Evolutionary Tuning of Compound Image Analysis Systems for Effective License Plate Recognition

Krzysztof Krawiec and Mateusz Nawrocki

Institute of Computing Science, Poznan University of Technology,
Piotrowo 2, 60965 Poznań, Poland

Abstract. This paper describes an evolutionary algorithm applied to tuning of parameters of license plate detection systems. We consider both simple and compound detection systems, where the latter ones consist of multiple simple systems fused by some aggregation operation (weighted sum or ordered weighted average). With the structure of a system given by a human and fixed, we perform an evolutionary search in the space of possible parameter combinations. Several simple and compound structures are considered and verified experimentally on frame collections taken from highly heterogeneous video sequences acquired in varying conditions. The obtained results demonstrate that all considered systems can be effectively tuned using evolutionary algorithm, and that compound systems can outperform the simple ones.

Keywords: Evolutionary computation, evolutionary image analysis, pattern recognition, mixtures of experts, license plate recognition.

1 Detection and Recognition of License Plates

License plate recognition (plate recognition for short) is a common benchmark for pattern recognition and computer vision systems, and one of their most frequent real-world applications. Typical use cases of such systems include authorization of entry for parking lots and gated blocks-of-flats, where the vehicle is close to the camera, the car is not moving (or close to still), and (sometimes) the lighting is partially controlled.

In this study, we focus on a more demanding scenario, where the working conditions of the system are much more unconstrained. This setup is more characteristic to CCTV monitoring in urban areas. Most importantly, the camera used in the experimental phase observes the moving vehicles from a relatively long distance. As a consequence, the observed projected dimensions of the plates are much smaller and the images can be distorted by motion blur and perspective projection. Also, nothing is assumed about the lighting conditions and the quality and state of the plates themselves (the presence of dust, for example). We allow also for the presence of multiple vehicles in the field of view of the camera.

P. Jędrzejowicz et al. (Eds.): ICCCI 2011, Part I, LNCS 6922, pp. 203–212, 2011.

The major contribution of this paper is a compound license plate recognition system that relies on the mixture of experts design pattern and employs evolutionary algorithm to tune the parameters of its particular components. After reviewing selected past work on this topic in section 2, we outline the overall architecture of the system that serves as a framework for this study (Section 3). In Section 4 we describe the elementary components of the considered compound detection systems. Section 5 details on the setup and results of experimental evaluation of the considered recognition systems, in particular on the approach to evolutionary tuning of is parameters. In the final Section 7 we discuss the results and point out the possible further research directions.

2 Related Work

Due to numerous publications, a complete review of all past work done in the area of license plate recognition is beyond the scope of this paper. Former research on this topic engaged various paradigms from computational intelligence, including artificial neural networks, fuzzy logic, and evolutionary computation. For instance, in [14], a fuzzy logic approach has been applied to the problem of number plate recognition. In [7] the author presents the survey of many techniques used in automatic plate recognition systems. Techniques for every stage of recognition process are discussed there: edge detection, image projection, statistical analysis and deskewing mechanism for number plate area detection, horizontal projection for plate segmentation, artificial neural networks for character recognition, and syntactic analysis. The reader interested in these topics is recommended to refer to this review.

There are quite numerous accounts on the use of evolutionary computation for plate recognition. In [12] authors use immune and genetic algorithms to acquire the parameters for the initial step of plate recognition. Thresholds and weights of the neural network are optimized by genetic algorithm in [10]. In [5], genetic algorithm is used to determine the region that covers the license plate.

The plate recognition systems presented in [11,1,3] can serve as another examples of approaches that could be compared side-by-side to the method presented in this paper. This applies also to many commercial solutions. Unfortunately, in most cases the methodology used and the values of performance indicators are the producers' secret, which renders such comparison difficult.

3 Architecture of the Complete System

The overall data flow in the system is unidirectional (bottom-up) and can be divided into four separate stages: motion segmentation (MS), plate detection (PD), character segmentation (CS), and character recognition (CR). The former three stages have been designed based exclusively on domain-specific knowledge and human experience; the last stage involves a powerful support vector machine classifier. In its current form, the method processes each video frame independently (apart from limited use of the previous frame in the MS stage).

The **motion segmentation** (MS) stage is responsible for constraining the system's region of interest (ROI) to those parts of the input frame that potentially represent moving vehicles. As the localization algorithms became efficient, this stage is omitted in the configuration considered in this paper.

The **plate detection** (PD) stage employs sophisticated filtering to determine the potential locations of license plates, called *plate candidates* in following. All the candidates are passed to the CS phase. This is important, not only because of the potential presence of objects that resemble plates, but also because a single frame may actually contain more than one plate, if more than one moving car may turn up in the field of view of the camera.

For each plate candidate returned by the PD stage, the **character segmentation** (CS) stage makes an attempt to segment it and produce a sequence isolated small images representing subsequent characters. Before segmentation, each candidate plate is deskewed. First, the most salient line within the region is found. It is always the top or bottom border of the plate. The slope of that line is then used to rotate the whole plate candidate.

Having deskewed the plate candidate, the CS proceeds to actual character segmentation, which is mostly focused on analyzing the horizontal profile or, in other words, a vertical shadowgraph of the plate candidate. The candidates whose horizontal profile does not resemble that of a typical plate are rejected. If no candidate's profile passes this test, the algorithm assumes that there is no plate in the frame and processing ends at this stage. Thus, this stage may in general produce more than one sequence of character images.

The subsequent **character recognition** stage (CR) processes independently each segmented character image provided by the previous stage. The pixels of character image are fed into support vector classifier (SVM, [9,2]), previously trained on a large collection of human-segmented characters belonging to 36 classes (26 uppercase Latin alphabet letters plus 10 digits). For each of the 36 decision classes, the SVM classifier returns a continuous value that reflects the likelihood of the character belonging to the class. The class with the highest likelihood determines the final decision of the classifier. The recognitions made by SVM for the subsequent characters are concatenated into one character string and returned as the final outcome of the method.

4 Using Mixtures of Experts for License Plate Detection

The concept of mixture of experts is founded on quite intuitive hypothesis that aggregation of multiple different subsystems (experts, predictors, classifiers) can perform better than each subsystem separately. Canonic instances of this paradigm are bagging and boosting algorithms in machine learning [4]. For some variants of this scheme, it has been even proven formally that combining multiple weak yet non-correlated classifiers leads in limit to a perfectly performing compound classifier. This feature has been exploited in many practical applications of machine learning and pattern recognition algorithms [6].

In this paper, we extend our former single-expert approach [8] by blending the mixture of experts paradigm with evolutionary tuning of the overall compound

Table 1. Summary of simple operators (single filters) and compound operators (aggregators) used in the experiment (see text for detailed description)

Operator	Description	Output type	Number of tunable parameters
M	Mask (convolution)	continuous	$4 + 15 \times 15 = 229$
C	Color filter	continuous	4
V	Variance thresholding	binary	5
P	Profile detector	continuous	21
WS()	Weighted Sum	binary	number of arguments+1
OWA()	Ordered Weighted Averaging	binary	number of arguments+1

system. There are two qualitatively different aspects that need to be considered when building a compound image analysis system: its structure (which determines the data flow in the system) and parameters. In this study, the structure of the system is given by the designer and remains fixed, so that the learning task is constrained to parameter optimization.

We assume that there are two types of blocks that our compound recognition systems are built of: *filters* and *aggregators*. Each filter takes a single image (typically the original RGB video frame) as an input and produces a single, same-sized, one-channel image at its output. An aggregator, on the other hand, accepts two or more identically sized single-channel images at input, and produces a single, same-sized, one-channel image at its output. Table 1 summarizes the filters and aggregators used in this paper.

The aggregators considered in the following experiment work pixel-wise, i.e., the intensity of a particular pixel in the output image depends only on intensities of the corresponding pixels in the input images. The first aggregator is Weighted Sum (WS), where the output pixel is simply sum of input pixels multiplied by scalar coefficients. The important characteristics of this filter is that weights are associated one-to-one to particular input images in a fixed way.

In the second aggregator, called Ordered Weighted Averaging (OWA, [13]), there is no such association. For a particular pixel coordinates, OWA first sorts descendingly the intensities of that pixel in the input images. Next, the intensities are multiplied by a vector of weights. The resulting dot product becomes the output value. In this way, OWA can pay different attention to particular input images when considering different pixels.

Both aggregators can have an arbitrary number of inputs, and with each of the inputs a single weight is associated. Additionally, each aggregator compares the obtained aggregated value to a threshold, which is also tunable. Therefore, the total number of parameters that determine the working of an aggregator is equal to the number of its inputs plus one. This is also the number of genes in individual's encoding that are required to encode an aggregator.

We use four filters, three of which, denoted by M, C, and V in Table 1, are quite generic. The first of them, called *Mask* (M) in the following, is a simple convolution of the input image with a mask. Both mask dimensions (width w

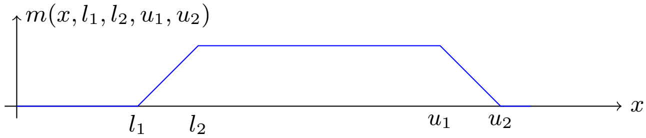

Fig. 1. Fuzzy membership function used in the Color filter

and height h) as well as its weights can be tuned by the search algorithm. When applied to an image, the mask is shifted horizontally and vertically with parameterized steps x_{inc} and y_{inc}. The upper limit on mask size is set to 15×15. Therefore, the complete encoding of this filter in individual's genome requires $2 + 2 + 15 \times 15 = 229$ genes.

The *Color* filter (C) works in the HSV color space and imposes pixel-wise soft thresholding on color saturation and value (intensity):

$$g(x, y) = m(s(x, y), 0, 0, s_1, s_2) + m(v(x, y), v_1, v_2, 255, 255) \quad (1)$$

where g denotes the intensity of the output pixel, $s(x, y)$ and $v(x, y)$ are respectively the saturation and the value of pixel (x, y) in the input image, s_1 and s_2 are the (soft) thresholds for saturation, v_l and v_u are the thresholds for value, and $m(x, l_1, l_2, u_1, u_2)$ is a trapezoidal fuzzy membership function defined as in Fig. 1. To encode the parameters of this filter in individual's genome, one needs four genes, one for each of l_1, l_2, u_1 and u_2.

The *Variance thresholding* filter (V) imposes crisp thresholding on the variance of intensity of the original image, calculated from $w \times h$ window, where $w \in [10, 100]$ and $h \in [5, 30]$ centered on the considered pixel. Similarly to the M filter, when applied to an image, the V filter is shifted horizontally and vertically with parameterized steps x_{inc} and y_{inc}. Thus, its encoding requires five genes (variables): 2 for w and h, 2 for x_{inc} and y_{inc}, and one for threshold.

The fourth filter, called *Profile* (P) in the following, stems from our former research on license plate recognition[8] and is more sophisticated and tailored specifically to the task of license plate detection. This filter's field of view is constrained to a single horizontal row of pixels of a certain length. The pixels are scanned from left to right, and 12 different statistical descriptors $(d_1...d_{12})$, $d_i \in [0, 1]$ are collected from them. They take into account the characteristics of pixels' brightness distribution, like for example the difference between maximal and minimal brightness. Based on these statistics, the output of the filter is defined as $\prod_i d_i$. The detailed construction of this filter is beyond the scope of this paper. The encode of a particular instance of this filter in individual's genome requires 21 genes.

5 The Experiment

The primary objective of the experiment is to verify the usefulness of the mixture-of-experts paradigm for the task license plate detection by comparing selected compound recognition systems with the single filters.

The second aspect to investigate is the usefulness of evolutionary tuning of system parameters. We claim that evolutionary algorithm is an appropriate tool for that purpose, because the number of optimized parameters may be quite large here (particularly for the compound systems), and the performance of a system depends on its parameter setting in a complex, non-linear way. Also, the settings of particular parameters interact with each other (epistasis).

We use collection of 1233 frames of 160 different vehicles (mostly passenger cars) passing in the field of view of the camera, previously used in [8]. Each frame has been manually inspected and the actual (true) license number has been assigned to it. All frames have been acquired using the same stationary FireWire camera working with resolution 1280×960 pixels, located at 15-20 meters from the passing-by cars. The following discussion concerns the frames after the motion segmentation phase, which have typically VGA-comparable resolution. In these frames, the vehicles occupy on average 75% of the frame area, almost all of them in frontal view (only a few frames present rear view). The plates to be recognized have typically dimensions of 150×30 pixels, however, they are often far from being rectangular due to perspective projection and vehicle's tilt and yaw.

It should be emphasized that the prepared dataset has been acquired in realistic conditions and is highly heterogeneous: it comprises various lighting conditions (different time of the day, including backlight as well as plates directly exposed to sunlight), different whether conditions (both sunny and cloudy days), and with license plates subject to dirt and mounted at different heights relative to road level.

The experiment consisted in evolutionary tuning of parameters for different setups of simple and compound systems. We assume that no more than one instance of aggregator is used per setup, which implies 26 possible setups: 4 setups that use single filters and $2 \times (2^4 - 5) = 22$ non-trivial compound setups that involve single aggregator and two, three, or four different filters (an aggregator has to have at least two arguments for the setup to be non-trivial). If one allows using the same filter more than once in a setup, the number of compound setups increases to 26. For brevity, rather than considering all possible compound setups, in this paper we focused on selected compound setups only, summarized in Table 2. Note that different setups imply different numbers of tunable parameters, and therefore different dimensionality of the search space.

Let us also note that among the considered setups we include also one that involves a 'blocked' filter, by which we mean a filter evolved in a separate, earlier evolutionary run, after which its parameters have been fixed to *prevent* the further tuning by the next evolutionary process. This setup is in a sense incremental, as it attempts to build upon a previous, independent learning process.

Given the actual (true) plate location (rectangle) P_{act} and the n plate candidates P_i, $i = 1, \ldots, n$, (also defined as rectangles), an individual's fitness is defined as the relative overlap of both rectangles:

$$\max_i \frac{area(P_{act} \cap P_i)}{area(P_{act} \cup P_i)} \tag{2}$$

Table 2. Summary of compound setups

Setup	Number of tunable parameters (genome length)
WS(P,P)	$21 + 21 + (2 + 1) = 45$
OWA(C,P-blocked)	$4 + 2 + 1 = 7$
OWA(C,P)	$4 + 21 + 2 + 1 = 28$
OWA(P,P)	$21 + 21 + 2 + 1 = 45$
OWA(M,C,V,P)	$229 + 4 + 5 + 21 + 4 + 1 = 264$

where $\cap$ denotes intersection of rectangles. We average this indicator over the entire training set, composed of 100 images drawn randomly from our collection of frames (of which 97 contain any license plates). The same training set has been used in all evolutionary runs.

For each setup, we run generational evolution algorithm on population of 1000 individuals for 100 generations. Individuals are represented as vectors of variables (floating-point numbers) that correspond to parameters used by particular setups. For breeding, both parent solutions are selected independently from the previous population via tournament of size 5 and recombined using one-point crossover. In the resulting offspring, genes (variables) undergo mutation with probability 0.05 per gene. Mutation is Gaussian and multiplicative, affecting more the genes that have large values. If the mutated value violates the $[0, 1]$ bounds, it is clamped.

In Table 3 we report the fitness of the best-of-run individual for each setup, and the performance on a disjoint test set of 100 images (of which 98 contain plates). Though the performance of compound detectors is on average better, none of them clearly outperforms the P filter.

Figure 2 presents exemplary results of the plate detection process carried out by the best-of-run individual of the WS(P,C) setup for one of the frames. Light-colored regions indicate the locations where the filters' belief in plate presence is higher. A closer inspection of images produced by constituent filters reveals that their fusion is synergetic, i.e., they complement each other in a way which leads to better performance of the overall compound detection system.

Table 3. Fitness of the best-of-run individuals for simple (left) and compound (right) setups, for the training set and the testing set

Setup	Training set	Testing set
M	0.5439	0.4128
C	0.6263	0.5335
V	0.3300	0.3162
P	0.7536	0.6416

Setup	Training set	Testing set
WS(P,P)	0.6805	0.5512
OWA(C,P-blocked)	0.7595	0.6437
OWA(C,P)	0.6554	0.5249
OWA(P,P)	0.7402	0.6148
OWA(M,C,V,P)	0.6276	0.5212

Fig. 2. The process of recognition implemented by the best-of-run individual of WS(P,C) setup: the output of the C filter (top left), the output of the P filter (top right), and the aggregated confidence image produced by the WS aggregator (bottom left), and the final detection outcome (bottom right) with the detected localizations

6 Driving Evolution by Recognition Accuracy

As the next step, we embedded the best evolved plate localization subsystems in the complete plate recognition system and tested its recognition rate. The performance turned out to be disappointing, with the number of erroneous plate readings far too high for practical use. This suggests that the evolutionary algorithm found a way to maximize the overlap between the actual plate location and the found plate candidates, but for some reasons this capability has not been reflected in the actual plate recognition rate.

This observation inclined us to redefine the learning task and guide the simulated evolution using the actual recognition rate. To this aim, we designed a new fitness function based directly on the detected character sequences:

$$\frac{1}{n}\sum_{i=1}^{n}\frac{d_{max} - \min(d_{max}, d(s, s_i))}{d_{max}} \tag{3}$$

where n is the number of plates in the training set, s is the character string representing the plate number as read by the recognition system, s_i is the actual plate number present in the ith training frame, and d is the Levenshtein distance metric $d_{max} = 5$, so the maximal distance that positively contributes to fitness is 4 (most plate numbers used here had 7 characters). If $d(s, s_i) \geq 5$, an individual scores 0 for the frame.

Table 4. Distribution of Levenshtein distance d

Levenshtein distance d	0	1	2	3	4	5	$d \geq 5$	Total
Training set images	60	25	5	0	0	4	3	97
Test set images	38	23	6	0	2	2	27	98
Test set images (corrected)	57	8	2	1	1	2	27	98

Fig. 3. Comparison of the 0 (zero, left) and O (right) characters

The experiment was conducted for three configurations that appeared to work best in the previous test: the simple P and C configurations, and the combined WS(C, P) configuration. Apart from the new definition of fitness function, we used the same parameter setting as in Section 5. Of the three considered configurations, WS(C,P) turned out to produce the best final training set fitness of 0.85, with over 61.9% of plates' images perfectly detected and recognized. This result confirms our working hypothesis that synergy between two or more localizators is possible.

Table 4 presents the distributions of Levenshtein distance for the training and test set. For the test set, the share of perfect recognitions drops to 38.8%. Clearly, most of the errors boil down to single-character mistakes, implying that the plate detection phase works well, as the remaining characters are correctly recognized. This suggests that, given better character recognizer, further improvements are likely. Thus, we analyzed the statistics of errors committed by the recognizer and found out that the majority of errors consist in confusing the 0 (zero) and O characters. This should not come as a surprise, given how similar these characters are in Polish license plates, as demonstrated in Fig. 3. These characters differ only in aspect ratio, which can be easily distorted by the projection of plate image onto camera sensor. Therefore, perfect discrimination of these decision classes is impossible without help of some additional information, like syntactic rules (e.g., 'license plate cannot start with a numeral'). If we accept this fact and treat the 0 and O characters exchangeably by merging them into one decision class, the structure of errors changes to the one shown in the bottom row of Table 4, meaning 58.2% of perfect recognitions. Though this result may still seem far form perfect, one has to take into account that the same vehicle is typically visible in a few consecutive frames, so the recognition rate can be potentially boosted by aggregating recognitions obtained for multiple frames. Given high performance of our approach (about 40 frames per second), this could be done at no extra cost.

7 Summary

In this study, an evolutionary algorithm proved useful for optimization of parameters of simple and compound licence plate recognition systems. The attained recognition rate turned out to be much higher than that obtained by manual tuning, and the compound recognition systems that aggregate the outputs of multiple heterogeneous or homogeneous subsystems improve the detection rate, compared to single filters.

Acknowledgments. Work supported by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n°218086.

References

1. Abdullah, S., Khalid, M., Yusof, R., Omar, K.: License plate recognition using multi-cluster and multilayer neural networks 1, 1818–1823 (2006)
2. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
3. For, W.K., Leman, K., Eng, H.L., Chew, B.F., Wan, K.W.: A multi-camera collaboration framework for real-time vehicle detection and license plate recognition on highways, pp. 192–97 (June 2008)
4. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. In: Vitányi, P.M.B. (ed.) EuroCOLT 1995. LNCS, vol. 904, pp. 23–37. Springer, Heidelberg (1995)
5. Ji-yin, Z., Rui-rui, Z., Min, L., Yin, L.: License plate recognition based on genetic algorithm. In: 2008 International Conference on Computer Science and Software Engineering, vol. 1, pp. 965–968 (2008)
6. Krawiec, K., Kukawka, B., Maciejewski, T.: Evolving cascades of voting feature detectors for vehicle detection in satellite imagery. In: IEEE Congress on Evolutionary Computation, July 18-23, pp. 2392–2399. IEEE Press, Barcelona (2010)
7. Martinsky, O.: Algorithmic and mathematical principles of anpr systems (2007)
8. Nawrocki, M., Krawiec, K.: A robuts method for real-time detection and recognition of licence pla tes. In: IEEE International Conference on Multimedia Communications, Services, and Security (MCSS 2010), pp. 171–175 (2010)
9. Platt, J.: Fast training of support vector machines using sequential minimal optimization. In: Schölkopf, B., Burges, C., Smola, A. (eds.) Advances in Kernel Methods – Support Vector Learning. MIT Press, Cambridge (1998)
10. Sun, G., Zhang, C., Zou, W., Yu, G.: A new recognition method of vehicle license plate based on genetic neural network. In: 2010 the 5th IEEE Conference on Industrial Electronics and Applications (ICIEA), pp. 1662–1666 (2010)
11. Tseng, P.-C., Shiung, J.-K., Huang, C.-T., Guo, S.-M., Hwang, W.-S.: Adaptive car plate recognition in qos-aware security network. In: SSIRI 2008: Proceedings of the 2008 Second International Conference on Secure System Integration and Reliability Improvement, pp. 120–127. IEEE Computer Society, Washington, DC, USA (2008)
12. Wang, F., Zhang, D., Man, L.: Comparison of immune and genetic algorithms for parameter optimization of plate color recognition. In: 2010 IEEE International Conference on Progress in Informatics and Computing (PIC), vol. 1, pp. 94–98 (2010)
13. Yager, R.R.: On ordered weighted averaging aggregation operators in multicriteria decisionmaking. IEEE Transactions on Systems, Man and Cybernetics 18(1) (1988)
14. Zimic, N., Ficzko, J., Mraz, M., Virant, J.: The fuzzy logic approach to the car number plate locating problem. In: IASTED International Conference on Intelligent Information Systems, p. 227 (1997)