Decision Fusion for Improved Automatic License Plate Recognition

CRISTIAN MOLDER, MIRCEA BOŞCOIANU, IULIAN C. VIZITIU, MIHAI I. STANCIU
Military Technical Academy
Department of Electronics and Informatics
George Coşbuc street 81-83, Bucharest
ROMANIA
cristianmolder@gmail.com http://www.mta.ro

Abstract: Automatic license plate recognition (ALPR) is a pattern recognition application of great importance for access, traffic surveillance and law enforcement. Therefore many studies are concentrated on creating new algorithms or improving their performance. Many authors have presented algorithms that are based on individual methods such as skeleton features, neural networks or template matching for recognizing the license plate symbols. In this paper we present a novel approach for decisional fusion of several recognition methods, as well as new classification features. The classification results are proven to be significantly better than those obtained for each method considered individually. For better results, syntax corrections are also considered. Several trainable and non-trainable decisional fusion rules have been taken into account, evidencing each of the classification methods at their best. Experimental results are shown, the results being very encouraging by obtaining a symbol good recognition rate (GRC) of more than 99.4% on a real license plate database.

Key–Words: Licence Plate, ALPR, Pattern Recognition, Skeleton Features, Neural Networks, Decision Fusion

1 Introduction

The automatic license plate recognition systems (ALPR) exists since the 80s [1] but only in the late 90s became an important application domain of pattern recognition. The information extracted from traffic video sequences is mainly used for access control, parking, motorway road tolling, border control, journey time measurement or for law enforcement. The main problem for those systems is that changing lighting conditions and the need for external illuminators for the video system are of great influence on the good recognition results.

Most ALPR systems are based on six primary algorithms required for identifying a license plate: (1) *plate localization*, responsible for finding and isolating the license plate on the acquired picture, (2) *plate orientation and sizing*, which compensates for the skew of the plate and adjusts the dimensions to the requires processing size, (3) *normalization*, adjusting the brightness and contrast of the image, (4) *character segmentation*, which finds the individual symbol images on the plate, (5) *optical character recognition*, and (6) *syntactical analysis* that checks the characters against specific rules.

Existing approaches vary from simple methods to more complex algorithms. A simple approach is based on image processing techniques used for detecting a license plate boundary in an image [15]. The image is processed to improve the license plate characteristics using algorithms such as convolution masks [10], edge detectors [2], rank filters [21], vertical and horizontal histograms [10], mean shift [6], spatial or spectral transform [12, 20, 11]. These operations can be made in RGB images [1], grayscale [19] or other color systems such as HSV or L*a*b* [1]. The choice of the color system is based on the particularities of each license plate standard such as the plate background color. Other approaches are based on finding characters in images, such as periodograms [1, 8], neural networks [3], or fuzzy logic [4], but are more sensitive to other non-license plate characters present in the image. Generally, if *a priori* information is used one can obtain better results than if this kind of informations isn't exploited. Once the license plate detected and extracted, skew and rotation corrections are made, usually by using the Radon transform [1, 8, 7].

The extracted license plate image is further used for symbol segmentation within. By far, the most used method is the search using vertical and horizontal projections [18, 2, 8, 25, 14].

The next step is the use of optical character recognition methods for recognizing the segmented symbols, with features extracted from symbol projections [1], matching templates [2, 7], skeletons [9],

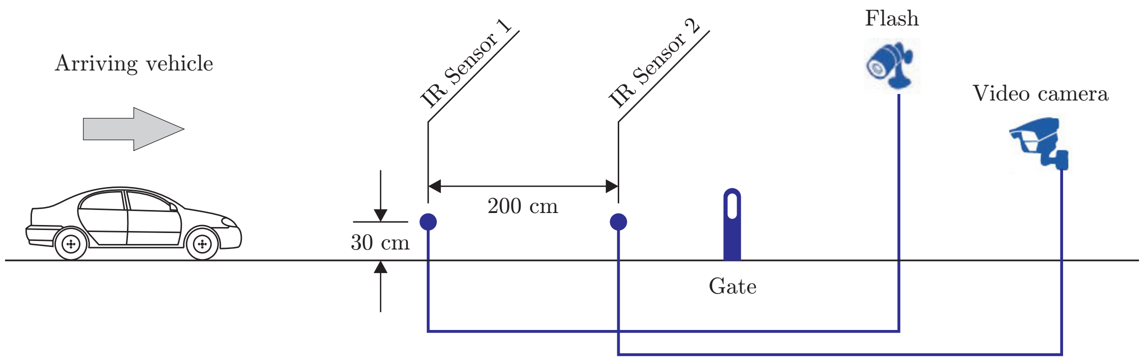


Figure 1: Automatic license plate recognition system configuration

genetic algorithms [6] or different neural networks [3, 4, 5, 12, 25, 24, 18]. Most of the existing systems are also using specific syntax corrections to improve recognition rates.

This article proposes the use of two new feature sets based on zoning of skeleton features for symmetrical and asymmetrical symbols, and syntax based symbol occurrences, as well as a decisional fusion method for good classification rate improvements. In the final section, experimental results based on a real database of images acquired in different lighting conditions, conclusions and further improvements are presented.

2 General processing framework

The proposed system is composed of four main modules: the license plate detection module, the symbol extraction module, the symbol recognition module and the decisional fusion module, which slightly follows the general structure of existing systems.

The license plate recognition system is designed to work as a parking and access control. Therefore, the camera is installed at the vehicle height level (Figure 1). The arriving car passes through an infrared sensor and then stops before the gate. To distinguish between a vehicle and a human, two identical pairs of infrared barriers are mounted at a relative distance comparable to the vehicle mean length. Once the presence detected, the camera takes a front vehicle snapshot and send it to the processing unit to be recognized.

The strategy used for the system conception is to consider as many significant situations that can occur as possible. The main hypothesis is that a license plate is a white background region of certain dimension and aspect ratio. Images are acquired using a visible light camera with a night flash. The *a priori* knowledge is used in order to ease the recognition process and improve the classification result.

3 License plate detection module

By the Romanian standard, the license plate has a white background and an ideal aspect ratio between 0.21 and 0.25, depending on the license type (see Figure 2). This *a priori* knowledge can be used to detect the license plate candidates from an acquired image. Usually, the images are captured at a resolution of 640×480 pixels using a color CCD daylight camera. Because of the noise and the possible artifacts, the aspect ratio is recalculated using tolerances to an interval between 0.17 and 0.26. The algorithm used to detect the license plate candidate is based on the conversion of color images into binary values for increasing speed.

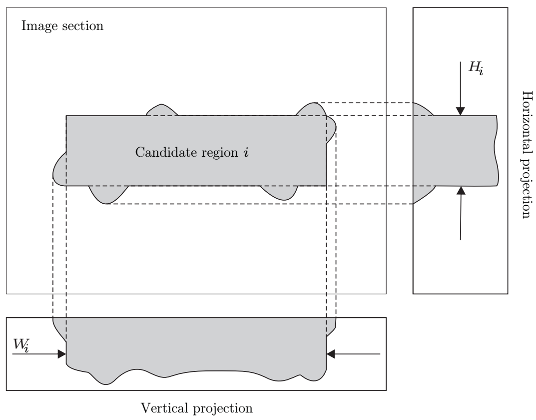


Figure 3: License plate deformations removal. Vertical and horizontal projection are used for thresholding and removing the lateral deformations.

To avoid problems caused by white backgrounds or artifacts, the image in converted to binary using local adaptive thresholding. The average threshold window size has a dimension of 64×12 pixels. Once the binary image obtained, all the connected regions are taken into account and only those who match the *a priori* criteria are considered for further use. The regions

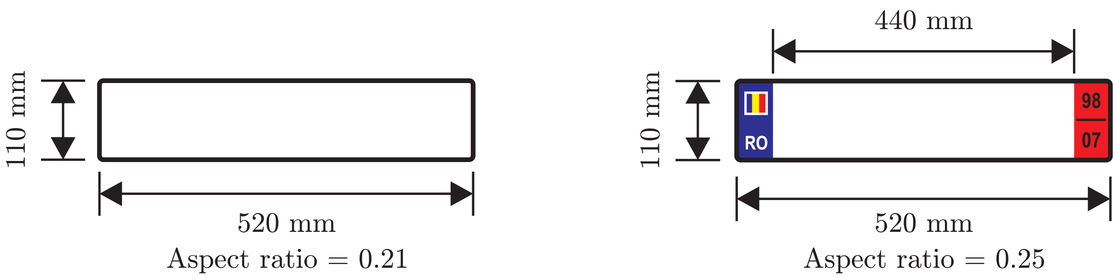


Figure 2: License plate templates. The aspect ratio varies between 0.21 and 0.25 and is used as an *a priori* knowledge for license plate detection

which contain license plate candidates are counting approximately between 7200 and 10200 pixels, considering that the license plate is one third wide of the image width. The assumption is made considering a constant distance between the car and the camera.

The candidate image does not always have a perfect rectangular aspect, so eventual deformations must be taken into account. Therefore, the candidate region is cropped using vertical and horizontal projections in order to obtain dimension median values. The new candidate dimensions are further used as a new selection criterion (Figure 3).

Once the license plate extracted, the image is rotation corrected using the Radon transform computed for angles between 81 and 100 degrees for speed enhancement. Corrections to 90 degrees are required for detected angles exceeding 2 degrees. In order to compensate the license plate frame, 10% increase of license plate dimensions is used when cropping.

Using a database of 309 color images obtained in different weather and lighting conditions, a 93.53% detection rate was obtained.

4 Symbol extraction module

Once the license plate obtained, one must extract the images associated with each of the symbol present in the license plate. Therefore, one must differentiate between a symbol and potentially existing artifacts by using also an *a priori* knowledge.

The extraction algorithm is based on two major parts, an image based segmentation and a syntax based estimation. The image based segmentation searches for symbols in the license plate image using a dimensional criteria. In normal situations, this process is sufficient, but sometimes plates are not perfect and the algorithm cannot extract all the existing symbols. Therefore, after the image based extraction we can observe if the resulting symbols are part of a valid syntax. If not, syntax corrections are required.

The image based segmentation algorithm is using the vertical image projection. The obtained histogram is further thresholded at a 96% value (experimentally chosen as optimal). The symbol start/stop positions are computed from the first derivative of the threshold signal. In order to differentiate between the symbols and artifacts, several criteria are applied. For each symbol/artifact candidate regions, after detecting all the connected regions, only the region with maximum number of pixels is retained. A second criterion is based on the symbol relative dimension to the license plate. Every region with a height greater that 50% of the license plate height is retained. The third criterion is the white filling ratio of more than 15% of the symbol occupied region. Those criteria are obtained experimentally on a database containing more than 300 real license plate images.

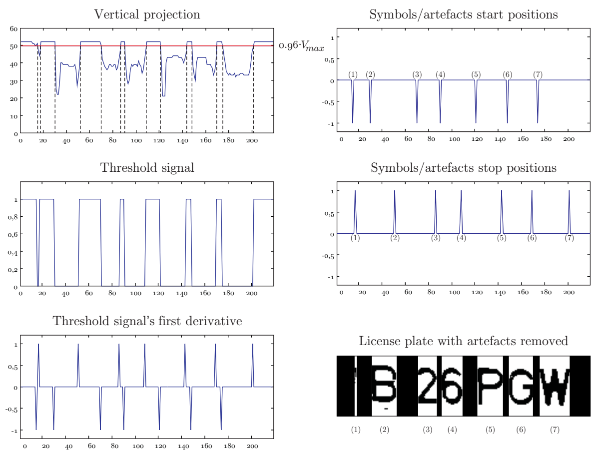


Figure 4: License plate artifact removal. Using the first derivative of the vertical projection threshold signal, start and stop positions of artifacts can be detected V/H projection

At the end of the image based segmentation algorithm, a pre-evaluation of the result is made by counting the resulting number of symbols. Following the Romanian standard, every plate with a number of

Table 1: Character occurrence probabilities based on license plate symbol numbers and positions

Symbols	$P_C^{(n)}(1)$	$P_C^{(n)}(2)$	$P_C^{(n)}(3)$	$P_C^{(n)}(4)$	$P_C^{(n)}(5)$	$P_C^{(n)}(6)$	$P_C^{(n)}(7)$	$P_C^{(n)}(8)$
4	1	0	0	0	-	-	-	-
5	1	1/2	0	0	0	-	-	-
6	1	1/3	0	1/3	1/3	1/3	-	-
7	1	2/3	0	0	1/3	1/3	1/3	-
8	1	1	0	0	0	0	0	0

symbols between 4 and 8 is considered a valid plate. If the number is greater than 8, further corrections are required. Errors are obtained mostly when the license plate is full of dirt or the driver is using illegal radar avoidance measures.

5 Symbol recognition module

The end of the symbol extraction gives a set of images containing each license plate symbol which need to be recognized using OCR algorithms. For the ease of the classification process, an Euler number-based pre-classifier can be used but this can potentially lead to artifact sensitive results. To compensate this deficiency, a morphologic dilatation operator with a structural disc element must be used [17, 16].

5.1 Symbol occurrence probabilities

The classification process proposed in this article is based on several classes of features such as skeleton, template matching and neural network output vectors. A new set of supplementary features which takes into account the probability of a character/digit occurrence at a certain license plate position is proposed. This can be done by using *a priori* knowledge such as the total number of symbols per plate and the valid syntaxes. For every total number of symbols per plate, which can vary between 4 and 8, several syntaxes are valid. Therefore, one can estimate the probability that, a certain position, a character or a digit may occur. The symbol occurrence probabilities are shown in Table 1.

The probability that, at a certain position k, in the license plate, a character appears ($P_C^{(n)}(k)$), depends on the number n of total detected symbols in the plate. The probabilities associated to digits are complementary, $P_D^{(n)}(k) = 1 - P_C^{(n)}(k)$. This information is later used as a feature set in the classification process.

The symbol segmentation score is 95.22%, computed as the ratio between the total number of symbols and the number of detected symbols in the database. Problems occur when the symbol is covered with artifacts or is erased.

5.2 Skeleton-based classification

A discriminator criterion for machine symbols is the use of features extracted from symbol skeletons. The skeletons can be obtained by using an infinite loop thinning morphologic operator on a dilated symbol image. This leads to the possibility of extracting relevant features such as the number of terminal points (TP), T junctions (TJ), or X junctions (XJ) which can be further used in the symbol classification process. Experimentally, a slightly difference between ideal and real symbol feature-based templates for the Arial-style fonts (such as the DIN-1451 Mittelschrift font used in Romanian standards) has been noticed.

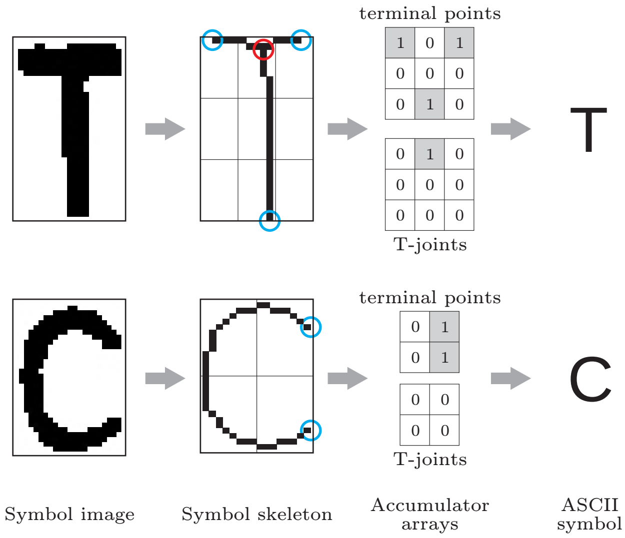


Figure 5: Skeleton feature classification examples. Median feature symbols are using 3×3 accumulator arrays (top), while non-median feature symbols are using 2×2 accumulator arrays (bottom)

To easily differentiate the symbols, a new zoning algorithm based on the presence of median features is proposed. Therefore, the region occupied by a symbol is divided into a 2×2 array for non-median feature symbols and into a 3×3 array for median feature symbols (see Figure 5). A decisional tree is also inferred

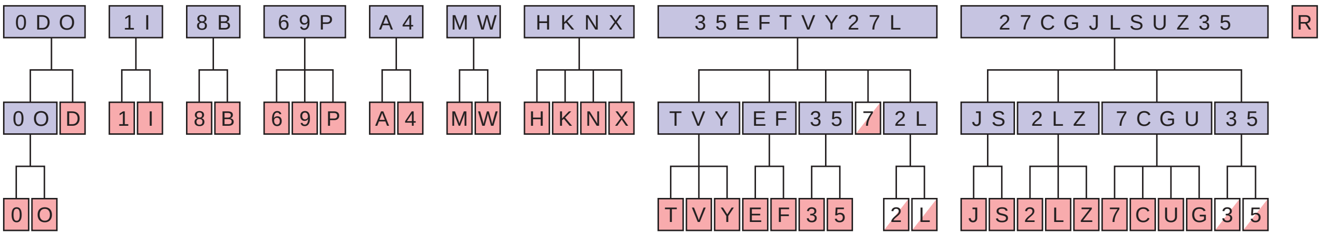


Figure 6: Skeleton feature decision tree. Similar feature configuration symbols are grouped into separate classes.

from the classification process, based on the feature similarities. For example, symbol 2 and L have three terminal points in cells 1, 3 and 4 and can only be differentiated by using a region filling comparator for cell 2. Therefore, there is a great similarity between those two symbols from the skeleton feature aspect.

The current zoning methods compute the mean intensity value in each cell of the array [22]. The use of cells as feature accumulators was preferred instead. Therefore, each cell will contain the exact number of teminal points, T-joints and X-joints for each candidate symbol. The use of a smaller number of cells speeds up the computing time and leads to a smaller number of decisional rules.

The classification vector contains, among the classification result, a probability vector which is based on the decision tree containing the similarities between the recognized symbol and all other possible symbols (see Figure 6). The tree was constructed experimentally, starting from groups of symbols with similar skeleton feature configuration. The probability is computed as the weighted sum of the number of branches passed to reach the corresponding symbol position, computed from the skeleton feature configurations. One can observe that several symbols may appear in different branches of the tree because of different feature configurations associated to those symbols.

As the tree is build with the purpose of grouping similar feature symbols, when a certain symbol is recognized, its classifier output value will have the greatest value. To combine the skeleton feature classifier output with other classifiers, a probability vector is needed. Therefore, all other similar symbols inside the group will also have non-zero output values, proportional to their normalized tree distance from the winner symbol. For example, if the symbol ”X” was recognized, all the classifier output values will be zero, except the values corresponding to symbols ”H”, ”K”, ”N”, and ”X”. The outputs for the first three symbols will be 0.5, and the output associated to symbol ”X” will be 1.

The classification results obtained with this feature set are good and can lead to a good classification rate of 94.43%. This includes also syntax correction in order to differentiate between hard classifiable symbol groups such as ”0”, ”O” and ”D”, ”1” and ”I”, or ”8” and ”B”.

The classification method is sensitive to rules and a more general decision tree based on statistic feature configurations must be completed. Moreover, the presence of artifacts can lead to possible misclassifications.

5.3 Template matching classification

As we are always considering machine symbols, the shape of a single symbol cannot have a great variation between samples. Therefore, the use of a template-based criterion is justified. For every possible symbol in the database a template which is based on real images exists. The choice of using real images instead of synthetic generated ones is based on the hypothesis that all the modifications of a real image cannot be entirely simulated using computer-based font symbols. The template database is shown in Figure 7.

Several comparing rules can be used in order to associate an unknown symbol to a certain class. As all the symbol images are binary, Boolean operations are preferred in order to speed up the algorithm. The chosen rules are AND, XOR and square mean root. The best result is obtained with the XOR rule because of the ability to binary compare all the associated pixels of the symbol image. If, at a certain point, the unknown symbol and a template both contain ones or zeros, a positive local match is considered. The resulting vector contains matching probabilities. The minimum value of this vector corresponds to the best match.

To detect the match between the symbol S and a template T, the matching score $m(S,T)$ is computed:

$$m(S,T) = \sum_{n=1}^{N} \delta_n(S,T) \tag{1}$$

where

$$\delta_n = \begin{cases} 1 & \text{if } S(n) = T(n) \\ 0 & \text{otherwise} \end{cases} , \tag{2}$$

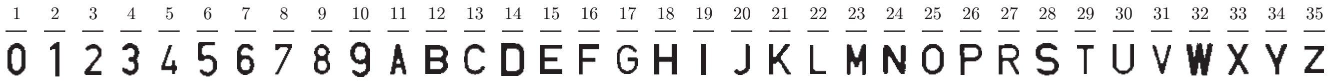


Figure 7: The template matching symbol library for the DIN-1451 Mittleschrift license plate font standard

N is the total number of pixels and n is the current pixel, $n = \overline{1, N}$.

To differentiate varying sizes of the acquired images, a symbol image scaling must be done to the size of 31×19 pixels. Therefore, the classification process is scale invariant. The good classification rate obtained on a large database of more than 2000 unknown symbols is 93.16% without symbol corrections. The classification method is noise robust but sensitive to similar aspect symbols.

5.4 Neural network classification

By far the most used method in OCR, the neural network can accept various sets of inputs, such as direct image values (binary or greyscaled) or image based features [22, 23].

To skip the feature extraction process and, more important, the avoid misclassifications which can occur in this step, we propose the use of binary image eigenvectors. As the binary image can lead to an input vector of significant dimension ($31 \times 19 = 589$ inputs), we use the image PCA projection on the first 40 principal components, therefore greatly reducing the size of the input vector. Before applying the PCA, a symbol scaling is made to obtain scale invariance.

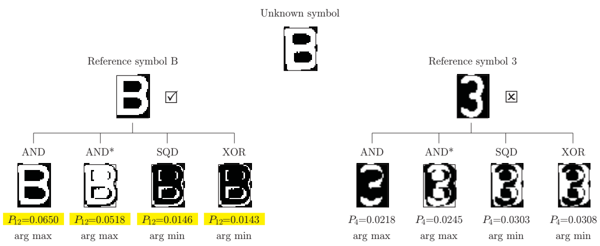


Figure 8: Symbol recognition using template matching. Each symbol is compared using several rules with every reference symbol from the database

The training database has more than 1300 symbols, uniformly distributed on each of the 35 classes (0 to 9 and A to Z, except Q). The MLP optimization leads to a neural network topology with 40 inputs, a hidden layer with 35 neurons, and an output layer with 35 real values. The good classification rate is 99.04%. Several classes are also difficult to be recognized, such as the ”1”s and the ”I”s.

We can notice that, in real conditions, neither of the classification methods can lead to satisfying classification results (see Table 2). Therefore, we propose the use of decisional fusion methods to improve the global classification result.

Table 2: Good classification rates for individual classifiers

w/Syntax	SF	TM	NN
No	-	93.16%	99.04%
Yes	94.34%	-	-

6 Symbol recognition module

Because each of the individual classification algorithms does not offer a satisfying good recognition rate, a decisional fusion is required to reveal the full potential of each classifier [13]. The features used as inputs for the meta-classifier are the output vectors of the skeleton feature, the template matching, and the neural network classifiers. Moreover, the probability occurrences for each symbol are used as inputs in the decisional process. The meta-classification scheme has a parallel *decision in - decision out* configuration.

The meta-classifier fusion rules can be non-trainable (MD - median voting and MJ - majority voting) or trainable (WM - weighted means, BC - Borda count, and EX - expert selection). The median and majority voting assume that all the classifiers are independent and their outputs are converted from real (fuzzy) to binary (hard) values. The trainable decisional rules are based on *a posteriori* good recognition rates for every classifier, computed for each class.

The weighted means rule computes the weighted sum of all the classifiers outputs $C_i(k)$:

$$Q(k) = \left\{ p_p^{(s)}(k) \sum_{i=1}^{3} w_i(k) \cdot C_i(k) \right\} \tag{3}$$

where $p_p^{(s)}(k)$ is the symbol k occurrence probability at location p of the valid syntax formed with all the s license plate symbols.

Table 3: Good classification rates for decisional fusion rules

w/Syntax	MD	MJ	WM	BC	EX
No	98.53%	98.24%	98.75%	84.85%	98.97%
Yes	99.41%	99.34%	99.26%	85.76%	99.34%

The weights $w_i(k)$ represent confidence coefficients based on individual classifier performances obtained with the testing database. The performances are indicated by the *a posteriori* classification probabilities $P_i(\omega_k|x_i)$:

$$w_i(k) = P_i(\omega_k|x_i) \tag{4}$$

where $i = \overline{1,3}$ is the current classifier, x_i is the feature vector for the input symbol, and $k = \overline{1,35}$ is the ω_k class index.

The Borda count rule is based on ranks. Therefore, each classifier outputs is converted from real probability-like values to ranks. The unknown symbol is associated to the symbol which corresponds to the maximum value of the sum of all classifier ranks.

Finally, the expert selection chooses the classifier that has the best performance over a certain class. For every possible symbol, only one classifier is used (the expert) which has the best *a posteriori* classification score.

The decisional fusion results are presented in Table 3. One can observe that the global good classification rate (GCR) is greatly improved, excepting the Borda count rule. This is justified by the fact that the Borda rule cannot incorporate the *a priori* knowledge about the classifier performances (confidence) in its classic form. The good classification rate is 99.41%, computed for a database of more than 1100 uniformly distributed symbols. The result is better than the performances of every individual classifier, therefore the choice of a decisional fusion method is justified.

Because each classifier has relatively good performances, there is no noticeable difference between trainable and non-trainable decisional rules. Nevertheless, a trainable rule must be considered for further use because its results are based on previously studied classifier performances.

7 Conclusions and perspectives

The overall performance of this system is very good. The system is able to detect and recognize license plates for varying conditions such as license plate artifacts or poor image contrast. Several successfully recognized images with critical conditions are presented in Figure 9.

The new skeleton features allow a faster classification generalization using a smaller number of rules, comparing to other similar methods. Moreover, the symbol occurrence probabilities increase by 1% the performances for the separation of similar aspect symbols in case one classifier cannot reveal the differences.

Further algorithm improvements can be considered, especially in the license plate detection module, where more *a priori* information can be included, and where the image based method can be combined with other methods that search for characters instead of white regions.

The probability of good recognition of a license plate from an acquired image is computed as the probability product of every step required in the processing framework. Considering the database of 309 images used in this experiment, we obtain a system recognition rate of $93.53\% \cdot 95.22\% \cdot 99.41\% = 88.53\%$. The rejected images correspond to license plate faults or detection avoidance techniques (erased or modified plates).

The processing time required for a full time computation is less than a second; therefore, a real-time implementation of the system is suitable on a dedicated hardware. The camera requirements are not critical, and no need for special lighting is required, significantly reducing the cost of the hardware.

Acknowledgements: This work was supported by the Defense Ministry Research Project number 211/2006.

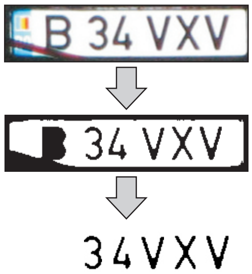
References:

[1] Beatriz Diaz Acosta. *Experiments in Image Segmentation for Automatic US License Plate Recognition*. M. Sc. Thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA, Jun 2004.

[2] Remus Brad. License Plate Recognition System. In *Proceedings of the 3rd International Confer-*

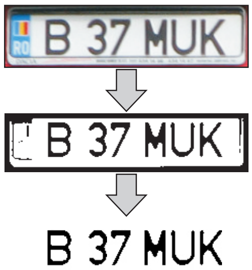
a)

License Plate:	B 34 VXV
Plate Area:	8798 pixeli
Aspect Ratio:	0.19574
Angle:	0 °
No. of Symbols:	5/6
Identified Plate:	34 VXV
Problems:	partial obturation



b)

License Plate:	B 37 MUK
Plate Area:	8753 pixeli
Aspect Ratio:	0.20435
Angle:	0 °
No. of Symbols:	6/6
Identified Plate:	B 37 MUK
Problems:	bordered plate

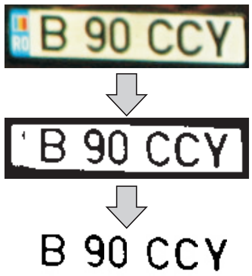


c)

License Plate:	B 38 URO
Plate Area:	6918 pixeli
Aspect Ratio:	0.20225
Angle:	6 °
No. of Symbols:	6/6
Identified Plate:	B 38 URO
Problems:	shearing

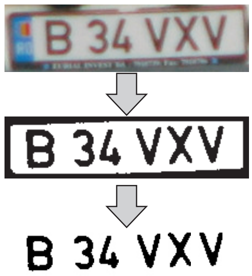
d)

License Plate:	B 90 CCY
Plate Area:	4397 pixeli
Aspect Ratio:	0.2
Angle:	2 °
No. of Symbols:	6/6
Identified Plate:	B 90 CCY
Problems:	multiple objects



e)

License Plate:	B 34 VXV
Plate Area:	6872 pixeli
Aspect Ratio:	0.21026
Angle:	-2 °
No. of Symbols:	6/6
Identified Plate:	B 34 VXV
Problems:	white & blurring



f)

License Plate:	B 73 FSF
Plate Area:	5679 pixeli
Aspect Ratio:	0.20725
Angle:	-8 °
No. of Symbols:	6/6
Identified Plate:	B 73 FSF
Problems:	artefacts & high angle

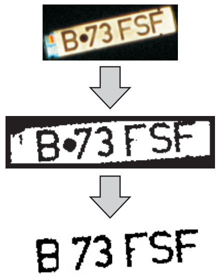


Figure 9: License plate detection results

ence on Information, Communications and Signal Processing, page 2D3.6, Oct 2001.

[3] A. Broumandnia and M. Fathi. Application of pattern recognition for Farsi license plate recognition. *ICGST International Journal on Graphics, Vision and Image Processing*, 5(2):25–31, Jan 2005.

[4] Shyang-Lih Chang, Li-Shien Chen, Yun-Chung Chung, and Sei-Wang Chen. Automatic License Plate Recognition. *IEEE Transactions on Intelligent Transportation Systems*, 5(1):42–53, Mar 2004.

[5] Haim Cohen, Gadi Bergman, and Johanan Erez. Car License Plate Recognition. Project Report, Israel Institute of Technology, 2002.

[6] V. P. de Araujo, R. D. Maia, M. F. S. V. D'Angelo, and G. N. R. D'Angelo. Automatic plate detection using genetic algorithm. In *Proceedings of the 6th WSEAS International Conference on Signal, Speech and Image Processing*, pages 43–48, Sep 2006.

[7] Louka Dlagnekov. Video-based Car Surveillance: License Plate, Make, and Model Recognition. M. Sc. Thesis, University of California, San Diego, 2005.

[8] Tran Duc Duan, Duong Anh Duc, and Tran Le Hong Du. Combining Hough Transform and Contour Algorithm for detecting Vehicles' License-Plates. In *Proceedings of the International Symposium on Intelligent Multimedia, Video and Speech Processing*, pages 747–750, Oct 2004.

[9] Pu Han, Wei Han, Dong-Feng Wang, and Yong-Jie Zhai. Car License Plate Feature Extraction and Recognition based on Multi-Stage Classifier. In *Proceedings of the 2nd International Conference on Machine Learning and Cybernetics*, volume 1, pages 128–132, Nov 2003.

[10] Henrik Hansen, Anders Wang Kristensen, Morten Porsporb Køhler, Allan Weber Mikelsen, Jens Mejdahl Pedersen, and Michael Trangeled. Automatic recognition of license plates. Technical Report, Aalborg University, May 2002.

[11] Ching Tang Hsieh and Yu-Shan Juan. Wavelet Transform Based License Plate Detection for Cluttered Scene. *WSEAS Transactions on Computers*, 4(1):40–44, Jan 2005.

[12] Fatih Kahraman, Binnur Kurt, and Muhittin Gokmen. License Plate Character Segmentation Based on the Gabor Transform and Vector Quantization. In *Proceedings of the 18th International Symposium on Computer and Informational Sciences - ISCIS*, volume 2869, pages 381–388, Nov 2003.

[13] Josef Kittler, Mohamad Hatef, Robert P. W. Duin, and Jiri Matas. On Combining Classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, Mar 1998.

[14] Halina Kwaśnicka and Bartosz Wawrzyniak. License plate localization and recognition in camera pictures. In *Proceedings of Symposium on Methods of Artificial Intelligence - AI-METH*, Nov 2002.

[15] J.-J. Liaw, L.-H. Chang, and L.-H. Fang. A simple car license plate detection method in the real time video system. *WSEAS Transactions on Information Science and Applications*, 4(6):1169–1174, Jun 2007.

[16] Cristian Molder and Munteanu Doru. Decisional Fusion for License Plate Recognition Systems in Surveillance Points. In *Proceedings of the International Conference of Military Technologies, Brno, Czech Republic*, volume 1, pages 639–647, May 2007.

[17] Cristian Molder and Doru Munteanu. Decisional tree for license plate recognition in access point surveillance systems. In *Proceedings of the 4th International Conference New Challenges in the field of Military Sciences 2006, Budapest, Hungary*, volume 1, Nov 2006.

[18] J.A.G. Nijhuis, M.H. Ter Brugge, K.A. Helmholt, J.P.W. Pluim, L. Spaanenburg, R.S. Venema, and M.A. Westenberg. Car License Plate Recognition with Neural Networks and Fuzzy Logic. In *Proceedings of IEEE International Conference on Neural Networks*, volume 5, pages 2232–2236, Nov 1995.

[19] J. R. Parker and P. Federl. An Approach To License Plate Recognition. Technical Report, Department of Computer Science, University of Calgary, Canada, Oct 1996.

[20] Heydar Toossian Shandiz and Seyed Saeid Mirsharifi. Iranian License Plate Recognition Based on Line Scanning and Wiener Filtering of Digital Image. *WSEAS Transactions on Signal Processing*, 2(5):605–611, May 2006.

[21] Vladimir Shapiro, Dimo Dimov, Stefan Bonchev, Veselin Velichkov, and Georgi Gluhchev. Adaptive License Plate Image Extraction. In *Proceedings of the 4th International Conference on Computer Systems and Technologies - CompSysTech*, volume 3, pages 2.1–2.7, Jun 2004.

[22] Oivind Due Trier, Anil K. Jain, and Torfinn Taxt. Feature extraction methods for character recognition - A survey. *Pattern Recognition*, 29(4):641–662, Apr 1996.

[23] Constantin-Iulian Vizitiu, Doru Munteanu, and Cristian Molder. A New Version of Flusser Moment Set for Pattern Feature Extraction. *WSEAS Transactions on Information Science & Applications*, 5(4):396–406, April 2008.

[24] Keem Siah Yap, Yong Haur Tay, Marzuki Khalid, and Tahir Ahmad. Vehicle License Plate Recognition by Fuzzy Artmap Neural Network. In *Proceedings of the World Engineering Congress - WEC*, 1999.

[25] Yungang Zhang and Changshui Zhang. A New Algorithm for Character Segmentation of License Plate. In *Proceedings of the 2003 Intelligent Vehicles Symposium - IV*, pages 106–109, Jun 2003.