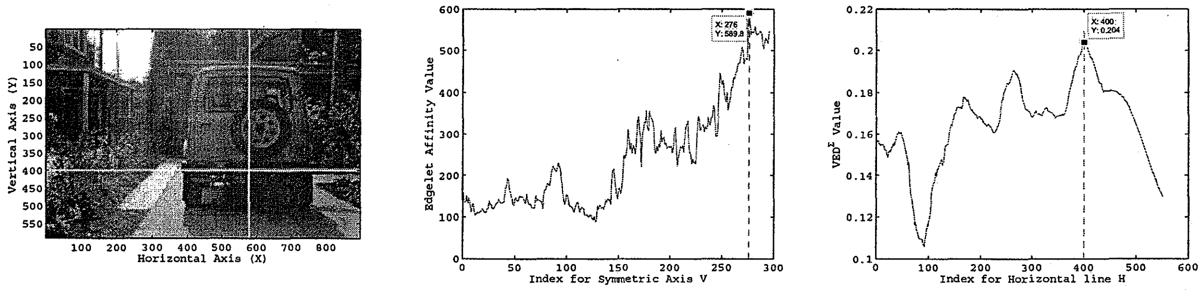
[7] C.Arth, F.Limberger, and H.Bischof. Real-Time License Plate Recognition on an Embedded DSP-Platform. In:*Proc. CVPR*, pp. 1–8, 2007.

[8] F.Faradji, A.H.Rezaie, M.Ziaratban. A Morphological-Based License Plate Location. In:*Proc. ICIP*, pp. I-57–I-60, 2007.

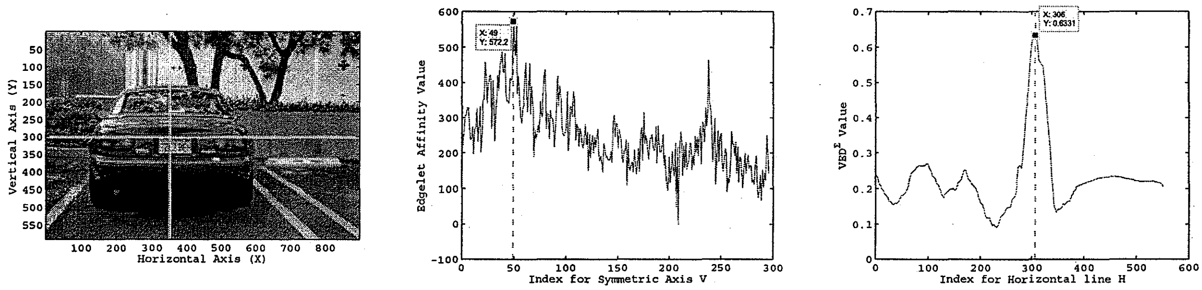
[9] L.Xu. A New Method for License Plate Detection Based on Color and Edge Information of Lab Space. In:*Proceedings of the 2011 International Conference on Multimedia and Signal Processing*, pp. 99–102, 2011.

[10] K.H.Lin, H.Tang, T.S.Huang. Robust License Plate Detection Using Image Saliency. *Proc. ICIP*, pp. 3945–3948, 2010.

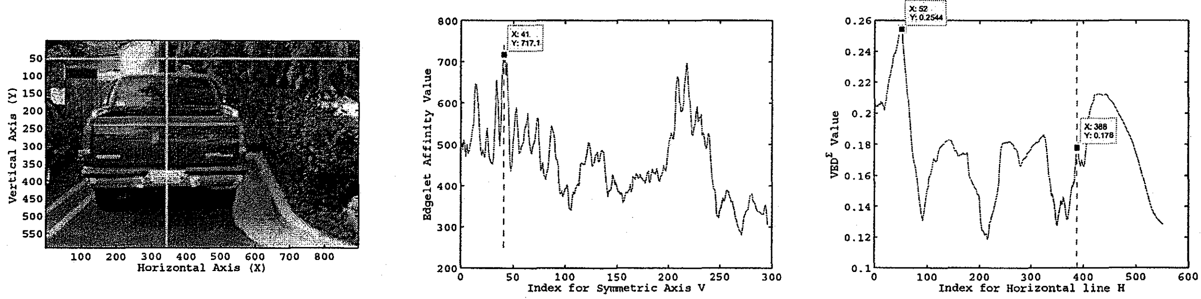
[11] R.E.Schapire, Y.Singer. Improved Boosting Algorithms Using Confidence-rated Predications. *Machine Learning*, 37:297–336, 1999.



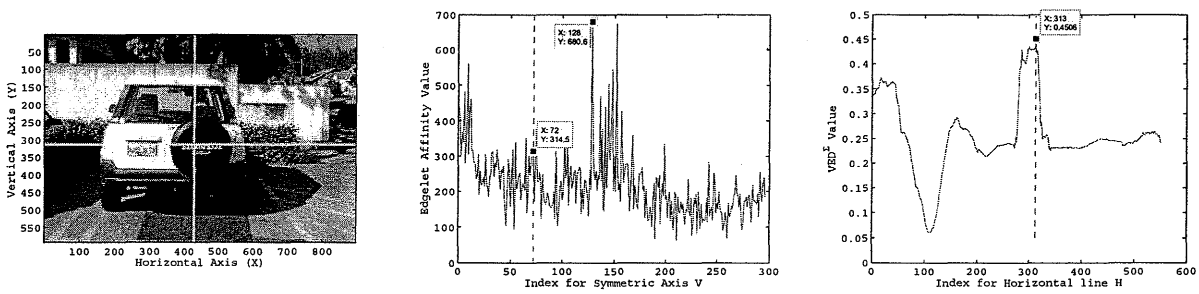
(a) Original Image (b) Edgelet Affinity (c) VED^{Σ}



(d) Original Image (e) Edgelet Affinity (f) VED^{Σ}



(g) Original Image (h) Edgelet Affinity (i) VED^{Σ}



(j) Original Image (k) Edgelet Affinity (l) VED^{Σ}

Fig. 4: Detailed edgelet affinity value and VED^{Σ} value for the examples shown in Fig. 3. Row 1 is for example (l), Row 2 is for example (j), Row 3 is for example (f), and Row 4 is for example (e).