New Generation Computing, 33(2015)389-408
Ohmsha, Ltd. and Springer



# An Improved Vehicle Logo Recognition Using a Classifier Ensemble Based on Pattern Tensor Representation and Decomposition

Bogusław CYGANEK,[1] Michał WOŹNIAK[2]
*AGH University of Science and Technology,*
*Al. Mickiewicza 30, 30-059 Kraków, POLAND*[1]
cyganek@agh.edu.pl
*Wrocław University of Technology,*
*Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, POLAND*[2]
Michal.Wozniak@pwr.wroc.pl

Received 15 December 2014
Revised manuscript received 1 August 2015

***Abstract*** The paper presents a vehicle logo recognition system based on novel combination of tensor based feature extraction and ensemble of tensor subspace classifiers. Each originally two-dimensional vehicle logotype is transformed to a three-dimensional feature tensor applying the extended structural tensor method. All such exemplary logo-tensors which correspond to a single class are stacked to form a 4D logo-class-tensor. Decomposing each 4D logo-class-tensor into the orthogonal tensor subspace allows classification of unknown logotypes. The proposed system allows reliable vehicle logo recognition in real conditions as shown by experiments.

**Keywords:** Vehicle Logo Recognition, Tensor Subspace Classification, Extended Structural Tensor, Higher Order Singular Value Decomposition.

## §1 Introduction

Automatic vehicle identification is an important topic in traffic surveillance and security monitoring. Of special interest is reliable recognition of the license plate as well as type of a vehicle. Thus, recognition of a vehicle brand and model has found broad interest.[18,19] In this respect, many of the reported systems focus on the appearance based approaches which require high computa-

tional load. However, recognition of vehicle logos is an alternative way of determining vehicle type. The problem of reliable logotype recognition was addressed in many scientific works and their overview is provided in the next section.

In our work, we propose a novel vehicle logo recognition method based on multilinear tensor processing, as well as employing ensemble of classifiers. A special attention of this paper is paid to the vehicle logo classification system, which extends the classification method proposed in our previous work.[6] The main assumptions, as well as modifications and improvements in respect to the previous version of the system are presented as follows. In the proposed system, the vehicle logo areas are fed to the ensemble of classifiers in which all the individual classifiers operate within the tensor orthogonal subspaces, as will be described. These are obtained from the *Prototype Pattern Tensors* (PPT) created from the *Extended Structural Tensor* (EST) of the geometrically deformed training exemplar prototypes. Each such PPT is then decomposed with the *Higher-Order Singular Value Decomposition* (HOSVD) method. Transformation of the EST into PPT form, which is then used to build the individual classifiers, is the main contribution of this paper. The proposed method shows high accuracy and reliable operation over variety of patterns and viewpoints.

The rest of the paper is organized as follows. Section 2 outlines related works in the area of the vehicle logo recognition. Section 3 contains description of the method. Specifically, Section 3.1 introduces the general structure of the proposed system and provides description of the main building blocks. Section 3.2 presents feature extraction based on the computation of the Extended Structural Tensor. Section 3.3 deals with construction of the base classifiers. Section 4 presents experimental results. The last part concludes the paper.

## §2 Overview of Related Works on Vehicle Logo Recognition

Car logo recognition falls into one of the interesting problems of computer vision and found also interest in the pattern recognition community. This may be viewed as an auxiliary problem to the license plates recognition, as well as to determination of a car type, which finds application to road traffic and parking surveillance. In this section, we briefly outline recent works devoted only to the car logo recognition.

Petrovic and Cootes[18] propose a system in which edge gradient with a refined matching algorithm is used. However, the results are sensitive to viewpoint change as well as plane rotation. On the other hand, Dai et al. propose the Tchebichef statistical moments for the purpose of vehicle-logo recognition.[7] However, these methods are very sensitive to noise and geometrical distortions.

In the work by Ki and Baik, vehicle classification with neural network is proposed.[10] The other system, proposed by Sulehria and Zhang, is based on the mathematical morphology methods.[23] However, both of them have many limitations and can operate with only few brands of vehicle logos.

Other published methods rely on object recognition with sparse feature detection. Psyllos *et al.* propose a system which is based on the SIFT descriptors.[19] However, its computation load is high, whereas the method is not free

from illumination and viewpoint variations, as well as it depends on many parameters.

Based on the literature survey, we propose a vehicle logo recognition system which operates in tensor domain. Despite high dimensionality of the input data, responses of the system allow its highly accurate and real-time operations, as will be discussed.

## §3 Tensor-Based Vehicle Logo Recognition Method

### 3.1 General Structure of the System

The general scheme of the proposed system is presented in Fig. 1, which follows the chain of processing modules. The front end consists of two detectors. The first one detects the license places based on their characteristic textures. For this purpose, the structural tensor is used, as described in [6]. Based on the position of a license plate, potential logo candidate areas are chosen. These are also selected using the criteria of highly textured area of specific size, related to the license plate size. We noticed that for majority of standard car models and arrangements, this strategy succeeds. However, there are models in which logo is not in the region estimated by this simple model or there are other car elements, such as the grille, lights, etc., which can be confused for logo region. All this means that the detection module provides many potential logo positions, which frequently are incorrect. Thus, there is another important processing step which does precise logo classification. Contrary to the preceding detection module, it is trained with the real prototype exemplars, which leads to more accurate classification, as will be discussed.

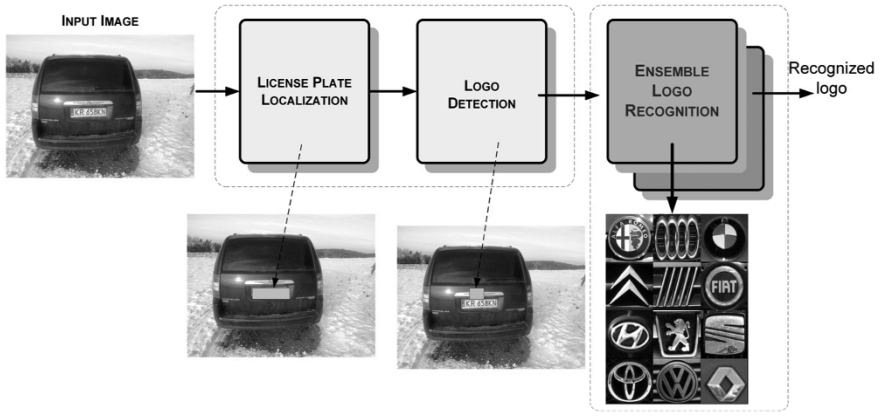


**Fig. 1** Architecture of the System for Vehicle Logo Recognition. The processing flow includes license plate detection, vehicle logo areas detection, and finally logo classification.

However, despite the fact that vehicle logotypes are quite simple for human recognition, their classification poses a challenge for computer vision methods. This is mostly due to their great variations in shape, scale, and viewpoint,

as well as other factors such as motion blur, small resolution, noise, etc. In the previous version of our system, we used a tensor based classifier, which was trained with single exemplars of each of the logotypes. The tensor approach inherently accounts for multidimensional nature of the classified patterns, whereas majority of the "classical" approaches operates in the 1D vector space. In this case, the patterns are two-dimensional monochrome images and their classification was based exclusively on the intensity signal.

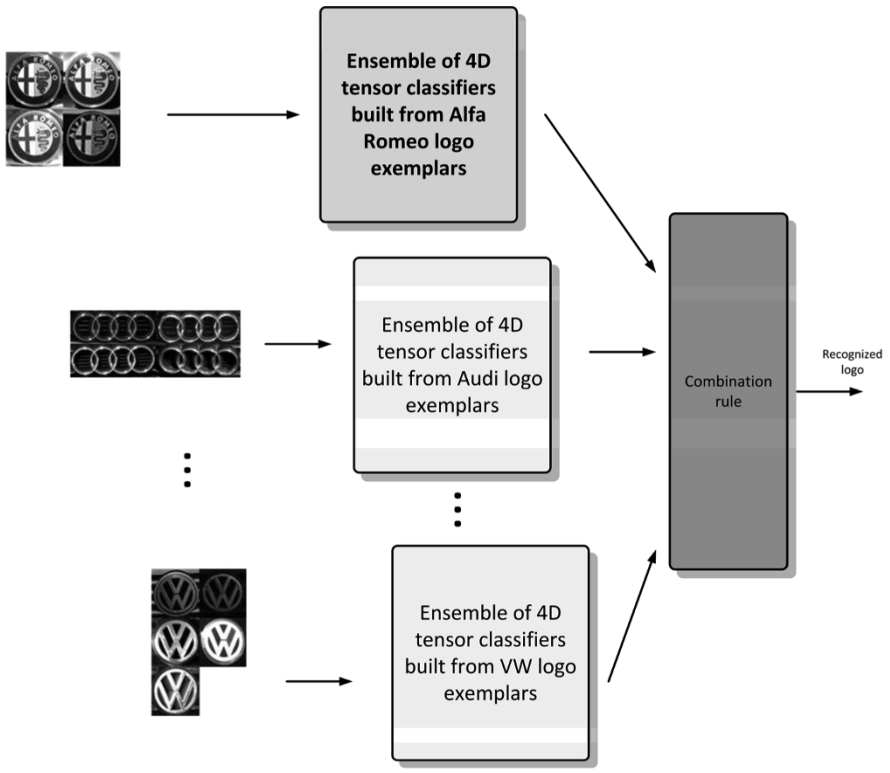


**Fig. 2** Structure of the ensemble of ensemble modules presented in Fig. 1. Each set of prototype patterns for a given logotype is processed by an ensemble of 4D tensor classifiers built from all prototypes but only one logotype.

The classification module of the proposed systems was greatly extended. Its overall view is depicted in Fig. 2, while details of the single ensemble module for one logotype are shown in Fig. 3.

Firstly, we use many exemplar patterns for each logotype (Fig. 3). These are some logo cropped from the well visible images. Secondly, to account for geometrical deformations present in real images, each of the exemplar patterns is geometrically deformed. In this version of the system, these are only new rotated patterns in the range $+/-18^\circ$ with step of $2^\circ$. Thus, a greatly enhanced training set of 2D image patterns is obtained. However, to make the method

more resistant to luminance variations and noise, we use more information than only the intensity signal. For this purpose, during the third step, each such generated image pattern is processed by the already mentioned EST. In result, each pixel of the 2D logo image is endowed with a 3x3 symmetrical matrix (a tensor). Thus, from a single 2D image a 4D tensor is obtained. We simplify this representation since the EST are symmetrical. Thus, finally we deal with 3D tensors. Next, all these 3D representations obtained from different training prototypes but of a single logo type, for instance Toyota, are stacked to form a final 4D tensor of deformed prototypes. Finally, this tensor is decomposed with the HOSVD algorithm to form orthogonal sub-space used for subsequent pattern classification. What is interesting, contrary to the classical orthogonal vector space,[24] the base "vectors" are not 1D vectors - these are multidimensional tensors (3D in our case). Combination of the EST preprocessing and application of HOSVD to build the orthogonal sub-spaces with 3D tensor bases, constitutes the main novelty of this work. Interestingly enough, although the described process seems to be memory and time demanding, training can be done in few minutes, while system response achieves more than 30 logo classifications per second in software implementation.

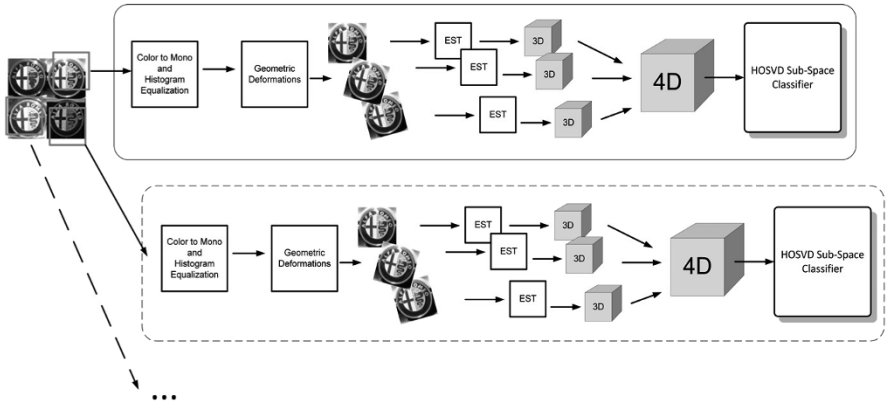


**Fig. 3** Architecture of one the ensemble of 4D tensor classifiers presented in Fig. 2 built from all prototypes of only one logotype. Each prototype serves to generate a number of its geometrically deformed versions. Each such prototype is processed by the extended structural tensor which produces a 3D tensor. All 3D tensor prototypes are then composed into a 4D tensor which after the HOSVD decomposition creates a subspace with orthogonal tensor bases.

Let us notice, that the classifier operates as an ensemble of classifiers, as shown in Fig. 2. Such ensembles find broad interest in scientific society due to their superior accuracy as compared with systems based on a single classifier, as well as their fast operation and flexibility in many applications.[12,13,29] Also in the presented system, operation and response of the ensemble are accurate, as will be shown. Also, the whole structure can be easily updated depending on a number of available logo prototypes. Last but not least, the operation of the

ensemble can be easily parallelized, as well as implemented in GPU, to speed up its response.

The role of the combination rule is to compute a single answer of the whole ensemble, given partial responses of each of the individuals. For this purpose, there are many algorithms proposed in the literature, which depend on a type of a response - discrete or continuous - given by the individual classifiers.[29] It is worth noticing that our system is a kind of an ensemble of classifier ensembles. Let's call them lower and upper ensemble, respectively. The HOSVD algorithm, used by the each individual classifier, responds with a continuous value in the range from 0 to 1, as will be described. On both levels of combination rule, the maximum value (*Winner Takes All*) strategy is used. It is fast in computation and shows good results in practice. However, an additional mechanism of measuring ratio of the first-best to the second-best is also implemented. We noticed that if their ration is substantially high (in our experiments $> 3\%$), then the answer is more reliable. It comes at a cost of omitting some "unsure" logos. In other words, this rises accuracy at a cost of un-classified (left) patterns, however.

### 3.2 Feature Extraction for Vehicle Logo Classification

Intensity signal provides important information on logo contents. However, classification based on intensity values sometimes fails due to signal variations and noise. Therefore, more discriminative features can be obtained computing signal derivatives which are then averaged in local areas in the image. In this section, we outline this procedure, describing the formation and computation of the Extended Structural Tensor.

Let us denote a 2D image with $I$. The structural tensor $\mathbf{T}$ can be computed at a point $\mathbf{p}_0$ and considering its compact nearest neighborhood $R(\mathbf{p}_0)$ in $I$, in accordance with the following Equation (1)

$$\mathbf{T}(\mathbf{p}_0) = G_{R(\mathbf{p}_0)} \left( \mathbf{D}\mathbf{D}^T \right). \tag{1}$$

In the above, $G_{R(\mathbf{p}_0)}(\cdot)$ denotes an averaging operator in the region $R$, centered at a point $\mathbf{p}_0$. On the other hand, the vector $\mathbf{D}$ denotes an image gradient which is computed at each point $\mathbf{p} \in R(\mathbf{p}_0)$. In the simple approaches, $G_{R(\mathbf{p}_0)}(\cdot)$ is realized as a discrete binomial or Gaussian filter.[3,8] Nevertheless, more precise computations require $G_{R(\mathbf{p}_0)}(\cdot)$ to be an anisotropic filter, as will be discussed later in this paper. Computation of the gradient $\mathbf{D}$ at a point $\mathbf{p}$ of $I$, is done as follows

$$\mathbf{D}(\mathbf{p}) = \left[ \frac{\hat{\partial}}{\partial x} I(\mathbf{p}) \quad \frac{\hat{\partial}}{\partial y} I(\mathbf{p}) \right]^T = \left[ I_x(\mathbf{p}) \quad I_y(\mathbf{p}) \right]^T, \tag{2}$$

where $I_x(\mathbf{p})$ and $I_y(\mathbf{p})$ are discrete spatial derivatives of $I$ at a point $\mathbf{p}$, in the $x$ and $y$ directions, respectively. From (1), it is easy to observe that $\mathbf{T}(\mathbf{p}_0)$ is a symmetric positive 2D matrix. Its elements describe averaged values of the gradient components in a neighborhood around the point $\mathbf{p}_0$. Thanks to this, the structural tensor $\mathbf{T}$ contains information on signal changes not only at a single point $\mathbf{p}_0$, but also in its closest neighborhood. The structural tensor $\mathbf{T}$ can be

also seen as a measure of a concordance of orientations of local gradients in $R$.[8] Let us also notice, that since $\mathbf{T}$ is computed at each point $\mathbf{p}$ of $I$, then each $\mathbf{T}(\mathbf{p})$ conveys information on overlapping regions around each point $\mathbf{p}$. Thanks to this property, $\mathbf{T}$ contains information on image texture and local curvature. However, to provide even more discriminative information on image content, the structural tensor can be further augmented with color information, as will be described. Now, inserting (2) into (1), $\mathbf{T}$ can be expressed as follows

$$\mathbf{T} = G_R \left( \begin{bmatrix} I_x I_x & I_x I_y \\ I_y I_x & I_y I_y \end{bmatrix} \right) = \begin{bmatrix} T_{xx} & T_{xy} \\ T_{yx} & T_{yy} \end{bmatrix}. \tag{3}$$

In (3), to keep it simple, the point $\mathbf{p}$ was omitted. As alluded to previously, the averaging operation, performed by the filter $G_R$, is over a set of points in the neighborhood $R$.

As already discussed, the structural tensor $\mathbf{T}$ in (3) provides rich information on signal changes in local patches in a 2D image. However, in many tasks, it is useful to use color information *and* image changes together. An extension, which unifies components of the structural tensor and intensity/color values, was proposed by Luis-García et al.[25] Their approach was used for image segmentation based on optimization of the energy functional. For this purpose, the two-dimensional gradient vector $\mathbf{D}$ in (2) is extended to form a three-dimensional vector $\mathbf{E}$. It is defined as follows

$$\mathbf{E}^T(\mathbf{P}) = \begin{bmatrix} \mathbf{D}^T(\mathbf{p}) & I(\mathbf{p}) \end{bmatrix}^T = \begin{bmatrix} I_x & I_y & I \end{bmatrix}^T. \tag{4}$$

Now, inserting (4) into (1) and substituting $\mathbf{E}$ for $\mathbf{D}$, the nonlinear *extended structural tensor* (EST) is obtained, as follows

$$\mathbf{T}_E = G_R \left( \mathbf{E}\mathbf{E}^T \right) = G_R \left( \begin{bmatrix} I_x^2 & I_x I_x & I_x I \\ I_y I_x & I_y^2 & I_y I \\ I_x I & I_y I & I^2 \end{bmatrix} \right). \tag{5}$$

Thus, the EST contains averaged components of the products of gradient, as well as mixed products of the gradient components and intensity. These form well discriminative features for object detection and classification, as will be discussed.

Figure 4 shows process of computation of the $\mathbf{T}_E$ at one pixel location. However, $\mathbf{T}_E$ expresses information contained in that point and its closest neighborhood. Thanks to this, $\mathbf{T}_E$ conveys more discriminative information than bare intensity signal.

Figure 5 depicts components of the Extended Structural Tensor computed from the monochrome version of one logo.

When computing the structural tensor, as well as its extended version EST, very important thing is choice of the averaging operator $G_R$. In the simple implementation, a fast binomial or Gaussian filters can be used with proper scale. However, both these are isotropic filters which do not follow local topology of the filtered regions. To remedy this, a nonlinear anisotropic approach was proposed in [2]. In the presented system we also follow this way and use a

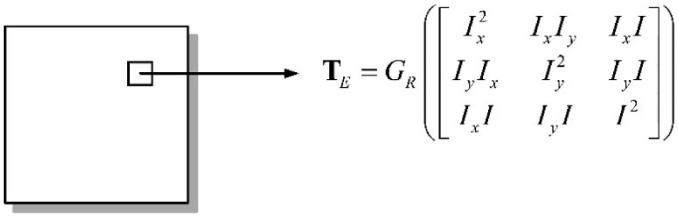


**Fig. 4** Illustration of computation of the Extended Structural Tensor in compact local regions around each pixel position. The operation converts a 2D image into a 4D tensor since each pixel becomes a 2D matrix by itself.

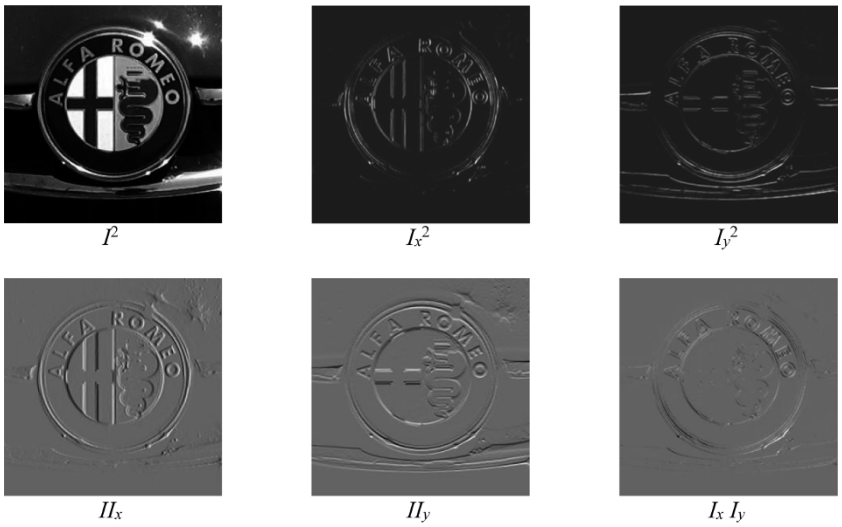


**Fig. 5** Components of the Extended Structural Tensor computed from the monochrome version of the logo.

nonlinear anisotropic filter. In the original formulation of the anisotropic filter, proposed by Perona and Malik, to filter an image $I$, the nonlinear heat equation is employed, as follows[27)]

$$\partial_t I(x,y,t) = div\left(c\left(\|\nabla I(x,y,t)\|\right) \cdot \nabla I(x,y,t)\right). \tag{6}$$

In the above, $c$ is a nonlinear control function with its argument being a module of the gradient of the filtered signal. The role of the function $c$ is to stop smoothing in directions of large gradients. Thanks to this, the smearing effect at the edge boundaries is avoided. There are many variants of the function $c$ proposed in literature. In our system, the following control function is used[28)]

$$c(x) = \begin{cases} \frac{1}{2}\left(1 - x^2/\sigma^2\right)^2, & |x| \le \sigma \\ 0, & |x| > \sigma \end{cases}. \tag{7}$$

It is called the Tukey bi-weight function and shows good results in many image filtering tasks which require untouched strong signals to preserve local features. The scale parameter $\sigma$ in (7) is chosen as a so called robust scale. It is computed in accordance with the following formula

$$\sigma = K \cdot MAD(\nabla I) = K \cdot med\left(\|\nabla I - med\left(\|\nabla I\|\right)\|\right), \tag{8}$$

where $MAD$ denotes for the median absolute deviation, and $med$ is a median. $MAD$ conveys a value of statistical dispersion. It tells how large a random variable can be before considering it as an outlier. Using median is more robust to outliers detection than, for example, relying on the mean and the standard variation. In our case, that is in the anisotropic filtering, the robust scale measures a deviation of the intensity gradient. The constant $K$ denotes a scale factor which assures that $MAD$ is a consistent estimator; Also, its value depends on the distribution of a random variable. It can be shown that for this purpose $K = 1/Q(0.75)$, where $Q(0.75)$ stands for the 0.75 quantile of the underlying distribution.[9] In the commonly assumed case of the zero-mean normal distribution with unit variance, this leads to $K = 1/Q_N(0.75) \approx 1.4826$, where $Q_N$ denotes the quantile function for the normal distribution.[16]

However, a downside of the anisotropic image filtering is the iterative computation of (6). This results in longer computation and is not easy to be parallelized. Nevertheless, in our serial software framework, 10-20 iterations were always sufficient.

Last but not least, for proper operation, the EST needs to be normalized before being put to the classifiers. The normalization is done by computation of the square root of each of the EST matrices. Since these are always positive definite, this operation is possible for all EST. Details, as well as code examples of these procedures, are presented in [4].

### 3.3 Construction of the Tensor Classifiers

The task of vehicle logo classification is to decide if an observed part of an image is a valid logo and, if so, what class it represents. There are many classification methods whose choice is determined by various factors. From these, the accuracy and respond time are the most important once, however, the type and dimensionality of the input data play an important role as well. Taking all these under consideration, and based on our previous experience, the multilinear tensor classifier was chosen. The main idea consists of gathering two dimensional samples of the prototype logos which are then stacked to form a 3D tensor. Then, this prototype tensor is decomposed to obtain the subspace suitable for classification. In consequence, an unknown pattern is classified by checking a distance of its projection onto tensor spanned subspace. To build the aforementioned subspace, the Higher-Order Singular Value Decomposition (HOSVD) can be used.[1, 11, 14] In effect, the orthogonal tensor bases are obtained, which are then used for pattern recognition in a similar way to the standard PCA based classifiers.[24] Let us recall, that the HOSVD allows decomposition of any $P$-dimensional tensor $\boldsymbol{\mathcal{T}} \in \mathfrak{R}^{N_1 \times N_2 \times \ldots N_m \times \ldots N_n \times \ldots N_P}$ to the following

representation[14,15]

$$\boldsymbol{\mathcal{T}} = \boldsymbol{\mathcal{Z}} \times_1 \mathbf{S}_1 \times_2 \mathbf{S}_2 \ldots \times_P \mathbf{S}_P. \tag{9}$$

The matrices $\mathbf{S}_k$ of size $N_k \times N_k$ are unitary matrices, and $\boldsymbol{\mathcal{T}} \in \mathfrak{R}^{N_1 \times N_2 \times \ldots N_m \times \ldots N_n \times \ldots N_P}$ is a core tensor. For pattern recognition task, one of the very useful properties of the core tensor $\mathcal{Z}$ is its all-orthogonality. That is, for all its two subtensors $\boldsymbol{\mathcal{Z}}_{n_k=a}$ and $\boldsymbol{\mathcal{Z}}_{n_k=b}$, and for all possible values of $k$ for which $a \neq b$, the following holds

$$\boldsymbol{\mathcal{Z}}_{n_k=a} \cdot \boldsymbol{\mathcal{Z}}_{n_k=b} = 0 \tag{10}$$

An algorithm for computation of the HOSVD is based on successive computations of the SVD decomposition of the matrices actually being the flattened versions of the tensor $\boldsymbol{\mathcal{T}}$.[4] The algorithm requires a number of the SVD computations which are equal to the valence of the tensor. The complete C++ code for its computation is described in [4], and may be downloaded from the accompanying Internet site.

Further, thanks to the commutative properties of the $k$-mode tensor multiplication,[22] the following sum can be constructed for each mode matrix $\mathbf{S}_i$ in (9),

$$\boldsymbol{\mathcal{T}} = \sum_{h=1}^{N_P} \boldsymbol{\mathcal{T}}_h \times_P \mathbf{s}_P^h, \tag{11}$$

where the tensors

$$\boldsymbol{\mathcal{T}}_h = \boldsymbol{\mathcal{Z}} \times_1 \mathbf{S}_1 \times_2 \mathbf{S}_2 \ldots \times_{P-1} \mathbf{S}_{P-1} \tag{12}$$

create the orthogonal tensors basis. The vectors $\mathbf{s}_P^h$ are just columns of the unitary matrix $\mathbf{S}_P$. Because each $\boldsymbol{\mathcal{T}}_h$ is of dimension $P$-1 then $\times_P$ in (11) represents an outer product, which is a product of two tensors of dimensions $P$-1 and 1. Moreover, due to the mentioned all-orthogonality property (10) of the core tensor $\boldsymbol{\mathcal{Z}}$ in (12), $\boldsymbol{\mathcal{T}}_h$ are also orthogonal. Thanks to this, they span and orthogonal subspace which is used for pattern classification, as described,

$$\boldsymbol{\mathcal{T}} \times_P \mathbf{S}_P = \boldsymbol{\mathcal{Z}} \times_1 \mathbf{S}_1 \times_2 \mathbf{S}_2 \ldots \times_{P-1} \mathbf{S}_{P-1} = \boldsymbol{\mathcal{T}}_h. \tag{13}$$

Let us now observe that, thanks to the commutative properties of the $k$-mode multiplication, for each mode matrix $\mathbf{S}_i$ in (9), the following sum can be constructed

$$\boldsymbol{\mathcal{T}} = \sum_{h=1}^{N_P} \boldsymbol{\mathcal{T}}_h \times_4 \mathbf{s}_4^h. \tag{14}$$

Further, it can be shown that tensors

$$\boldsymbol{\mathcal{T}}_h = \boldsymbol{\mathcal{Z}} \times_1 \mathbf{S}_1 \times_2 \mathbf{S}_2 \times_3 \mathbf{S}_3, \tag{15}$$

in (11) constitute the basis tensors and $\mathbf{s}_P^h$ are columns of the unitary matrix $\mathbf{S}_P$. Thus, they form an orthogonal basis which spans a subspace. This property is used to construct a HOSVD based classifier.[4,5,22] However, in this case, they are *3D tensors*, as shown. This constitutes *a novelty* of the proposed method.

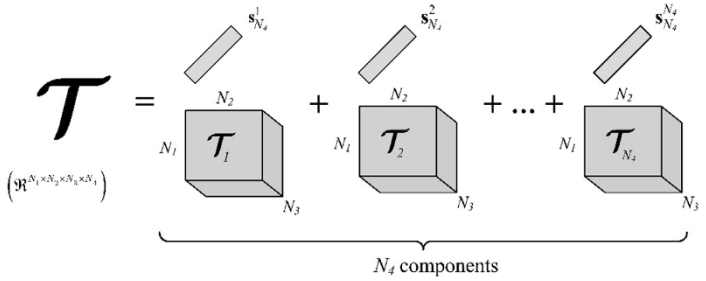


**Fig. 6** Orthogonal tensor subspace of a 4D prototype pattern tensor in Fig. 3. Decomposition allows representation of a 4D prototype pattern tensor as a sum of products of the 3D base tensors and the mode vectors. The base tensors span an orthonormal subspace which is used for pattern recognition.[5)]

In the tensor subspace, pattern recognition consists of computing a distance of a test pattern $\mathbf{P}_x$ to its projections onto each of the spaces spanned by the set of the bases $\boldsymbol{\mathcal{T}}_h$ in (12). As shown by Savas and Eldén, this can be written as the following minimization problem[22)]

$$\min_{i,c_h^i} \underbrace{\left\| \mathbf{P}_x - \sum_{h=1}^{H} c_h^i \boldsymbol{\mathcal{T}}_h^i \right\|}_{Q_i}^2 . \tag{16}$$

In the above, the scalars $c_h^i$ are unknown coordinates of $\mathbf{P}_x$ in the subspace spanned by $\boldsymbol{\mathcal{T}}_h^i$. The value of $H \leq N_P$ controls a number of chosen dominating components.

It can be shown that to minimize (16), the following value needs to be maximized

$$\hat{\rho}_i = \sum_{h=1}^{H} \left\langle \hat{\boldsymbol{\mathcal{T}}}_h^i, \hat{\mathbf{P}}_x \right\rangle^2 . \tag{17}$$

Thanks to the above, the HOSVD classifier returns a class $i$ for which its $\rho_i$ from (17) is the largest, as follows

$$\arg\max_i \left[\hat{\rho}(i)\right] = \sum_{h=1}^{H} \left\langle \hat{\boldsymbol{\mathcal{T}}}_h^i, \hat{\mathbf{P}}_x \right\rangle^2 . \tag{18}$$

The C++ implementation of the above classification algorithm is discussed in [4)]. Finally, for two tensors $\boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{T}}$, their inner product, as computed in (17), is defined as follows

$$\langle \boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{T}} \rangle = \boldsymbol{\mathcal{S}} \cdot \boldsymbol{\mathcal{T}} = \sum_{k_1=1}^{K_1} \sum_{k_2=1}^{K_2} \cdots \sum_{k_C=1}^{K_C} \mathbf{s}_{k_1 k_2 \ldots k_C} t_{k_1 k_2 \ldots k_C} . \tag{19}$$

It is easy to notice, that it is a sum of all possible products of pairs of components with the same indices and the result is a single real scalar value. Also, computation of (18)-(19) is very simple and can be easily parallelized to meet the real-time conditions, as will be shown in the next section.

As already mentioned, to increase system reliability, a ratio of the 2nd best to the 1st best response in (18) is also measured. That is, following condition is checked

$$1 - \frac{\hat{\rho}_{2nd}}{\hat{\rho}_{1st}} > \tau, \tag{20}$$

where $\hat{\rho}_{1st}, \hat{\rho}_{2nd}$ denote the first largest and the second largest value of the residuum computed in accordance with (17), respectively, and $\tau$ denotes a threshold value. In our experiments, the latter was set and measured in the range 0-5%. Setting the parameter $\tau$ to a value greater than 0 allows the classification system to deny a response if the answer cannot be given reliably enough. In this case, the testing dataset is portioned into two sets: the patterns with a reliable answer and the rejected patterns which are not classified. Accuracy is computed only from the former group as a ratio of correctly classified patterns to all in this group. On the other hand, the latter group of patterns can be left apart or fed to another classifier.

## §4 Experimental Results

All the algorithms realizing the classification chain shown in Fig. 2 and Fig. 3, were implemented in C++. For tensor processing, the *DeRecLib* software was used.[4] Experiments were run on the PC with 32 GB RAM and the i7-4800QM 4-core microprocessor. For the tests, the Medialab database was used.[17] Locations of the license plates, as well as vehicle logotypes where by us annotated for reference. Examples are depicted in Fig. 7. Thanks to this, the ground truth information was available which was then used to assess performance of the algorithms. As already mentioned, in this work, we focused on improvements to the classification module. Therefore, only logo classification results are discussed.

The ensemble depends on a number of parameters. However, their choice usually is not critical and suitable values can be easily determined based on experiments, as will be discussed. These are presented in Table 1.

The system was trained with 12 different logo types. For each brand, a different number of the training exemplar patterns were available. However, only images with well visible and sufficiently large logotypes were selected. For reliable recognition, the minimal size of the pattern was experimentally set to 30x30 pixels. A total of 112 test images were available and they were manually outlined for ground truth information. The logo training databases used in the presented experiments are depicted in Fig. 9.

Table 2 shows performance parameters for each of the tested classes. The confusion matrix contains true class in columns, and system response in rows. In this case the match separation threshold $\tau$ in (20), was set to 0. As already discussed, this means that all objects were classified and neither case was rejected


**Fig. 7** Examples of Annotated Images from the Medialab Database.[17)]

for classification. On the other hand, the $\tau$ parameter allows higher reliability of recognition at a cost of some patterns rejected from the classification. Performance of the method in respect to different settings of $\tau$ are shown in Table 3.

Figure 8 shows logo recognition performance as a plot of logo-acceptance-ratio $R$ versus 1-Accuracy (accuracy of nonrejected object, i.e., where (20) is higher than the given threshold) in the group of accepted objects, obtained by

**Table 1** Parameters of the key modules used in the tested vehicle logo classification system.

| *Module* | *Parameter* | *Values range* | *Values used in the presented experiments* |
|---|---|---|---|
| **EST** | Number of iterations in anisotropic diffusion | 1-5 | 2 |
| | | | |
| | Smoothing filter $G_R$ in (1) | Gaussian Binomial Anisotropic | Anisotropic |
| | Differentiating filter type in (2) | Sobel Simoncelli | Simoncelli |
| | Number of iterations in (6) | 5-25 | 9 |
| | | | |
| **Generator of deformable prototypes** (Fig. 3) | Type and parameters of deformations | Rotations 0 - $\pm 25^\circ$ | $\pm 18^\circ$ in steps $2^\circ$ |
| | Common pattern size | 32x32 - 164x164 | 64x64 |
| | | | |
| **HOSVD classifier** | Number of components $H$ in (18) | 20%-100% of all tensor bases | 75% |
| | Number of tested vehicle types | - | 12 |
| | Residuum ratio in (20) | 0 - 7% | 0% |

**Table 2** Confusion Matrix for Different Logo Classes (vertical - true class vs. horizontal response). Threshold $\tau = 0$.

| | **Alpha** | **Audi** | **BMW** | **Citroen** | **Fiat1** | **Hyundai** | **Opel** | **Peugeot** | **Renault** | **Seat** | **Toyota** | **VW** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Alpha** | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Audi** | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **BMW** | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Citroen** | 0 | 0 | 0 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Fiat1** | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Hyundai** | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Opel** | 0 | 0 | 0 | 0 | 0 | 0 | 19 | 0 | 0 | 0 | 0 | 0 |
| **Peugeot** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 10 | 0 | 0 | 0 | 0 |
| **Renault** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 |
| **Seat** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 12 | 2 | 0 |
| **Toyota** | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 14 | 0 |
| **VW** | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 23 |

changing the threshold $\tau$ in (20). This was changed in the range 0-5% with step of 1%.

The plot in Fig. 8 shows that with higher value of $\tau$, it is possible to obtain almost a perfect accuracy at a cost of some rejected logos. As already pointed out, this opens a way to incorporate other type of classifier to resolve such difficult classifications. Nevertheless, a choice of a proper classification strategy in this case is left for further research.

Recognition usually fails on patterns with poor contrast, low level of details (e.g. due to small size) or excessive noise or distortions (such as motion

**Table 3** Performance of the Method on the Annotated Database in the Function of Threshold $\tau$.

| Threshold $\tau$ | Accuracy in the group of classified logotypes | Number of objects rejected from classification |
|---|---|---|
| **0.00** | **0.95** | **0** |
| 0.01 | 0.96 | 5 |
| 0.02 | 0.99 | 9 |
| 0.03 | 1.00 | 13 |

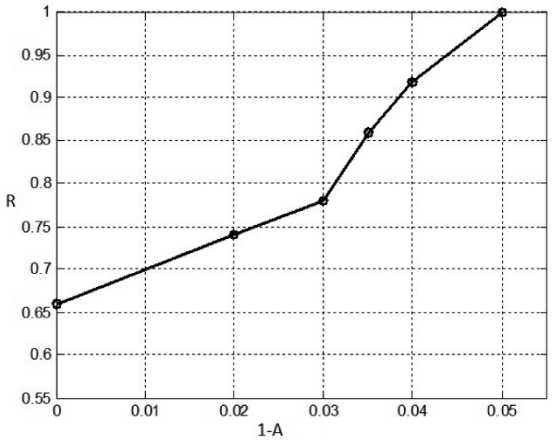


**Fig. 8** Logo-acceptance-ratio R vs. 1-*Accuracy* (1-A) in a function of the threshold $\tau$ in (20).

blur). Examples of images which were not correctly recognized are shown in Fig. 10. In the first of the shown image, the logo pattern is obscured and there is not enough details for the pattern to be recognized. Nevertheless, for human observers, a correct pattern can be told. In the second image, quality of the visible pattern is not high, which apparently makes the recognition process to fail.

Despite multidimensional patterns, the method shows relatively high speed of training. In the aforementioned setup, the whole training takes less than a minute. Even more interesting is very fast response of the system, which allows 50 responses per second, assuming pattern size of 64x64 pixels. The timings are shown in Table 4.

**Table 4** Average execution time for training and response in function of the prototype size.

| **Prototype size** | 64x64 | 128x128 | 164x164 |
|---|---|---|---|
| **Training** [s] | 53 | 57 | 61 |
| **Testing** [s] (for single pattern) | 0.02 | 0.03 | 0.03 |

We compared the above results with the results of the reported methods, which were mentioned in first section. The most advanced are the versions of the systems reported by Psyllos et al. in their works,[19–21] They also provided the Medialab database which were used in our experiments. They report using

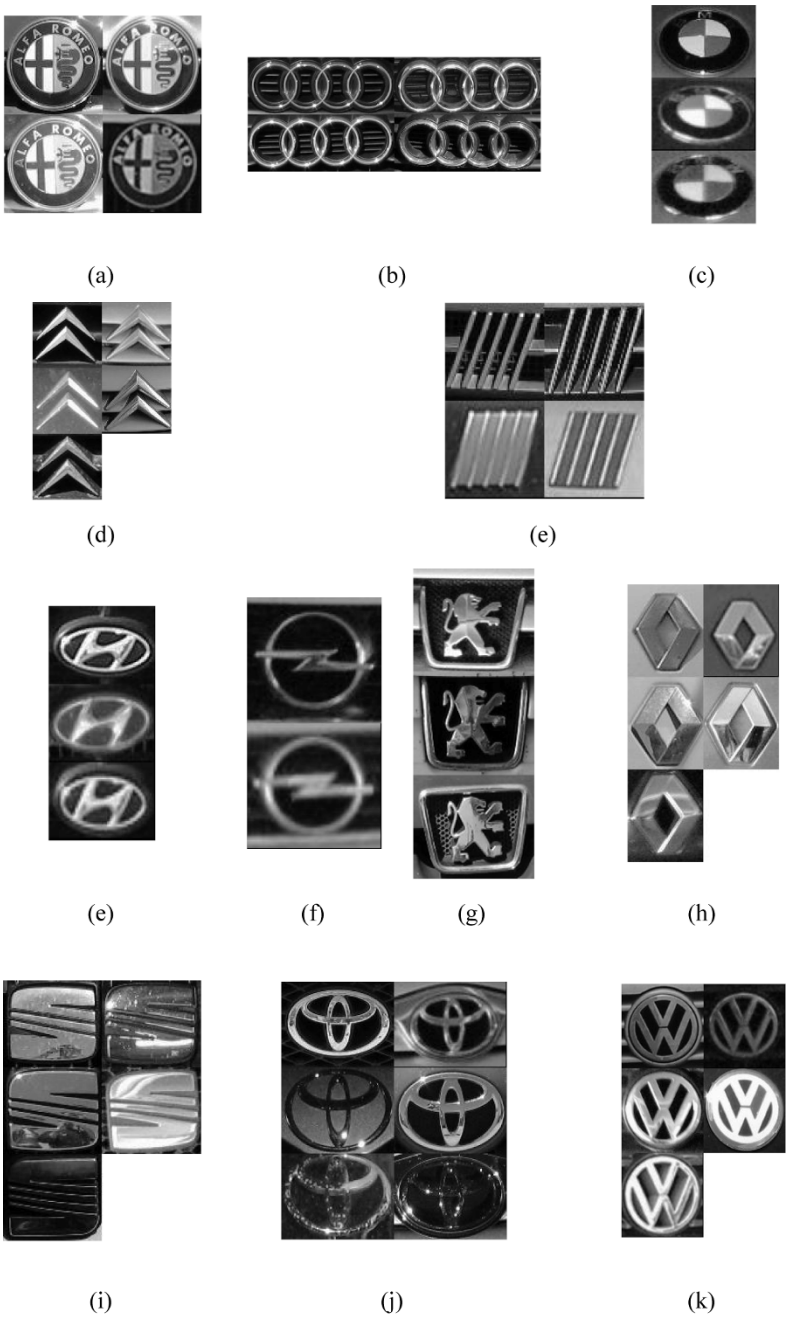


**Fig. 9** All databases used in our experiments. A total of 12 brands was analyzed (Alfa Romeo, Audi, BMW, Citroen, Fiat1, Hyundai, Opel, Peugeot, Renault, Seat, Toyota, and Volkswagen). The logotypes were downloaded from the Medialab database.

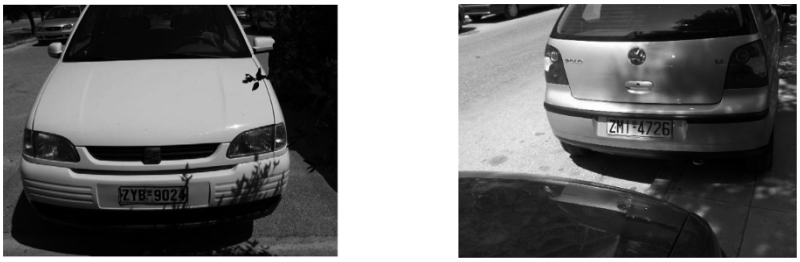

**Fig. 10** Incorrectly recognized logotypes from the Medialab database.[17] Despite sufficient size, in both images, insufficient quality caused not legible logotypes.

as many as 400 images, which are not all available from the Internet, however.[17] Also, their method, although shows very high recognition accuracy, reaching 91% and 94%, respectively,[19,21] relies on a different training method. As reported, for each of the classes, 40 images were selected to compute the reference keypoints and their descriptors (training). Then, a reference image needs to be chosen by an expert, whereas the remaining 39 images are registered in accordance with that reference view, using the homography calculated by the Ransac method. Then, only regions belonging to the common parts of the images are selected and the keypoint descriptors are recomputed to reflect the new position, scale, as well as orientation. At the end, all the keypoints for each brand form the reference database. In our method, the training procedure is simpler. For each brand, characteristic exemplars are chosen, as previously described. Also, to allow for skew views, the original database is endowed with such exemplars, if available. Then, all the patterns are geometrically deformed to reflect possible slight deviations of the testing patterns. Finally, they form the PPT which is processed with the EST followed by the HOSVD, as described. The database can be easily updated, since each pattern is treated separately. Nevertheless, the group of deformations is limited to the available views and their slight deformations. As shown in Table 3, the obtained accuracy of the presented method reaches 95%, however on much smaller sample than in the systems.[19,21] Last but not least, the response time of the proposed method is very competitive. As measured, the method allows response of about 50 patterns per second, in pure software implementation. This is due to the simple testing rule, expressed by the sum of inner products in [18]. Such operations can be very effectively implemented and parallelized on modern microprocessors. In our system, the microprocessor is endowed with 8 cores. In the method reported by Psyllos et al., the recognition rate is at least 10 times slower.

Concluding, we can list the following positive aspects of the proposed method:

1. High accuracy;
2. Very fast response;
3. Fast training;
4. Training possible with any set of training prototypes (even single exem-

plars possible).
5. Possible on-line modification to the database.

On the other hand, there are also some problems, which can be summarized as follows:

1. The system depends on database of the prototype exemplars.
2. It works relatively well in the case of well visible logotypes and in daily conditions.
3. The method does not work well with small patterns (in our experiments, the minimal size was constrained to be higher than 35x35 pixels).

Further work will focus on building a larger database which will be used for training and testing. Also, a very interesting research subject would be to join the super-resolution method, as well as other classification methods, with the proposed ensemble. Last but not least, the further research should aim at improvement and joining of the suitable detection module.

## §5 Conclusions

In this paper, the hybrid system for vehicle logo recognition is presented. The system operates as an ensemble of tensor based classifiers. Our approach consists of transforming the training patterns into a group representing a local manifold of pattern deformations. Each pattern is then processed by the extended structural tensor and composed into the representative prototype pattern tensor. Based on such representation, a tensor orthogonal subspace is constructed by higher-order singular value decomposition method. The unique feature of this approach is the orthogonal sub-space in which bases are 3D tensors. The system is organized as a twofold classifier ensemble, in which the first group of the individual classifiers is responsible for classification of single training prototype exemplar. The second level is composed of classifiers for the same logo type. All these are orchestrated by the combination rules operating in the maximum rule. As shown by the experiments, the method shows high accuracy and very rapid response. Therefore, it is competitive to other reported methods. In further research, we plan to extend our own training and testing database. Moreover, we plan to add new types of classifiers to the ensemble which will further improve on accuracy.

## *Acknowledgements*

The financial support from the Polish National Science Centre NCN in the year 2014, contract no. DEC-2011/01/B/ST6/01994, is greatly acknowledged.

## *References*

1) Bigün, J., Granlund, G. H., Wiklund, J., "Multidimensional Orientation Estimation with Applications to Texture Analysis and Optical Flow," *IEEE PAMI, 13, 8*, pp. 775–790, 1991.

2) Brox, T., Boomgaard van den, R., Lauze, F., Weijer van de, J., Weickert, J., Mrázek, P., Kornprobst, P., "Adaptive Structure Tensors and their Applications, in *Visualization and Processing of Tensor Fields* (Weickert, J., Hagen, H. ed.), Springer, pp. 17–47, 2006.
3) Bugaj, M., Cyganek, B., "GPU Based Computation of the Structural Tensor for Real-Time Figure Detection," in *Proc. of the 20th International Conference on Computer Graphics, Visualization and Computer Vision (WSCG'12)*, Czech Republic, 2012.
4) Cyganek, B., *Object Detection and Recognition in Digital Images: Theory and Practice*, Wiley, 2013.
5) Cyganek, B., "Object Recognition with the Higher-Order Singular Value Decomposition of the Multi-Dimensional Prototype Tensors," *3rd Computer Science On-line Conference (CSOC 2014). Advances in Intelligent Systems and Computing*, Springer, pp. 395–405, 2014.
6) Cyganek, B., Woźniak, M., "Vehicle Logo Recognition with an Ensemble of Classifiers," *ACIIDS 2014, Part II, LNAI 8398* (Nguyen, N. T. et al. eds.), pp. 117–126, 2014.
7) Dai, S., Huang, H., Gao, Z., "Vehicle-logo recognition method based on Tchebichef moment invariants and SVM," *Software Engineering, WCSE'09*, pp. 18–21, 2009.
8) Jähne, B., *Digital Image Processing (6th edition)*, Springer-Verlag, 2005.
9) Huber, P. J., *Robust statistics*, New York, John Wiley 1981.
10) Ki, Y., Baik, D., "Vehicle-classification algorithm for single-loop detectors using neural networks," *Vehicular Technology, IEEE Transactions on, 55, 6*, pp. 1704–1711, 2006.
11) Kolda, T. G., Bader, B. W., "Tensor Decompositions and Applications," *SIAM Review*, pp. 455–500, 2008.
12) Krawczyk, B., Woźniak, M., Herrera, F., "On the Usefulness of One-Class Classifier Ensembles for Decomposition of Multi-Class Problems," *Elsevier Pattern Recognition, doi:10.1016/j.patcog.2015.06.001*, 2015.
13) Krawczyk, B., "One-class classifier ensemble pruning and weighting with firefly algorithm," *Neurocomputing, 150*, Elsevier, pp. 490–500, 2015.
14) Lathauwer, de L., "Signal Processing Based on Multilinear Algebra," Ph. D. dissertation, Katholieke Universiteit Leuven, 1997.
15) Lathauwer, de L., Moor de, B., Vandewalle, J., "A Multilinear Singular Value Decomposition," *SIAM Journal of Matrix Analysis and Applications, 21, 4*, pp. 1253–1278, 2000.
16) Leys, C., Ley, C., Klein, O., Bernard, P., Licata, L., "Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median," *Journal of Experimental Social Psychology, 49, 4*, pp. 764–766, July 2013.
17) Medialab LPR database, *http://www.medialab.ntua.gr/research/LPRdatabase.html*, 2015.
18) Petrovic, V. S., Cootes, T. F., "Analysis of features for rigid structure vehicle type recognition," in *Proc. BMVC*, 2004.
19) Psyllos, A. P., Anagnostopoulos, C.-N. E., Kayafas, E., "Vehicle logo recognition using a sift-based enhanced matching scheme," *IEEE ITS, 11, 2*, pp. 322–328, 2010.

20) Psyllos, A., Anagnostopoulos, C., Kayafas, E., "Vehicle model recognition from frontal view image measurements," *Computer Standards & Interfaces, 33, 2*, pp. 142–151, 2011.

21) Psyllos, A., Anagnostopoulos, C.-N., Kayafas, E., "M-SIFT: A new method for Vehicle Logo Recognition, *2012 IEEE International Conference on Vehicular Electronics and Safety, Istanbul, Turkey*, 2012.

22) Savas, B., Eldén, L., "Handwritten digit classification using higher order singular value decomposition," *Pattern Recognition, 40*, pp. 993–1003, 2007.

23) Sulehria, H., Zhang, Y., "Vehicle logo recognition using mathematical morphology," *Proc. of the 6th WSEAS Intl Conf. on Telecommunications and Informatics*, 2007.

24) Turk, M., Pentland, A., "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, 3, 1, pp. 71–86, 1991.

25) Luis-García, R., Deriche, R., Rousson, M., Alberola-López, C., "Tensor Processing for Texture and Colour Segmentation," *LNCS, 3540*, pp. 1117–1127, 2005.

26) Meyer, C.D., *Matrix Analysis and Applied Linear Algebra*, SIAM, 2000.

27) Perona, P., Malik, J., "Scale-Space and Edge Detection Using Anisotropic Diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence, 12, 7*, pp. 629–639, July 1990.

28) Sapiro, G., *Geometric Partial Differential Equations and Image Analysis*, Cambridge University Press, 2001.

29) Woźniak, M., Grana, M, Corchado, E., "A survey of multiple classifier systems as hybrid systems," *Information Fusion, 16*, pp. 3–17, March 2014.

**Bogusław Cyganek, Ph.D.:** He received his M.Sc. degree in electronics in 1993, and then M.Sc. in computer science in 1996 from the AGH University of Science and Technology, Krakow, Poland. He obtained his Ph.D. degree cum laude in 2001 with a thesis on correlation of stereo images, and D.Sc. degree in 2011 with a thesis on methods and algorithms of object recognition in digital images. His research interests include computer vision, pattern recognition, as well as development of programmable devices and embedded systems.

**Michal Wozniak, Ph.D.:** He is a full professor of computer science at the Department of Systems and Computer Networks, Wrocław University of Technology, Poland. He received M.Sc. degree in biomedical engineering from the Wrocław University of Technology in 1992, and Ph.D. and D.Sc. (habilitation) degrees in computer science in 1996 and 2007, respectively, from the same university. His research focuses on compound classification methods, hybrid artificial intelligence and medical informatics.