Efficient Visual Classification by Fuzzy Rules

Marcin Korytkowski, Rafał Scherer
Czestochowa University of Technology
Al. Armii Krajowej 36
42-200 Czestochowa, Poland
{marcin.korytkowski, rafal.scherer}@pcz.pl

Dominik Szajerman
Institute of Information Technology
Lodz University of Technology
Lodz, Poland
dominik.szajerman@p.lodz.pl

Dawid Połap, Marcin Woźniak
Institute of Mathematics
Faculty of Applied Mathematics
Silesian University of Technology
Kaszubska 23
44-100 Gliwice, Poland
{dawid.polap, marcin.wozniak}@polsl.pl

***Abstract*—The paper proposes a method for classifying and fast retrieving images which uses boosting metalearning to search for the most salient image features. We use local image keypoints as image features. We construct by boosting a set fuzzy rules describing image feature parameters. The rules constitute a set of weak classifiers voting for the final image class. The method can use various image features, engineered and learned by deep learning methods. We checked the methods on some real-world images.**

***Index Terms*—content-based image retrieval, fuzzy rules, boosting, image keypoints**

I. Introduction

In recent times, one can observe the increasing development of multimedia technologies and their rising dominance in life and business. Healthcare, and in particular medical diagnostics, is one of the areas that provide a relatively broad spectrum of possible applications for computer vision solutions. In the past, most methods focused on processing and delivery of results in the most readable form to the doctor's diagnosis for analysis. These include medical imaging, such as computed tomography, magnetic resonance, and ultrasonography, which transform signals from the device into a diagnostic readable image. Now, the diagnosis can be automatized thanks to image classification. 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. The 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 [1], [2], [3], [4], [5], [6] [7], [8], [9], [10] and classification [11], [12], [13], [14], [15], [16], [17] 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 [18], [19], [20], textures [21], [22], [23], [24], shape [25], [26], [27] or edge detectors [28]. Local invariant features have gained a wide popularity [29], [30], [31], [32], [33]. The most popular local keypoint detectors and descriptors are SURF [34], SIFT [29] or ORB [35].

In content-based image retrieval and classification, we can distinguish two approaches. The first one gradually generalises information from an image. To this group, we can include methods based on machine learning such as convolutional neural networks, e.g. [36], [37], [38] , statistical classifiers [39] or older methods based on histograms [40]. These methods try to reduce the amount of visual feature data to describe the entire image at the highest possible level. Neural networks can be trained to recognise and classify particular elements of an image, but they lose some information that is crucial to determine if the content between images is identical.

To check similarity between images we can use methods from the second popular group that is based on local interest points (keypoints), or other features that describe the local content of an image [41], [42]. Such methods do not generalise the content of an image and do not try to classify it. They usually generate significant amount of data, but they can find similar fragments of content between images. Thanks to this, this group of methods found multiple applications in video tracking and processing, for example, to correct content transition between frames during the camera move or video tamper detection [43]. Another popular application is a three-dimensional object reconstruction from a set of images. Some popular methods include SIFT, SURF, HOG, ORB, BRIEF, FREAK, with many modifications.

In the case of the first group of methods, work with a larger set of images is easier, because the result features are simple and in most cases can be easily stored and searched. But in the case of the second group, the main problem is a large and variable amount of data per image, what makes them appropriate for, e.g. two or more image stitching (for panorama purposes or image stacking). To speed up the search process, we can use methods that create keypoint structure representation or descriptors [44], [45].

In this paper we present a method for classifying and fast retrieving images (partially inspired by [46], [47], [48]) which

The project financed under the program of the Polish Minister of Science and Higher Education under the name "Regional Initiative of Excellence" in the years 2019–2022 project number 020/RID/2018/19, the amount of financing 12,000,000.00 PLN.

978-1-7281-6932-3/20/$31.00 ©2020 IEEE

uses boosting metalearning to search for the most salient image features. In [46], [47] certain feature values become weak classifiers for detecting faces. In our approach, boosting is used to select the salient image descriptors to generate fuzzy rules which use fuzzy sets to describe information [49], [50], [51]. We draw randomly one descriptor from the positive set to make a base for a new fuzzy rule (new classifier). The parameters of this rule are changed to better accommodate the rule to its class. The presented approach can use various image local features, hand-crafted (e.g. SIFT or SURF) and learned ones. The remainder of the paper is organised as follows. In Section II we present the fuzzy rule generation algorithm and query image classification. Section III compares the algorithm for two t-norms with an established image retrieval algorithm and Section IV concludes the paper.

II. Boosting-Generated Simple Fuzzy Classifiers

As we mentioned earlier, we were sparked by the work of Viola et al. [46], [47], where the authors presented two ideas: an integrated image for quick calculation features and selection of salient image features by boosting every visual class. They used very simple features (filters) similar to Haar Basis functions, and some of them are chosen by the AdaBoost as weak classifiers. We use much more advanced features, i.e., computed keypoint descriptor vectors and fuzzy rules describing the descriptor's universe of discourse. The similarity to Viola's work is in using boosting to find the most representative fuzzy rules for the visual class ω_c, $c = 1, \ldots, V$, which we use to classify images. We can use various types of image local features, and we chose the SIFT descriptors; thus classifiers have $N = 128$ features. The fuzzy rules have the following form

$$\begin{array}{c} R_t^c\text{: IF } x_1 \text{ is } G_{1,t}^c \text{ AND } x_2 \text{ is } G_{2,t}^c \text{ AND} \ldots \\ \ldots \text{ AND } x_{128} \text{ is } G_{128,t}^c \text{ THEN image } i \in \ \omega_c(\beta_t^c) \end{array}, \quad (1)$$

where $t = 1, \ldots, T^c$ is the rule number, T^c is the number of rules voting for class ω_c and β_t^c is the weak classifier weight. we apply the Gaussian membership functions

$$G_{n,t}^c(x) = e^{-\left(\frac{x - m_{n,t}^c}{\sigma_{n,t}^c}\right)^2}, \quad (2)$$

where $m_{n,t}^c$ is the center of the Gaussian function (2) and $\sigma_{n,t}^c$ is its width. We pass over the class index c as the further considerations are for one class.

The training dataset has I images (I_{pos} positive ones and I_{neg} negative ones). Initially, descriptors have the same boosting weights

$$D_1^l = \frac{1}{L} \text{ for } l = 1, \ldots, L\ , \quad (3)$$

where L is the number of descriptors for a given visual class. Two matrices are the training dataset of image descriptors

$$\mathbf{P}_t = \begin{bmatrix} \mathbf{p}^1 & D_t^1 \\ \vdots & \vdots \\ \mathbf{p}^{L_{pos}} & D_t^{L_{pos}} \end{bmatrix} = \begin{bmatrix} p_1^1, \ldots, p_N^1 & D_t^1 \\ \vdots & \vdots \\ p_1^{L_{pos}}, \ldots, p_N^{L_{pos}} & D_t^{L_{pos}} \end{bmatrix}, \quad (4)$$

$$\mathbf{N}_t = \begin{bmatrix} \mathbf{n}^1 & D^1 \\ \vdots & \vdots \\ \mathbf{n}^{L_{neg}} & D_t^{L_{neg}} \end{bmatrix} = \begin{bmatrix} n_1^1, \ldots, p_N^1 & D^1 \\ \vdots & \vdots \\ n_1^{L_{neg}}, \ldots, p_N^{L_{neg}} & D_t^{L_{neg}} \end{bmatrix}. \quad (5)$$

We train the system to obtain a set of T simple classifiers (weak learners) as fuzzy rules (1). After each run t, $t = 1, \ldots, T$, of the algorithm, we obtain rule R_t. A detailed description of the process is as follows.

1) Randomly choose one vector $\mathbf{p}^r$, $1 \le r \le L_{pos}$ from the set of positive descriptors with normalized distribution of elements $D_t^1, \ldots, D_t^{L_{pos}}$ in matrix (4). This vector becomes the set of initial parameters of a new classifier and the boosting weights contribute to the probability of choosing a keypoint.
2) The nearest descriptor to $\mathbf{p}^r$ from the positive set is added to matrix $\mathbf{M}_t$ of the size $I_p \times N$. Its each row is one descriptor from image v_i, $i = 1, \ldots, I_{pos}$, and images do not repeat

$$\mathbf{M}_t = \begin{bmatrix} \tilde{p}_{t,1}^1 & \cdots & \tilde{p}_{t,N}^1 \\ \vdots & \cdots & \vdots \\ \tilde{p}_{t,1}^j & \ddots & \tilde{j}_{t,N}^j \\ \vdots & \cdots & \vdots \\ \tilde{p}_{t,1}^{I_{pos}} & \cdots & \tilde{p}_{t,N}^{I_{pos}} \end{bmatrix}, \quad (6)$$

Each vector $\begin{bmatrix} \tilde{p}_{t,1}^j & \cdots & \tilde{p}_{t,N}^j \end{bmatrix}$, $j = 1, \ldots, I_{pos}$, in matrix (6) is one descriptor from $\{\mathbf{p}^i; i = 1, .., L_{pos}\}$.
3) Here we look for the fuzzy rules parameters (1).
   a) We determine absolute value $d_{t,n}$ as the difference between the smallest and the highest values in each column of the matrix (6)

$$d_{t,n} = |\min_{i=1,\ldots,I_p} p_n^i - \max_{i=1,\ldots,I_p} p_n^i| \quad (7)$$

where $n = 1, \ldots, N$. Then, we calculate the center of fuzzy Gaussian membership function (2) $m_{t,n}$

$$m_{t,n} = \max_{i=1,\ldots,I_p} p_n^i - \frac{d_{t,n}}{2}\ . \quad (8)$$

To compute the widths the fuzzy set membership functions we 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 values satisfy $G_{n,t}(x) \ge 0.5$. 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 $\sigma_{t,n}$

$$\sigma_{t,n} = \frac{d_{t,n}}{2\sqrt{-\ln(0.5)}} \quad (9)$$

We calculate values $m_{t,n}$ and $\sigma_{n,t}$ for every element of the nth column of matrix (6); thus we repeat the above steps for all N dimensions. In this way, we obtain N Gaussian membership functions of N fuzzy sets, labeled by $G_{n,t}$, where n,

$n = 1,..,N$, is the index associated with feature vector elements and t is the fuzzy rule number.

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

4) We calculate the quality of the classifier (like in the AdaBoost algorithm [52]). We compute the activation level of rule R_t by a t-norm of all fuzzy sets membership function values

$$f_t(\bar{\mathbf{x}}) = \mathop{T}_{n=1}^{N} G_{n,t}(\bar{x}_n) \ , \tag{10}$$

where $\bar{\mathbf{x}} = [\bar{x}_1, \ldots, \bar{x}_N]$ is a vector of the values of linguistic variables $x_1, \ldots, x_N$. Generally, the intersection of fuzzy sets is defined as

$$\mu_{A\cap B}\left(x\right) = T\left(\mu_A\left(x\right), \mu_B\left(x\right)\right), \tag{11}$$

where the function *T* is the so-called *t*-norm. Therefore, $\min(\mu_A\left(x\right), \mu_B\left(x\right)) = T\left(\mu_A\left(x\right), \mu_B\left(x\right)\right)$ is an example of operation of the *t*-norm. Similarly, the union of fuzzy sets is defined as follows:

$$\mu_{A\cup B}\left(x\right) = S\left(\mu_A\left(x\right), \mu_B\left(x\right)\right), \tag{12}$$

where the function S is *t*-conorm. In this case, $\max(\mu_A\left(x\right), \mu_B\left(x\right)) = S\left(\mu_A\left(x\right), \mu_B\left(x\right)\right)$ is an example of the *t*-conorm. It is worth noting that the *t*-norms and the *t*-conorms belong to the so-called triangular norms. Below formal definitions will be presented.

Definition 1. The function of two variables T

$$T : [0,1] \times [0,1] \to [0,1] \tag{13}$$

is called a t-norm, if (i) function T is nondecreasing with relation to both arguments

$$T\left(a,c\right) \leq T\left(b,d\right) \quad \text{for} \quad a \leq b, \ c \leq d \tag{14}$$

(ii) function T satisfies the condition of commutativity

$$T\left(a,b\right) = T\left(b,a\right) \tag{15}$$

(iii) function T satisfies the condition of associativity

$$T\left(T\left(a,b\right),c\right) = T\left(a, T\left(b,c\right)\right) \tag{16}$$

(iv) function T satisfies the boundary condition

$$T\left(a,1\right) = a, \tag{17}$$

where $a,b,c,d \in [0,1]$. From the assumptions it follows that

$$T\left(a,0\right) = T\left(0,a\right) \leq T\left(0,1\right) = 0. \tag{18}$$

Therefore, the second boundary condition takes the form

$$T\left(a,0\right) = 0. \tag{19}$$

Using property (16), the definition of *t*-norm may be generalized for the case of a *t*-norm of multiple variables

$$\mathop{T}_{i=1}^{n}\left\{a_i\right\} = T\left\{\mathop{T}_{i=1}^{n-1}\left\{a_i\right\}, a_n\right\} = \tag{20}$$

$$= T\left\{a_1, a_2, ..., a_n\right\} = \tag{21}$$

$$= T\left\{\mathbf{a}\right\} = a_1 \overset{T}{*} a_2 \overset{T}{*} \ldots \overset{T}{*} a_n.$$

The most popular triangular norms are the minimum and product t-norms, described by the following formulas

$$T_M\left\{a_1, a_2\right\} = \min\left\{a_1, a_2\right\}, \tag{22}$$

$$T_P\left\{a_1, a_2\right\} = a_1 \cdot a_2, \tag{23}$$

$$T_M\left\{a_1, a_2, ..., a_n\right\} = \min_{i=1,\ldots,n}\left\{a_i\right\}, \tag{24}$$

$$T_P\left\{a_1, a_2, ..., a_n\right\} = \prod_{i=1,\ldots,n}\left\{a_i\right\}. \tag{25}$$

For example, in the case of the minimum t-norm, formula (10) has the following form

$$f_t(\bar{\mathbf{x}}) = \min_{n=1}^{N} G_{n,t}(\bar{x}_n) \ . \tag{26}$$

The current boosting run is for class ω_c. This is a binary classification, that is $y^l = 1$ for positive images, and $y^l = 0$ for other images. Thus, we calculate the prediction by

$$h_t(\bar{\mathbf{x}}^l) = \begin{cases} 1 & \text{if} \quad f_t(\bar{\mathbf{x}}^l) \geq \frac{1}{2} \\ 0 & \text{otherwise} \end{cases} \ . \tag{27}$$

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 (27) for all $L = L_{pos} + L_{neg}$ descriptors of all positive and negative images

$$\varepsilon_t = \sum_{l=1}^{L} D_t^l I(h_t(\bar{\mathbf{x}}^l) \neq y^l) \ , \tag{28}$$

where I is the indicator function

$$I(a \neq b) = \begin{cases} 1 & \text{if} \quad a \neq b \\ 0 & \text{if} \quad a = b \end{cases} \ . \tag{29}$$

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

$$\alpha_t = 0.5 \ln \frac{1 - \varepsilon_t}{\varepsilon_t} \ . \tag{30}$$

$$D_{t+1}^l = \frac{D_t^l \exp\{-\alpha_t I(h_t(\bar{\mathbf{x}}^l) = y^l)\}}{C} \ , \tag{31}$$

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

$$\beta_t = \frac{\alpha_t}{\sum_{t=1}^{T} \alpha_t} \ . \tag{32}$$

We use the obtained set of rules R for the query image classification. We have to generate the rules for every class of images ω_c, $c = 1, \ldots, V$ to obtain finally a set of V strong classifiers. For a new query image, we have to generate *u* descriptors in $\mathbf{Q}$

$$\mathbf{Q} = \begin{bmatrix} \mathbf{q}^1 \\ \mathbf{q}^2 \\ \vdots \\ \mathbf{q}^u \end{bmatrix} = \begin{bmatrix} q_1^1 \cdots q_N^1 \\ q_1^2 \cdots q_N^2 \\ \vdots \\ q_1^u \cdots q_N^u \end{bmatrix} \ . \tag{33}$$

To classify the query image we have to compute

$$F_t(\mathbf{Q}) = \mathop{S}_{j=1}^{u} \left(\mathop{T}_{n=1}^{N} G_{n,t}(q_n^j) \right) , \tag{34}$$

where S and T are t-norm and t-conorm, respectively. To compute the overall output of the ensemble of classifiers, for each class ω_c we sum weak classifiers outputs (34) taking into consideration their importance (32), i.e.

$$H^c(\mathbf{Q}) = \sum_{t=1}^{T^c} \beta_t F_t(\mathbf{Q}) . \tag{35}$$

We also assign a class label to the query image in the following way

$$f(\mathbf{Q}) = \arg \max_{c=1,\ldots,V} H^{*c}(\mathbf{Q}) . \tag{36}$$

In formulas (35) and (36) we retrne with class label index c removed earlier. We show example fuzzy rules created during the boosting learning in Figure 1.

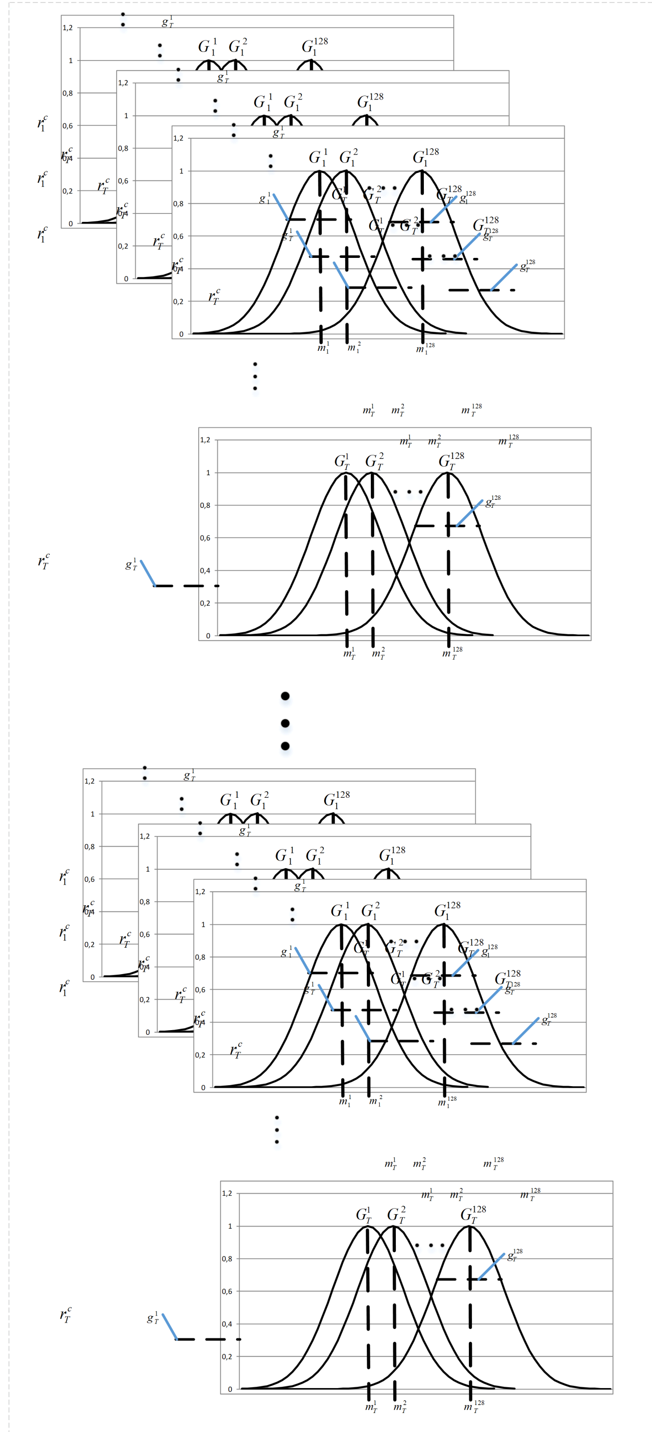

Fig. 1. Example antecedents of fuzzy rules for image classification created by boosting.

III. Experiments

We evaluated the presented approach on images taken from the PASCAL Visual Object Classes (VOC) dataset [53] by checking the speed and accuracy. We present some examples in Fig. 2. We divided each class of objects into training and testing examples (15 %). We generated local keypoint descriptors with the SIFT algorithm; for complex images there would be even thousands of descriptors. We used negative images from a different kind of images from the dataset. We checked the proposed method performance against the Support Vector Machine (SVM) [54] with the Chi-Square kernel. The training procedure described in Section II requires a set of negative examples for each considered class of objects. We picked randomly negative examples from other classes. We ran it with a dictionary of the size of 400 words. We created dictionaries for BoF in C++ language, based on the OpenCV Library [55]. Both methods were evaluated with the same images (Table I). In the BoF algorithm the column "Training time" is empty as the training is performed for the whole dataset. As we can see, the algorithm presented in the paper is faster and more accurate than the BoF approach. Moreover, the product t-norm performs better than the minimum one.

IV. Conclusions

The paper presented the method for fast content-based image classification by fuzzy rules. The fuzzy rules describe image feature parameters; in our it is the SIFT algorithm, but almost any local features can be used. The rules are created by the AdaBoost algorithm which picks the most important features for a given visual class. The rules then work as a classification ensemble of weak classifiers. The proposed approach outperformed the state-of-the-art method in image retrieval, which is a combination of the bag of features method with SVM. Our approach is faster and more accurate. Moreover, contrary to the bag-of-features approach, it is relatively simple to train the system to recognize new image classes. In our experiments, the product t-norm performed slightly better than the minimum one. We used the SIFT image features, but the proposed method can use other image keypoint detectors and descriptors, hand-crafted as SURF or ORB and learned ones as LIFT [56] or that proposed in [57].

TABLE I
COMPARISON OF THE PROPOSED METHOD FOR TWO T-NORMS WITH THE BAG OF WORDS COMBINED WITH THE SUPPORT VECTOR MACHINES.

	Proposed approach				Bag of features and SVM		
	Classification accuracy on testing set (prod. t-norm)	Classification accuracy on testing set (min t-norm)	Training time [s]	Testing time [s]	Classification accuracy on testing set	Training time [s]	Testing time [s]
bicycle	81.45%	81.12%		2.236	69.54%		7.141
boat	75.52%	75.03%		2.435	66.84%		6.274
bus	82.35%	80.45%		3.023	70.89%		5.241
car	76.33%	75.47%		3.274	88.45%		7.274
cat	76.47%	75.54%		3.137	88.72%		5.134
plane	72.29%	72.45%		3.272	80.45%		6.233
train	64.71%	63.63%		3.458	54.34%		5.381
Total	75.59%	74.81%	287.381	20.925	74.17%	544.323	42.678

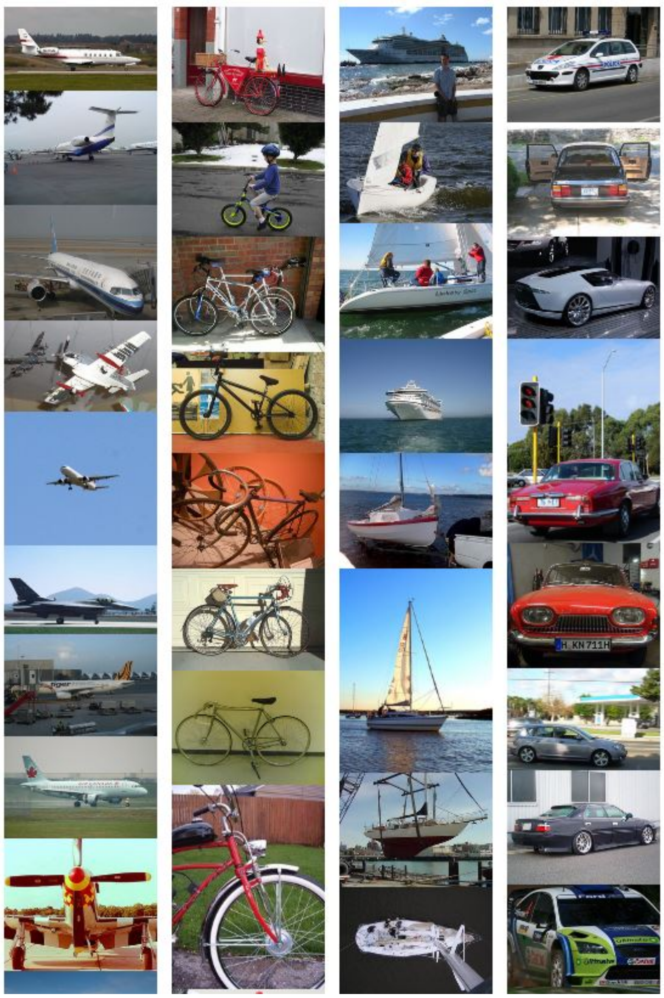

Fig. 2. Examples of images from the PASCAL Visual Object Classes (VOC) dataset, namely aeroplanes, bicycles, boats and cars.

REFERENCES

[1] J. A. Daniel Carlos, Guimaraes Pedronette and R. da S. Torres, "A scalable re-ranking method for content-based image retrieval," *Information Sciences*, vol. 265, no. 0, pp. 91 – 104, 2014.

[2] T. Kanimozhi and K. Latha, "An integrated approach to region based image retrieval using firefly algorithm and support vector machine," *Neurocomputing*, vol. 151, Part 3, no. 0, pp. 1099 – 1111, 2015.

[3] E. Karakasis, A. Amanatiadis, A. Gasteratos, and S. Chatzichristofis, "Image moment invariants as local features for content based image retrieval using the bag-of-visual-words model," *Pattern Recognition Letters*, vol. 55, no. 0, pp. 22 – 27, 2015.

[4] C.-H. Lin, H.-Y. Chen, and Y.-S. Wu, "Study of image retrieval and classification based on adaptive features using genetic algorithm feature selection," *Expert Systems with Applications*, vol. 41, no. 15, pp. 6611 – 6621, 2014.

[5] G.-H. Liu and J.-Y. Yang, "Content-based image retrieval using color difference histogram," *Pattern Recognition*, vol. 46, no. 1, pp. 188 – 198, 2013.

[6] S. Liu and X. Bai, "Discriminative features for image classification and retrieval," *Pattern Recognition Letters*, vol. 33, no. 6, pp. 744 – 751, 2012.

[7] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "A simultaneous feature adaptation and feature selection method for content-based image retrieval systems," *Knowledge-Based Systems*, vol. 39, no. 0, pp. 85 – 94, 2013.

[8] X.-Y. Wang, H.-Y. Yang, Y.-W. Li, W.-Y. Li, and J.-W. Chen, "A new svm-based active feedback scheme for image retrieval," *Engineering Applications of Artificial Intelligence*, vol. 37, no. 0, pp. 43 – 53, 2015.

[9] J. Wu, H. Shen, Y.-D. Li, Z.-B. Xiao, M.-Y. Lu, and C.-L. Wang, "Learning a hybrid similarity measure for image retrieval," *Pattern Recognition*, vol. 46, no. 11, pp. 2927 – 2939, 2013.

[10] J. Yu, Z. Qin, T. Wan, and X. Zhang, "Feature integration analysis of bag-of-features model for image retrieval," *Neurocomputing*, vol. 120, no. 0, pp. 355 – 364, 2013, image Feature Detection and Description.

[11] A. Akusok, Y. Miche, J. Karhunen, K.-M. Bjork, R. Nian, and A. Lendasse, "Arbitrary category classification of websites based on image content," *Computational Intelligence Magazine, IEEE*, vol. 10, no. 2, pp. 30–41, May 2015.

[12] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 3304–3311.

[13] H. Jégou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid, "Aggregating local image descriptors into compact codes," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 9, pp. 1704–1716, 2012.

[14] B. Karimi and A. Krzyzak, "A novel approach for automatic detection and classification of suspicious lesions in breast ultrasound images," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 3, no. 4, pp. 265–276, 2013.

[15] L. Liu, L. Shao, and X. Li, "Evolutionary compact embedding for large-scale image classification," *Information Sciences*, no. 0, pp. –, 2014.

[16] N. Shrivastava and V. Tyagi, "Content based image retrieval based on relative locations of multiple regions of interest using selective regions matching," *Information Sciences*, vol. 259, no. 0, pp. 212 – 224, 2014.

[17] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, June 2009, pp. 1794–1801.

[18] J. Huang, S. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih, "Image indexing using color correlograms," in *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, Jun 1997, pp. 762–768.

[19] S. Kiranyaz, M. Birinci, and M. Gabbouj, "Perceptual color descriptor based on spatial distribution: A top-down approach," *Image Vision Comput.*, vol. 28, no. 8, pp. 1309–1326, Aug. 2010.

[20] G. Pass and R. Zabih, "Histogram refinement for content-based image retrieval," in *Applications of Computer Vision, 1996. WACV '96., Proceedings 3rd IEEE Workshop on*, Dec 1996, pp. 96–102.

[21] T. Chang and C.-C. Kuo, "Texture analysis and classification with tree-structured wavelet transform," *Image Processing, IEEE Transactions on*, vol. 2, no. 4, pp. 429–441, Oct 1993.

[22] J. Francos, A. Meiri, and B. Porat, "A unified texture model based on a 2-d wold-like decomposition," *Signal Processing, IEEE Transactions on*, vol. 41, no. 8, pp. 2665–2678, Aug 1993.

[23] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using gabor filters," *Pattern Recognition*, vol. 24, no. 12, pp. 1167 – 1186, 1991.

[24] J. Śmietański, R. Tadeusiewicz, and E. Łuczyńska, "Texture analysis in perfusion images of prostate cancer—a case study," *International Journal of Applied Mathematics and Computer Science*, vol. 20, no. 1, pp. 149–156, 2010.

[25] H. V. Jagadish, "A retrieval technique for similar shapes," *SIGMOD Rec.*, vol. 20, no. 2, pp. 208–217, Apr. 1991.

[26] H. Kauppinen, T. Seppanen, and M. Pietikainen, "An experimental comparison of autoregressive and fourier-based descriptors in 2d shape classification," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 17, no. 2, pp. 201–207, Feb 1995.

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

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

[29] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[30] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and Vision Computing*, vol. 22, no. 10, pp. 761 – 767, 2004, british Machine Vision Computing 2002.

[31] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," *International Journal of Computer Vision*, vol. 60, no. 1, pp. 63–86, 2004.

[32] D. Nister and 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*, ser. CVPR '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 2161–2168.

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

[34] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, Jun. 2008.

[35] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, Nov 2011, pp. 2564–2571.

[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[37] O. Chang, P. Constante, A. Gordon, and M. Singana, "A novel deep neural network that uses space-time features for tracking and recognizing a moving object," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 7, no. 2, pp. 125–136, 2017.

[38] R. Kumar, E. Weill, F. Aghdasi, and P. Sriram, "A strong and efficient baseline for vehicle re-identification using deep triplet embedding," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 10, no. 1, pp. 27–45, 2020.

[39] E. Rafajłowicz, H. Pawlak-Kruczek, and W. Rafajłowicz, "Statistical classifier with ordered decisions as an image based controller with application to gas burners," in *International Conference on Artificial Intelligence and Soft Computing*. Springer, 2014, pp. 586–597.

[40] G. Pass, R. Zabih, and J. Miller, "Comparing images using color coherence vectors," in *Proceedings of the fourth ACM international conference on Multimedia*. ACM, 1997, pp. 65–73.

[41] E. Rafajłowicz, M. Wnuk, and W. Rafajłowicz, "Local detection of defects from image sequences," *International Journal of Applied Mathematics and Computer Science*, vol. 18, no. 4, pp. 581–592, 2008.

[42] E. Rafajłowicz and W. Rafajłowicz, "Testing (non-) linearity of distributed-parameter systems from a video sequence," *Asian Journal of Control*, vol. 12, no. 2, pp. 146–158, 2010.

[43] W. Wei, X. Fan, H. Song, and H. Wang, "Video tamper detection based on multi-scale mutual information," *Multimedia Tools and Applications*, vol. 78, no. 19, pp. 27 109–27 126, 2019.

[44] M. Grabner, H. Grabner, and H. Bischof, "Learning features for tracking," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–8.

[45] M. Calonder, V. Lepetit, and P. Fua, "Keypoint signatures for fast learning and recognition," in *European Conference on Computer Vision*. Springer, 2008, pp. 58–71.

[46] K. Tieu and P. Viola, "Boosting image retrieval," *Int. J. Comput. Vision*, vol. 56, no. 1-2, pp. 17–36, Jan. 2004.

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

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

[49] K. Łapa, K. Cpałka, and L. Wang, "New method for design of fuzzy systems for nonlinear modelling using different criteria of interpretability," in *Artificial Intelligence and Soft Computing*, L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. A. Zadeh, and J. M. Zurada, Eds. Cham: Springer International Publishing, 2014, pp. 217–232.

[50] R. Scherer, "Designing boosting ensemble of relational fuzzy systems," *International Journal of Neural Systems*, vol. 20, no. 5, pp. 381–388, 2010.

[51] ——, *Multiple Fuzzy Classification Systems*. Springer Publishing Company, Incorporated, 2014.

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

[53] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[54] R. K. Nowicki, K. Grzanek, and Y. Hayashi, "Rough support vector machine for classification with interval and incomplete data," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 10, no. 1, pp. 47–56, 2020.

[55] G. Bradski, "The opencv library," *Doctor Dobbs Journal*, vol. 25, no. 11, pp. 120–126, 2000.

[56] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "Lift: Learned invariant feature transform," in *European Conference on Computer Vision*. Springer, 2016, pp. 467–483.

[57] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, "Discriminative learning of deep convolutional feature point descriptors," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 118–126.