Efficient Visual Classification by Fuzzy Rules

Marcin Korytkowski, Rafal 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*  
*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...
uses boosting metalearning to search for the most salient image features. In [46] 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 $\omega_c$, $c = 1, \ldots, V$, 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

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

(1)

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

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

(2)

where $n^c_{n,t}$ is the center of the Gaussian function (2) and $\sigma^c_{n,t}$ 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

$$D^1_l = \frac{1}{L} \text{ for } l = 1, \ldots, L ,$$

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

$$P_t = \begin{bmatrix} p^1 & D^1_t \\ \vdots & \vdots \\ p_{I_{pos}} & D_{I_{pos}}^t \end{bmatrix} = \begin{bmatrix} p^1 \ldots, p_N & D^1_t \\ \vdots & \vdots \\ p_{I_{pos}} \ldots, p_{I_{pos}} & D_{I_{pos}}^t \end{bmatrix},$$

(4)

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 $R^c_t$. A detailed description of the process is as follows.

1) Randomly choose one vector $p^r$, $1 \leq r \leq L_{pos}$, from the set of positive descriptors with normalized distribution of elements $D^1_t, \ldots, D^t_{I_{pos}}$ 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 $p^r$ from the positive set is added to matrix $M_t$ 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

$$M_t = \begin{bmatrix} \tilde{p}_{i,1}^1 & \cdots & \tilde{p}_{i,I}^1 \\ \vdots & \ddots & \vdots \\ \tilde{p}_{i,1}^{I_{pos}} & \cdots & \tilde{p}_{i,I}^{I_{pos}} \end{bmatrix},$$

(6)

Each vector $[\tilde{p}_{i,1}^j \cdots, \tilde{p}_{i,I}^j]$; $j = 1, \ldots, I_{pos}$, in matrix (6) is one descriptor from $\{p^i; i = 1, \ldots, L_{pos}\}$.

3) Here we look for the fuzzy rules parameters (1).

a) We determine absolute value differences between the smallest and the highest values in each column of the matrix (6)

$$d_{t,n} = |\min_{i=1,\ldots,I_p} p^i_n - \max_{i=1,\ldots,I_p} p^i_n|$$

(7)

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

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

(8)

To compute the widths the fuzzy set membership functions we assume that for all real arguments in the range of $[m_{t,n} - \frac{d_{t,n}}{2}; m_{t,n} + \frac{d_{t,n}}{2}]$, the Gaussian function values 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

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

(9)

We calculate values $m_{t,n}$ and $\sigma_{n,t}$ for every element of the $n$th 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 $n$,
\[ n = 1, \ldots, N, \] 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 \( R_i \) by a t-norm of all fuzzy sets membership function values

\[
f_t(\bar{x}) = \frac{N}{T} \sum_{n=1}^N G_{n,t}(\bar{x}_n),
\]

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

\[
\mu_{A \cap B}(x) = T(\mu_A(x), \mu_B(x)),
\]

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

\[
\mu_{A \cup B}(x) = S(\mu_A(x), \mu_B(x)),
\]

where the function \( S \) is t-conorm. In this case, \( \max(\mu_A(x), \mu_B(x)) = S(\mu_A(x), \mu_B(x)) \) 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 \( T \)

\[
T : [0,1] \times [0,1] \rightarrow [0,1]
\]

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

\[
T(a, c) \leq T(b, d) \quad \text{for} \quad a \leq b, \ c \leq d
\]

(ii) function \( T \) satisfies the condition of commutativity

\[
T(a, b) = T(b, a)
\]

(iii) function \( T \) satisfies the condition of associativity

\[
T(T(a, b), c) = T(a, T(b, c))
\]

(iv) function \( T \) satisfies the boundary condition

\[
T(a, 1) = a
\]

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

\[
T(a, 0) = T(0, a) \leq T(0, 1) = 0.
\]

Therefore, the second boundary condition takes the form

\[
T(a, 0) = 0.
\]

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

\[
\sum_{i=1}^n \{a_i\} = T\left(\sum_{i=1}^{n-1} \{a_i\}, a_n\right) = T\{a_1, a_2, \ldots, a_n\} = T\{a\} = a_1 \ast a_2 \ast \ldots \ast a_n.
\]

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

\[
T_M\{a_1, a_2\} = \min\{a_1, a_2\},
\]

\[
T_P\{a_1, a_2\} = a_1 \cdot a_2,
\]

\[
T_M\{a_1, a_2, \ldots, a_n\} = \min_{i=1,\ldots,n} \{a_i\},
\]

\[
T_P\{a_1, a_2, \ldots, a_n\} = \prod_{i=1,\ldots,n} \{a_i\}.
\]

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

\[
f_t(\bar{x}) = \min_{n=1}^N G_{n,t}(\bar{x}_n).
\]

The current boosting run is for class \( \omega_c \). 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

\[
h_t(\bar{x}^l) = \begin{cases} 1 & \text{if } f_t(\bar{x}^l) \geq \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}
\]

For all the keypoints stored in matrices \( P_t \) and \( N_t \), we compute 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

\[
\varepsilon_t = \sum_{l=1}^L D_t^l I(h_t(\bar{x}^l) \neq y^l),
\]

where \( I \) is the indicator function

\[
I(a \neq b) = \begin{cases} 1 & \text{if } a \neq b \\ 0 & \text{if } a = b \end{cases}.
\]

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

\[
\alpha_t = 0.5 \ln \frac{1 - \varepsilon_t}{\varepsilon_t},
\]

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

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

\[
\beta_t = \frac{\alpha_t}{\sum_{l=1}^L \alpha_t}.
\]

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

\[
Q = \begin{bmatrix} q^*_1 & \cdots & q^*_u \\ q^*_2 & \cdots & q^*_u \\ \vdots & \ddots & \vdots \\ q^*_N & \cdots & q^*_N \end{bmatrix}.
\]
To classify the query image we have to compute

$$F_t(Q) = \sum_{j=1}^{u} S_j \sum_{n=1}^{N} T_n G_{n,t}(q_{jn}), \quad (34)$$

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

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

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

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

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

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.

<table>
<thead>
<tr>
<th>Object</th>
<th>Classification accuracy on testing set (prod. t-norm)</th>
<th>Proposed approach Training time [s]</th>
<th>Classification accuracy on testing set (min t-norm)</th>
<th>Bag of features and SVM Training time [s]</th>
<th>Testing time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>bicycle</td>
<td>81.45%</td>
<td>2.236</td>
<td>69.54%</td>
<td>7.141</td>
<td></td>
</tr>
<tr>
<td>boat</td>
<td>75.52%</td>
<td>2.435</td>
<td>66.84%</td>
<td>6.274</td>
<td></td>
</tr>
<tr>
<td>bus</td>
<td>82.35%</td>
<td>3.023</td>
<td>70.89%</td>
<td>5.241</td>
<td></td>
</tr>
<tr>
<td>car</td>
<td>76.33%</td>
<td>3.274</td>
<td>88.45%</td>
<td>7.274</td>
<td></td>
</tr>
<tr>
<td>cat</td>
<td>76.47%</td>
<td>3.137</td>
<td>88.72%</td>
<td>5.134</td>
<td></td>
</tr>
<tr>
<td>plane</td>
<td>72.29%</td>
<td>3.272</td>
<td>80.45%</td>
<td>6.233</td>
<td></td>
</tr>
<tr>
<td>train</td>
<td>64.71%</td>
<td>3.458</td>
<td>54.34%</td>
<td>5.381</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>75.59%</td>
<td>287.381</td>
<td>74.17%</td>
<td>42.678</td>
<td></td>
</tr>
</tbody>
</table>

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

REFERENCES


