Information Sciences 327 (2016) 175–182

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Fast image classification by boosting fuzzy classifiers

Marcin Korytkowski, Leszek Rutkowski, Rafał Scherer*

Institute of Computational Intelligence, Częstochowa University of Technology, al. Armii Krajowej 36, 42–200 Częstochowa, Poland

ARTICLE INFO

Article history:
Received 24 March 2015
Revised 21 June 2015
Accepted 14 August 2015
Available online 24 August 2015

Keywords:
Visual object categorization
Fuzzy classification
Boosting meta-learning

ABSTRACT

This paper presents a novel approach to visual objects classification based on generating simple fuzzy classifiers using local image features to distinguish between one known class and other classes. Boosting meta-learning is used to find the most representative local features. The proposed approach is tested on a state-of-the-art image dataset and compared with the bag-of-features image representation model combined with the Support Vector Machine classification. The novel method gives better classification accuracy and the time of learning and testing process is more than 30% shorter.

1. Introduction

The most popular way to search vast collections of images and video which are generated every day in a tremendous amount is realized by keywords and meta tags or just by browsing them. Emergence of content-based image retrieval (CBIR) in the 1990s enabled automatic retrieval of images to a certain extent. Various CBIR tasks include searching for images similar to the query image or retrieving images of a certain class [7,9,18,19,23,24,26,35,50,51,53] and classification [1,3,6,16,17,20,25,43,52] of the query image. Such content-based image matching remains a challenging problem of computer science. Image matching consists of two relatively difficult tasks: identifying objects on images and fast searching through large collections of identified objects. Identifying objects on images is still a challenge as the same objects and scenes can be viewed under different imaging conditions. There are many previous works dedicated to the problem formulated in this way. Some of them are based on color representation [13,22,33], textures [5,11,15,45], shape [14,21,47] or edge detectors [31,32,56]. Recently local invariant features have gained a wide popularity [27–30,44]. The most popular local keypoint detectors and descriptors are SURF [2], SIFT [27] or ORB [36].

The second substantial problem is a fast retrieval of identified objects. To find similar images to a query image, we need to compare all feature descriptors of all images usually by some distance measures. Such comparison is enormously time consuming and there is ongoing worldwide research to speed up the process. Yet, the current state of the art in the case of high dimensional computer vision applications is not fully satisfactory. The literature presents countless methods and variants utilizing e.g. a voting scheme or histograms of clustered keypoints. They are mostly based on some form of approximate search. One of the solutions to the problem can be descriptor vector hashing. In [8] the authors proposed a locality-sensitive hashing method for the approximate nearest neighbor algorithm. In [30] the authors built a hierarchical quantizer in the form of a tree. Such tree is a kind of an approximate nearest neighbor algorithm and constitutes a visual dictionary. Recently, the bag-of-features (BoF) approach [12,34,44,49,54] has gained in popularity. In the BoF method clustered vectors of image features are collected and

* Corresponding author. Tel.: +48601365426.
E-mail addresses: marcin.korytkowski@iisi.pcz.pl (M. Korytkowski), leszek.rutkowski@iisi.pcz.pl (L. Rutkowski), rafal.scherer@iisi.pcz.pl (R. Scherer).
URL: http://iisi.pcz.pl/ (R. Scherer)

http://dx.doi.org/10.1016/j.ins.2015.08.030

sorted by the count of occurrence (histograms). There are some modifications of this method, for example a solution that uses earth movers distance (EMD) presented in [12]. The main problem with the aforementioned approaches is that all individual descriptors or approximations of sets of descriptors presented in the histogram form must be compared. Such calculations are very computationally expensive. Moreover, the BoF approach requires redesign of the classifier when new visual classes are added to the system. In this paper we will overcome these limitations and we will present a novel method to classify fast various images in large collections on the basis of their content. Our method is partly inspired by the ideas of Viola et al. [46,48,55]. They used a modified version of the AdaBoost algorithm to select the most important features from a large number of very simple rectangular features similar to Haar basis functions. The selected features are treated by the authors of [46,48] as weak classifiers for the content-based image retrieval task, mainly images containing faces. Contrary to the previous authors, who developed CBIR systems based on boosting techniques, in the approach proposed in this paper: (i) we use the original version AdaBoost algorithm to choose the most important local features; (ii) a wide variety of local and global visual features descriptors (e.g. SURF SIFT or ERB) can be incorporated into our classifier; (iii) our method is applicable to a wider class of images (not only face images); (iv) incorporating new visual classes in the system requires only adding new fuzzy rules to the rule base without restructuring the existing rule base. In this paper we propose a novel approach to use fuzzy logic and fuzzy rules as the adjustable representation of visual feature clusters. Fuzzy logic [37–39,41,42] is a very convenient method for describing partial membership to a set. This allows creating a very efficient method for fast object classification in large databases. We combine fuzzy logic and boosting meta-learning to choose the most representative set of image features for every considered class of objects. In each step we randomly choose one feature from a set of positive images taking into consideration feature weights computed using the AdaBoost algorithm. This feature constitutes a base to build a weak classifier. The weak classifier is given in the form of a fuzzy rule and the selected feature is a base to determine the initial parameters of the fuzzy rule. In the learning process the weak classifiers are adjusted to fit positive image examples. This approach could be very useful for the search based on the image content in a set of complex graphical objects in a database. The main contribution and novelty of the paper is as follows:

- We present a novel method for automatic building a fuzzy rule base for image classification based on local features; the method does not require knowledge of any initial parameters, contrary to e.g. the popular BoF method which requires specifying the dictionary size.
- We develop an efficient technique for fast classification of images, in particular the learning and the testing time of our method is, respectively, 35% and 32% shorter than in the case of BoF.
- We propose a method for automatic search of the most salient local features for a given class of images.
- We design a flexible system; expanding the system knowledge is efficient because adding new visual classes to the system requires only adding new fuzzy rules whereas in the case of BoF it requires the whole new dictionary generation and re-learning of classifiers.

The paper is organized as follows. Section 2 describes the proposed method of creating the weak classifier ensemble as a fuzzy rule base. The method of classifying new query images is presented in Section 3. Section 4 provides simulation results on the PASCAL Visual Object Classes (VOC) 2012 dataset [10] to compare the proposed method with the BoF method combined with Support Vector Machines classifiers.

2. Boosting-generated simple fuzzy classifiers

The main idea of this paper is to find the most representative fuzzy rules for a given class ω_c, $c = 1, \ldots, V$, of visual objects and to fast classify query images afterwards. This section describes the learning process i.e. generating fuzzy rules from a set of examples. The algorithm uses the boosting meta-learning to generate suitable number of weak classifiers. The classifiers feature space $\mathbb{R}^N$ consists of elements x_n, $n = 1, .., N$. For example, in the case of using the standard SIFT descriptors, $N = 128$.

In each step we randomly choose one local feature from the set of positive images according to its boosting weight. Then we search for similar feature vectors from all positive images. Using these similar features we construct one fuzzy rule. Undoubtedly, it is impossible to find exactly the same features in all images from the same class, thus we search for feature vectors which are similar to the feature picked randomly in the current step. This is the one of the reasons for using fuzzy sets and fuzzy logic. The rules have the following form:

$$\begin{aligned} &R^c_t : \text{IF } x_1 \text{ is } G^c_{1,t} \text{ AND } x_2 \text{ is } G^c_{2,t} \text{ AND} \ldots \\ &\ldots \text{ AND } x_{128} \text{ is } G^c_{128,t} \text{ THEN image } i \in \ \omega_c(\beta^c_t), \end{aligned} \tag{1}$$

where $t = 1, \ldots, T^c$ is the rule number in the current run of boosting, T^c is the number of rules for the class ω_c and β^c_t is the importance of the classifier, designed to classify objects from the class ω_c, created in the tth boosting run. The weak classifiers (1) consist of fuzzy sets with Gaussian membership functions

$$G^c_{n,t}(x) = e^{-\left(\frac{x-m^c_{n,t}}{\sigma^c_{n,t}}\right)^2}, \tag{2}$$

where $m^c_{n,t}$ is the center of the Gaussian function (2) and $\sigma^c_{n,t}$ is its width. For the clarity of presentation this section describes generating the ensemble of weak classifiers for a class ω_c, thus the class index c will be omitted.

Let I be the number of all images in the learning set, divided into two sets: positive images and negative images, having, respectively, I_{pos} and I_{neg} elements. Obviously $I = I_{pos} + I_{neg}$. Positive images belong to a class ω_c that we train our classifier with. For every image from these two sets we determine local features, for example local interest points using e.g. SIFT or SURF algorithms. The points are represented by descriptors, and we operate on two sets of vectors: positive descriptors $\{\mathbf{p}^i; i = 1, .., L_{pos}\}$ and negative ones $\{\mathbf{n}^j; j = 1, .., L_{neg}\}$. In the case of the standard SIFT algorithm, each vector $\mathbf{p}^i$ and $\mathbf{n}^j$ consists of 128 real values. Let v^i be the number of keypoint vectors in the ith positive image, let u^j be the number of keypoint vectors in the jth negative image. Then, the total number of learning vectors is given by

$$L = \sum_{i=1}^{I_{pos}} v^i + \sum_{j=1}^{I_{neg}} u^j, \tag{3}$$

where $L = L_{pos} + L_{neg}$. According to the AdaBoost algorithm, we have to assign a weight to each keypoint in the learning set. When creating new classifiers the weights are used to indicate keypoints which were difficult to handle. At the start of the algorithm, all the weights have the same, normalized values

$$D_1^l = \frac{1}{L} \quad \text{for } l = 1, \ldots, L. \tag{4}$$

Let us define matrices $\mathbf{P}_t$ and $\mathbf{N}_t$ constituting the learning set

$$\mathbf{P}_t = \begin{bmatrix} \mathbf{p}^1 & D_t^1 \\ \vdots & \vdots \\ \mathbf{p}^{L_{pos}} & D_t^{L_{pos}} \end{bmatrix} = \begin{bmatrix} p_1^1, \ldots, p_N^1 & D_t^1 \\ \vdots & \vdots \\ p_1^{L_{pos}}, \ldots, p_N^{L_{pos}} & D_t^{L_{pos}} \end{bmatrix}, \tag{5}$$

$$\mathbf{N}_t = \begin{bmatrix} \mathbf{n}^1 & D^1 \\ \vdots & \vdots \\ \mathbf{n}^{L_{neg}} & D_t^{L_{neg}} \end{bmatrix} = \begin{bmatrix} n_1^1, \ldots, p_N^1 & D^1 \\ \vdots & \vdots \\ n_1^{L_{neg}}, \ldots, p_N^{L_{neg}} & D_t^{L_{neg}} \end{bmatrix}. \tag{6}$$

The learning process consists in creating T simple classifiers (weak learners in boosting terminology) in the form of fuzzy rules (1). After each run $t, t = 1, \ldots, T$, of the proposed algorithm, one fuzzy rule R_t is obtained. The process of building a single fuzzy classifier is presented below.

1. Randomly choose one vector $\mathbf{p}^r, 1 \leq r \leq L_{pos}$ from positive samples using normalized distribution of elements $D_t^1, \ldots, D_t^{L_{pos}}$ in matrix (5). This drawn vector becomes a basis to generate a new classifier and the learning set weights contribute to the probability of choosing a keypoint.
2. For each image from the positive image set find the feature vector which is nearest to $\mathbf{p}^r$ (for example according to the Euclidean distance) and store this vector in matrix $\mathbf{M}_t$ of the size $I_p \times N$. Every row represents one feature from a different image $v_i, i = 1, \ldots, I_{pos}$, and no image occurs more than once

$$\mathbf{M}_t = \begin{bmatrix} \tilde{p}_{t,1}^1 & \cdots & \tilde{p}_{t,N}^1 \\ \vdots & \cdots & \vdots \\ \tilde{p}_{t,1}^j & \ddots & \tilde{j}_{t,N}^j \\ \vdots & \cdots & \vdots \\ \tilde{p}_{t,1}^{I_{pos}} & \cdots & \tilde{p}_{t,N}^{I_{pos}} \end{bmatrix}, \tag{7}$$

Each vector $\begin{bmatrix} \tilde{p}_{t,1}^j & \cdots & \tilde{p}_{t,N}^j \end{bmatrix}$, $j = 1, \ldots, I_{pos}$, in matrix (7) contains one visual descriptor from the set $\{\mathbf{p}^i; i = 1, .., L_{pos}\}$. For example, in view of descriptions (5) and (3), the first row in matrix (7) is one of the rows of the following matrix

$$\begin{bmatrix} p_1^1, \ldots, p_N^1 & \\ \vdots & \vdots \\ p_1^{v_1}, \ldots, p_N^{v_1} & \end{bmatrix}, \tag{8}$$

where v_1 is the number of feature vectors in the first positive image.
3. In this step a weak classifier is built, i.e. we find centers and widths of Gaussian functions which are membership functions of fuzzy sets in a fuzzy rule (1).
   (a) Compute the absolute value $d_{t,n}$ as the difference between the smallest and the highest values in each column of the matrix (7)

$$d_{t,n} = |\min_{i=1,\ldots,I_p} p_n^i - \max_{i=1,\ldots,I_p} p_n^i| \tag{9}$$

where $n = 1, \ldots, N$. Compute the center of fuzzy Gaussian membership function (2) $m_{t,n}$ in the following way

$$m_{t,n} = \max_{i=1,\ldots,I_p} p_n^i - \frac{d_{t,n}}{2}. \tag{10}$$

Now we have to find the widths of these fuzzy set membership functions. We have to assume that for all real arguments in the range of $\left[m_{t,n} - \frac{d_{t,n}}{2}; m_{t,n} + \frac{d_{t,n}}{2}\right]$ the Gaussian function (fuzzy set membership function) values should satisfy $G_{n,t}(x) \geq 0.5$. Only in this situation do we activate the fuzzy rule. As we assume that $G_{n,t}(x)$ is at least 0.5 to activate a fuzzy rule, using simple substitution $x = m_{t,n} - \frac{d_{t,n}}{2}$, we obtain the relationship for $\sigma_{t,n}$

$$\sigma_{t,n} = \frac{d_{t,n}}{2\sqrt{-\ln(0.5)}} \tag{11}$$

Finally, we have to calculate the values $m_{t,n}$ and $\sigma_{n,t}$ for every element of the nth column of matrix (7), thus we have to repeat the above steps for all N dimensions. In this way, we obtain N Gaussian membership functions of N fuzzy sets. Of course, we can label them using fuzzy linguistic expressions such as 'small', 'large' etc., but for the time being we mark them only in a mathematical sense by $G_{n,t}$, where n, $n = 1, .., N$, is the index associated with feature vector elements and t means the fuzzy rule number.

(b) Using values obtained in point a) we can construct a fuzzy rule which creates a fuzzy classifier (1).

4. Now we have to evaluate the quality of the classifier obtained in step 3. We do this using the standard AdaBoost algorithm [40]. Let us determine the activation level of the rule R_t which is computed by a t-norm of all fuzzy sets membership function values

$$f_t(\bar{\mathbf{x}}) = \mathop{T}_{n=1}^{N} G_{n,t}(\bar{x}_n), \tag{12}$$

where $\bar{\mathbf{x}} = [\bar{x}_1, \ldots, \bar{x}_N]$ is a vector of values of linguistic variables $x_1, \ldots, x_N$. In the case of minimum t-norm formula (12) becomes

$$f_t(\bar{\mathbf{x}}) = \min_{n=1}^{N} G_{n,t}(\bar{x}_n). \tag{13}$$

As a current run of the AdaBoost is for a given class ω_c, we can treat the problem as a binary classification (dichotomy) i.e. $y^l = 1$ for descriptors of positive images and $y^l = 0$ for descriptors of negative images. Then the fuzzy classifier decision is computed by

$$h_t(\bar{\mathbf{x}}^l) = \begin{cases} 1 & \text{if} \quad f_t(\bar{\mathbf{x}}^l) \geq \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}. \tag{14}$$

For all the keypoints stored in matrices $\mathbf{P}_t$ and $\mathbf{N}_t$ we calculate new weights D_t^l. To this end, we compute the error of classifier (14) for all $L = L_{pos} + L_{neg}$ descriptors of all positive and negative images

$$\varepsilon_t = \sum_{l=1}^{L} D_t^l I(h_t(\bar{\mathbf{x}}^l) \neq y^l), \tag{15}$$

where I is the indicator function

$$I(a \neq b) = \begin{cases} 1 & \text{if} \quad a \neq b \\ 0 & \text{if} \quad a = b \end{cases}. \tag{16}$$

If $\varepsilon_t = 0$ or $\varepsilon_t > 0.5$, we finish the training stage. If not, we compute new weights:

$$\alpha_t = 0.5 \ln \frac{1 - \varepsilon_t}{\varepsilon_t}. \tag{17}$$

$$D_{t+1}^l = \frac{D_t^l \exp\{-\alpha_t I(h_t(\bar{\mathbf{x}}^l) = y^l)\}}{C}, \tag{18}$$

where C is a constant such that $\sum_{l=1}^{L} D_{t+1}^l = 1$. Finally classifier importance is determined by

$$\beta_t = \frac{\alpha_t}{\sum_{t=1}^{T} \alpha_t}. \tag{19}$$

Remark. It should be noted that the classifier importance (19) is needed to compute overall response of the boosting ensemble for the query image, which will be described in detail in the next section.

Remark. The concept of 'word' used in the BoW method [12,34,44,54] corresponds to a fuzzy rule in the presented method, which in the case of the SIFT application, consists of 128 Gaussian functions.

The next section will describe a classification of a new query image by the ensemble.

3. Classification of a query image

The boosting procedure described in the previous section should be executed for every visual object class ω_c, $c = 1, \ldots, V$, thus after the learning procedure we obtain a set of V strong classifiers. Let us assume that we have a new query image and an associated set of u visual features represented by matrix $\mathbf{Q}$

$$\mathbf{Q} = \begin{bmatrix} \mathbf{q}^1 \\ \mathbf{q}^2 \\ \vdots \\ \mathbf{q}^u \end{bmatrix} = \begin{bmatrix} q_1^1 \cdots q_N^1 \\ q_1^2 \cdots q_N^2 \\ \vdots \\ q_1^u \cdots q_N^u \end{bmatrix}. \tag{20}$$

Let us determine the value of

$$F_t(\mathbf{Q}) = \mathop{S}_{j=1}^{u} \left(\mathop{T}_{n=1}^{N} G_{n,t}(q_n^j) \right), \tag{21}$$

where S and T are t-norm and t-conorm, respectively (see [39]). To compute the overall output of the ensemble of classifiers designed in Section 2, for each class ω_c we sum weak classifiers outputs (21) taking into consideration their importance (19), i.e.

$$H^c(\mathbf{Q}) = \sum_{t=1}^{T^c} \beta_t F_t(\mathbf{Q}). \tag{22}$$

Eventually, we assign a class label to the query image in the following way

$$f(\mathbf{Q}) = \arg \max_{c=1,\ldots,V} H^{*c}(\mathbf{Q}). \tag{23}$$

In formulas (22) and (23) we restored class label index c, which had been removed at the beginning of Section 2. In formula (21) t-norm and t-conorm can be chosen as min and max operators, respectively.

4. Experiments

The goal of the experiments was to evaluate the proposed approach and compare it with the state-of-the-art method in terms of accuracy and speed. We tested the proposed method on three classes of visual objects taken from the PASCAL Visual Object Classes (VOC) dataset [10], namely: Bus, Cat and Train. Examples of such visual objects are presented in Fig. 1. We divided these three classes of objects into learning and testing examples. The testing set consists of 15% images from the whole dataset. Before the learning procedure we generated local keypoint vectors for all images from the Pascal VOC dataset using the SIFT algorithm. These 128-element vectors were stored in separate files for every image in the dataset. Each file contained hundreds of vectors, depending on the complexity of the image.

All simulations were performed on a Hyper-V virtual machine with MS Windows Operating System (8 GB RAM, Intel Xeon X5650, 2.67 GHz). We determined the time and quality of classification. The testing set contained only images that had never been presented to the system during learning process.

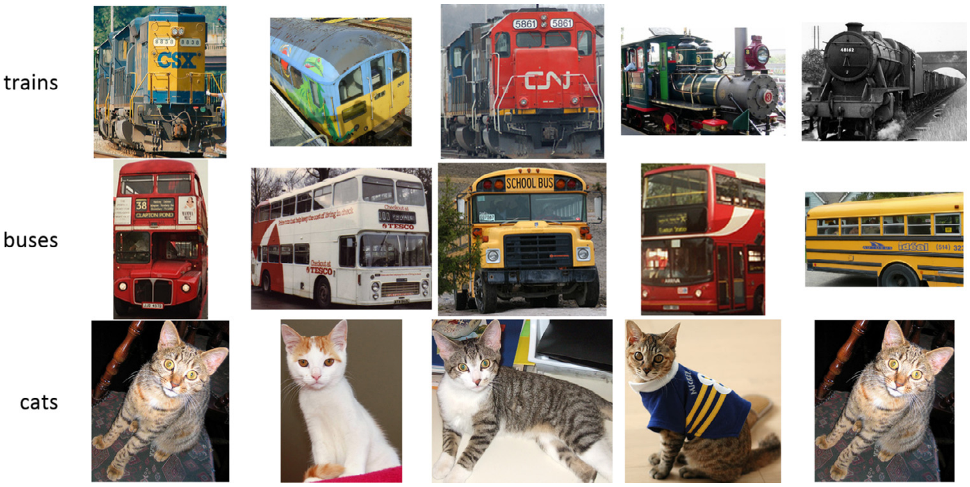

Fig. 1. Examples of objects from Bus, Cat and Train class taken from PASCAL Visual Object Classes (VOC) dataset.

Table 1
Results of the learning and testing processes for the BoF and SVM algorithms (CQ - Classification Quality ([%]), LT - Learning time ([s]), TT - Testing time ([s])). The learning time is given only as the overall time for all classes.

	CQ	LT	TT
	Dictionary size: 200		
Buses	70.59%		3.532
Cats	100%		5.199
Trains	41.17%		4.833
Total	41.18%	195.57	13.564
	Dictionary size: 250		
Buses	70.59%		3.627
Cats	88.24%		5.858
Trains	35.29%		5.177
Total	64.71%	208.241	14.662
	Dictionary size 300		
Buses	76.47%		3.678
Cats	88.24%		5.734
Trains	41.18%		5.134
Total	68.63%	213.317	14.546
	Dictionary size 350		
Buses	70.59%		3.696
Cats	94.12%		5.862
Trains	52.94%		5.436
Total	72.55%	246.48	14.994
	Dictionary size 400		
Buses	70.59%		4.116
Cats	88.24%		6.136
Trains	52.94%		5.344
Total	70.59%	265.469	15.596

Table 2
Classification accuracy and time of the learning and testing processes obtained by the method proposed in this paper (CQ - Classification Quality ([%]), LT - Learning time ([s]), TT - Testing time ([s])). The learning time is given only as the overall time for all classes.

	Positive learning samples	Negative learning samples	Classif. accuracy on testing set	Learning time ([s])	Testing time: ([s])
Buses	76	17	82.35%		3.236
Cats	82	17	76.47%		4.495
Trains	73	17	64.71%		3.593
Total	231	51	74.51%	182.117	11.324

The method proposed in the paper was implemented using C# language. For the learning process needs, we built an extra set of examples (negative examples) for each considered class of objects. The negative examples were picked by random from other classes. We chose only the most representative objects for considered classes, whereas for the testing purposes we chose various kinds of images from a considered class.

We have compared our results with the bag-of-features image representation model combined with the Support Vector Machine (SVM) classification based on the Chi-Square kernel. The BoF algorithm is currently one of the most popular algorithms in computer vision and it was run five times for various dictionary sizes: 200, 250, 300, 350 and 400 words. Dictionaries for the BoF were created using C++ language, based on the OpenCV Library [4]. The BoF experiments were performed on the same set of objects as the experiments for the method proposed in the paper. The results of the BoF and SVM classification, both learning and testing, are presented in Table 1. In the BoF algorithm the learning process (dictionary generation) is run globally for all classes, thus column LT is empty in Table 1 for each class. In Table 2 we depict simulation results of the method described in Sections 2 and 3. The BoF combined with the SVM algorithm achieved the best overall classification accuracy for three classes for the dictionary of size 350 (Table 1), which was approximately worse by 2% than in the case of the proposed method. Moreover, the learning and classification time for the proposed method is considerably shorter than the BoF-SVM (182.117 s vs. 246.48 s), which is better, respectively by 35 and 32%. The dictionaries of other sizes performed slightly worse than the dictionary of size 350. It can be clearly seen that our method gives a better classification accuracy and the time of learning and testing process is shorter. It should be also emphasized that our method has an extra advantage; namely we can add a new class of visual objects to the existing system by just adding new rules. In the BoF we have to recreate the whole dictionary. The most time-consuming part of the bag-of-features classification is the SVM learning.

5. Conclusions

We proposed a new approach to fast image classification. Our approach, which works by repeatedly creating fuzzy rules based on most salient image features, has shown promising results on a real-world dataset. Despite its simplicity, it outperformed the bag-of-features method in terms of accuracy and speed. It demonstrates the following advantages:

- the method is relatively accurate in terms of visual object classification,
- learning and classification is very fast,
- expanding the system knowledge is efficient as adding new visual classes to the system requires generation of new fuzzy rules whereas in the case of bag-of-features it requires new dictionary generation and re-learning of classifiers.

The method also demonstrates a potential in terms of possibility to expand it in order to incorporate different features or different meta-learning algorithms.

It should be noted that the system can work with virtually any type of fuzzy membership functions, e.g. triangular or bell-shape. Moreover, various types of t-norm can be used in the algorithm but the application of the minimum t-norm leads to faster computation than in the case of other t-norms.

Acknowledgment

This work was supported by the Polish National Science Centre (NCN) under project number DEC-2011/01/D/ST6/06957.

References

[1] A. Akusok, Y. Miche, J. Karhunen, K.-M. Bjork, R. Nian, A. Lendasse, Arbitrary category classification of websites based on image content, Comput. Intell. Mag. IEEE 10 (2) (2015) 30–41, doi:10.1109/MCI.2015.2405317.
[2] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, Speeded-up robust features (surf), Comput. Vis. Image Underst. 110 (3) (2008) 346–359.
[3] M. Bazarganigilani, Optimized image feature selection using pairwise classifiers, J. Artif. Intell. Soft Comput. Res. 1 (2) (2011) 147–153.
[4] G. Bradski, The opencv library, Doctor Dobbs J. 25 (11) (2000) 120–126.
[5] T. Chang, C.-C. Kuo, Texture analysis and classification with tree-structured wavelet transform, IEEE Trans. Image Process. 2 (4) (1993) 429–441, doi:10.1109/83.242353.
[6] Y. Chang, Y. Wang, C. Chen, K. Ricanek, Improved image-based automatic gender classification by feature selection, J. Artif. Intell. Soft Comput. Res. 1 (3) (2011) 241–253.
[7] J.A. Daniel Carlos Guimaraes Pedronette, R. da S. Torres, A scalable re-ranking method for content-based image retrieval, Inf. Sci. 265 (0) (2014) 91–104. http://dx.doi.org/10.1016/j.ins.2013.12.030.
[8] M. Datar, N. Immorlica, P. Indyk, V.S. Mirrokni, Locality-sensitive hashing scheme based on p-stable distributions, in: Proceedings of the 20th Annual Symposium on Computational Geometry, in: SCG '04, ACM, New York, NY, USA, 2004, pp. 253–262.
[9] P. Drozda, K. Sopyla, P. Górecki, Online crowdsource system supporting ground truth datasets creation, in: Proceedings of the 12th International Conference Artificial Intelligence and Soft Computing, ICAISC 2013, Zakopane, Poland, June 9–13, 2013, Part I, 2013, pp. 532–539, doi:10.1007/978-3-642-38658-9_48.
[10] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman, The pascal visual object classes (voc) challenge, Int. J. Comput. Vision 88 (2) (2010) 303–338.
[11] J. Francos, A. Meiri, B. Porat, A unified texture model based on a 2-d wold-like decomposition, IEEE Trans. Signal Process. 41 (8) (1993) 2665–2678, doi:10.1109/78.229897.
[12] K. Grauman, T. Darrell, Efficient image matching with distributions of local invariant features, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005, vol. 2, 2005, pp. 627–634, doi:10.1109/CVPR.2005.138.
[13] J. Huang, S. Kumar, M. Mitra, W.-J. Zhu, R. Zabih, Image indexing using color correlograms, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1997, pp. 762–768, doi:10.1109/CVPR.1997.609412.
[14] H.V. Jagadish, A retrieval technique for similar shapes, SIGMOD Rec. 20 (2) (1991) 208–217.
[15] A.K. Jain, F. Farrokhnia, Unsupervised texture segmentation using gabor filters, Patt. Recognit. 24 (12) (1991) 1167–1186.
[16] H. Jégou, M. Douze, C. Schmid, P. Pérez, Aggregating local descriptors into a compact image representation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 3304–3311.
[17] H. Jégou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, C. Schmid, Aggregating local image descriptors into compact codes, IEEE Trans. Patt. Anal. Mach. Intell. 34 (9) (2012) 1704–1716.
[18] T. Kanimozhi, K. Latha, An integrated approach to region based image retrieval using firefly algorithm and support vector machine, Neurocomputing 151, Part 3 (0) (2015) 1099–1111.
[19] E. Karakasis, A. Amanatiadis, A. Gasteratos, S. Chatzichristofis, Image moment invariants as local features for content based image retrieval using the bag-of-visual-words model, Patt. Recognit. Lett. 55 (0) (2015) 22–27.
[20] B. Karimi, A. Krzyzak, A novel approach for automatic detection and classification of suspicious lesions in breast ultrasound images, J. Artif. Intell. Soft Comput. Res. 3 (4) (2013) 265–276.
[21] H. Kauppinen, T. Seppanen, M. Pietikainen, An experimental comparison of autoregressive and fourier-based descriptors in 2d shape classification, IEEE Trans. Patt. Anal. Mach. Intell. 17 (2) (1995) 201–207, doi:10.1109/34.368168.
[22] S. Kiranyaz, M. Birinci, M. Gabbouj, Perceptual color descriptor based on spatial distribution: a top-down approach, Image Vision Comput. 28 (8) (2010) 1309–1326.
[23] C.-H. Lin, H.-Y. Chen, Y.-S. Wu, Study of image retrieval and classification based on adaptive features using genetic algorithm feature selection, Expert Syst. Appl. 41 (15) (2014) 6611–6621.
[24] G.-H. Liu, J.-Y. Yang, Content-based image retrieval using color difference histogram, Patt. Recognit. 46 (1) (2013) 188–198.
[25] L. Liu, L. Shao, X. Li, Evolutionary compact embedding for large-scale image classification, Inf. Sci. (0) (2014). http://dx.doi.org/10.1016/j.ins.2014.06.030.
[26] S. Liu, X. Bai, Discriminative features for image classification and retrieval, Patt. Recognit. Lett. 33 (6) (2012) 744–751.
[27] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vision 60 (2) (2004) 91–110.
[28] J. Matas, O. Chum, M. Urban, T. Pajdla, Robust wide-baseline stereo from maximally stable extremal regions, Image Vision Comput. 22 (10) (2004) 761–767. British Machine Vision Computing 2002.
[29] K. Mikolajczyk, C. Schmid, Scale and affine invariant interest point detectors, Int. J. Comput. Vision 60 (1) (2004) 63–86.
[30] D. Nister, H. Stewenius, Scalable recognition with a vocabulary tree, in: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2, in: CVPR '06, IEEE Computer Society, Washington, DC, USA, 2006, pp. 2161–2168.

[31] M.R. Ogiela, R. Tadeusiewicz, Syntactic reasoning and pattern recognition for analysis of coronary artery images, Artif. Intell. Med. 26 (1) (2002) 145–159.

[32] M.R. Ogiela, R. Tadeusiewicz, Nonlinear processing and semantic content analysis in medical imaging-a cognitive approach, IEEE Trans. Instrum. Meas. 54 (6) (2005) 2149–2155.

[33] G. Pass, R. Zabih, Histogram refinement for content-based image retrieval, in: Proceedings of the 3rd IEEE Workshop on Applications of Computer Vision, 1996. WACV '96, 1996, pp. 96–102, doi:10.1109/ACV.1996.572008.

[34] J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, Object retrieval with large vocabularies and fast spatial matching, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR '07, 2007, pp. 1–8.

[35] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, A simultaneous feature adaptation and feature selection method for content-based image retrieval systems, Knowledge-Based Syst. 39 (0) (2013) 85–94.

[36] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, Orb: An efficient alternative to sift or surf, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2011, pp. 2564–2571, doi:10.1109/ICCV.2011.6126544.

[37] D. Rutkowska, R. Nowicki, New neuro-fuzzy architectures, Proceedings of the International Conference Artificial and Computational Intelligence for Decision, Control and Automation in Engineering and Industrial Applications, ACIDCA, 2000, pp. 82–87.

[38] L. Rutkowski, Flexible Neuro-Fuzzy Systems, Kluwer Academic Publishers, 2004.

[39] L. Rutkowski, Computational Intelligence Methods and Techniques, Springer Berlin Heidelberg, 2008.

[40] R.E. Schapire, A brief introduction to boosting, in: Proceedings of the 16th International Joint Conference on Artificial Intelligence - Volume 2, in: IJCAI'99, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999, pp. 1401–1406.

[41] R. Scherer, Designing boosting ensemble of relational fuzzy systems, Int. J. Neural Syst. 20 (5) (2010) 381–388.

[42] R. Scherer, Multiple Fuzzy Classification Systems, Springer Publishing Company, Incorporated, 2014.

[43] N. Shrivastava, V. Tyagi, Content based image retrieval based on relative locations of multiple regions of interest using selective regions matching, Inf. Sci. 259 (0) (2014) 212–224. http://dx.doi.org/10.1016/j.ins.2013.08.043.

[44] J. Sivic, A. Zisserman, Video google: a text retrieval approach to object matching in videos, in: Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on, vol. 2, 2003, pp. 1470–1477.

[45] J. Śmietański, R. Tadeusiewicz, E. Łuczyńska, Texture analysis in perfusion images of prostate cancera case study, Int. J. Appl. Math. Comput. Sci. 20 (1) (2010) 149–156.

[46] K. Tieu, P. Viola, Boosting image retrieval, Int. J. Comput. Vision 56 (1–2) (2004) 17–36.

[47] R.C. Veltkamp, M. Hagedoorn, State of the art in shape matching, in: M.S. Lew (Ed.), Principles of Visual Information Retrieval, Springer-Verlag, London, UK, 2001, pp. 87–119.

[48] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. CVPR, vol. 1, 2001, pp. I–511–I–518.

[49] S. Voloshynovskiy, M. Diephuis, D. Kostadinov, F. Farhadzadeh, T. Holotyak, On accuracy, robustness, and security of bag-of-word search systems, in: IS&T/SPIE Electronic Imaging, International Society for Optics and Photonics, 2014, p. 902807.

[50] X.-Y. Wang, H.-Y. Yang, Y.-W. Li, W.-Y. Li, J.-W. Chen, A new svm-based active feedback scheme for image retrieval, Eng. Appl. Artif. Intell. 37 (0) (2015) 43–53.

[51] J. Wu, H. Shen, Y.-D. Li, Z.-B. Xiao, M.-Y. Lu, C.-L. Wang, Learning a hybrid similarity measure for image retrieval, Patt. Recog. 46 (11) (2013) 2927–2939.

[52] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR, 2009, pp. 1794–1801, doi:10.1109/CVPR.2009.5206757.

[53] J. Yu, Z. Qin, T. Wan, X. Zhang, Feature integration analysis of bag-of-features model for image retrieval, Neurocomputing 120 (0) (2013) 355–364. Image Feature Detection and Description.

[54] J. Zhang, M. Marszalek, S. Lazebnik, C. Schmid, Local features and kernels for classification of texture and object categories: a comprehensive study, in: Proceedings of the Conference on Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06, 2006, p. 13, doi:10.1109/CVPRW.2006.121.

[55] W. Zhang, B. Yu, G. Zelinsky, D. Samaras, Object class recognition using multiple layer boosting with heterogeneous features, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005, vol. 2, 2005, pp. 323–330, doi:10.1109/CVPR.2005.251.

[56] C. Zitnick, P. Dollar, Edge boxes: locating object proposals from edges, in: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), Computer Vision ECCV 2014, Lecture Notes in Computer Science, vol. 8693, Springer International Publishing, 2014, pp. 391–405.