*Journal of Universal Computer Science, vol. 24, no. 4 (2018), 357-371*
*submitted: 3/1/18, accepted: 15/4/18, appeared: 28/4/18 © J.UCS*

# The Bag-of-Words Method with Different Types of Image Features and Dictionary Analysis

**Marcin Gabryel**
(Institute of Computational Intelligence
Czestochowa University of Technology, Poland
marcin.gabryel@iisi.pcz.pl)

**Abstract:** Algorithms from the field of computer vision are widely applied in various fields including security, monitoring, automation elements, but also in multimodal human-computer interactions where they are used for face detection, body tracking and object recognition. Designing algorithms to reliably perform these tasks with limited computing resources and the ability to detect the presence of nearby people and objects in the background, changes in illumination and camera pose is a huge challenge for the field. Many of these problems use different classification methods. One of many image classification algorithms is Bag-of-Words (BoW). Originally, the classic BoW algorithm was used mainly for the natural language, so its direct application to computer vision issues may not be effective enough. The algorithm presented in this article contains a number of modifications that facilitate application of many types of characteristic features extracted from an image, image representation analysis and an adaptive clustering algorithm to create a dictionary of image features. These modifications affect classification result, which was confirmed in the experimental research.

**Keywords:** Bag-of-Words algorithm, image recognition, multi-objective optimisation, image classification
**Categories:** I.4.10, I.4.9, I.2.10

## 1 Introduction

Image classification and recognition is one of the most dynamically evolving fields using artificial intelligence methods. Traditional monitoring, security and automation systems are mainly based on human perceptiveness. The idea of extracting features from images has made it possible to save them in such a way that they can be automatically compared regardless of the scale, rotation and variety of colors. This has enabled saving to the database and quick indexation, and ultimately their use in classification methods.

There are a number of algorithms that allow images to be represented, classified and searched in a database. One of such algorithms is the Bag of Words (BoW). BoW was originally used for indexing text documents. For over ten years now the BoW has also been successfully used in computer vision, where it can also be found in the literature on the subject under the name of Bag of Features or Bag of Visual Words. In the classic Bag-of-Words algorithm, a document is represented as a histogram, whose particular values indicate the occurrence of words in a text. A single histogram is a representation of one document. Histograms can be compared with each other, which makes it easy to index and search for similar texts. Instead of natural language elements (sentences, words, letters, etc.) computer vision uses local characteristic features extracted from an image.

The BoW method makes it possible to store images in the database in a secure form. Operations performed by the algorithm only use representations. Characteristic features as such are stored as vectors. The features presented in the article (for example, characteristic points generated by the SURF algorithm) do not transfer any direct information about image fragments. This data is lost when creating a dictionary by the adaptive *k*-means algorithm.

In literature there are numerous works found in which the classification of images is presented with the use of the classic Bag-of-Words algorithm. One of the first works is [Csurka, 04], where the authors presented a system called the bag-of-keypoints. The results obtained by using two different classifiers, i.e.: Naïve Bayes and SVM were compared. Another paper [Fei-Fei, 05] uses unsupervised learning to create regions, (designated as codewords and being part of so called "themes") which are treated as image characteristic features. Codewords are learnt under a theme, which is also conducted without supervision. Another method presented in [Lazebnik, 06] creates so called "spatial pyramid" from images divided into fragments which are treated as image local features. This method is used in natural scene recognition. Its advantage involves simplicity and efficiency of performance. An idea, similar to the one presented in this article, of combining different image features needed by the BoW algorithm appeared in [Yu, 13]. In that paper the descriptors were obtained as a result of combining two different algorithms (i.e. SIFT and Local Binary Pattern or HOG – Histogram of Oriented Gradients and LBP). The authors made attempts to store image representations in one or two histograms. When developing their method, the authors confirmed in their research that it is more efficient to store various features in two histograms than combing them into one descriptor.

This article is an extension of the work presented in [Gabryel, 17]. The results proved inspiring enough to encourage our further work, which resulted in the development of several modifications to the Bag-of-Words algorithm allowing for a better adjustment of its operation to the needs of image indexing. Many applications of the BoW algorithm for searching, indexing and classifying images can be found in literature. However, these methods most often involve direct transfer of the BoW algorithm to computer vision. Still, images differ significantly from text documents. First of all, it is possible to extract more different types of characteristic features from them and then use them for indexation. Most commonly used are key points [Csurka, 04][Fei-Fei, 05][Yuan, 15], fragments of images [Lazebnik, 06][Li, 16], textures and shapes [Chang, 13][Nanni, 15][Ramesh, 15]. In the works cited, these characteristics are in no way combined. Our algorithm is designed so as to make it possible to use many different features at the same time.

The number, size and diversity of features extracted from images is so large that clustering algorithms are commonly used to reduce their number. The most commonly used clustering algorithm is the *k*-means [Li, 16][Zhao, 15]. Its disadvantage, however, results from the necessity to establish an initial number of clusters which will have to remain unchanged during further operation of the algorithm. In order to eliminate this problem, in this paper an adaptive *k*-means algorithm is used, which selects by itself the appropriate number of clusters matching itself to the number of samples. As a result of the operation of the clustering algorithm image characteristic features are assigned to specific clusters. A dictionary is created from clusters, where cluster-images links are created. Ultimately, a single image is represented by a histogram in which occurrences of clusters associated with this image are included. Similarity between images is determined by the distance between the histograms stored in the database and the query image histogram. It is assumed that images with the smallest distance between histogram values are similar to each other. In the presented innovative algorithm it is possible to use many features of an image simultaneously. A multi-criteria comparison is used to compare different types of characteristic features, which results in providing so-called Parento front with non-dominated images.

Another novelty introduced to improve classification is the optimisation of histograms so as to remove information about those clusters that have a negligible impact on the classification result or influence false classification results. Optimization is done using an evolutionary algorithm that selects a certain threshold value. If the number of occurrences of a given cluster is below this threshold value, that particular cluster is not involved in creating a histogram. As a result, the histogram only contains information about the most significant clusters.

Another modification simplifies the BoW algorithm itself at the time when classification is being carried out. Commonly used is the SVM classifier [Sivic, 03], whose task is to classify the histogram given at its input. In this approach, a majority vote is proposed instead of a classifier. As a result, this algorithm is so simplified that it can be successfully implemented directly on a relational or non-relation database, as presented in the earlier papers [Gabryel, 16] and [Gabryel, 16-2]. This solution allows for using the mechanisms of database indexation.

To sum up, the presented modified BoW algorithm differs from its classic version in that it includes the following novel elements:

1. A dictionary of characteristic features is built with the use of the clustering algorithm, which, unlike classical methods of this type, selects an appropriate number of clusters by itself.
2. At least two of its characteristics are used to represent each image. The BoW algorithm creates separate histograms for each feature.
3. A resulting set of similar images is used when using a multi-criteria comparison in the sense of Pareto optimal. The distances between different histograms describing their belonging to particular image characteristics are compared.
4. No decision making classifier is used. Instead, the majority voting method is used.
5. An additional phase of histogram analysis is applied as well as their modification. There is a threshold value below which histogram elements are removed. Removing these values improves overall classification.
6. The proposed BoW algorithm can be used both for classifying and searching images.

The paper consists of several parts. The next section presents the possibility of applying computer vision in security and privacy aspects of multimodal interfaces. Section 3 includes the algorithms used in the presented BoW method. Section 4 provides a description of the operation of the modified BoW algorithm and the subsequent section describes the practical experimental research. The final section contains a recapitulation of the whole work.

## 2 Application of computer vision algorithms to multimodal interfaces with security and privacy maintained

Computer vision finds many practical applications, one of which is the application of creating multimodal interfaces. Computer vision is one of the information sources about interactions that can be combined multimodally. Its task is to observe objects, users, their locations, expressions, gestures, etc. The use of computer vision to detect, for example, objects in the context of human-computer interaction is often referred to as Vision Based Interfaces (VBI) [Turk, 03]. The basic elements of VBI, for which it is necessary to use computer vision include:

- person-level, whole body and limb tracking,
- hand and arm gestures,
- head and face detection and tracking,
- facial expression analysis, eye tracking,
- handheld objects.

Each of these elements requires different kinds of algorithms. The latter is found interesting from the point of view of classification. Interfaces that detect and track objects other than parts of the human body can be easier to use than direct human gestures. Examples include various types of passive wands, objects with active transmitters such as LEDs, and specially colored or marked objects. Another example is a camera tracking objects in the environment, where mechanisms of image classification, recognition and search are used. In this case, the BoW algorithm presented in this article can be successfully used.

The BoW algorithm ensures a high level of security and privacy. Its main task, i. e. image search and classification does not work directly on images - files that could be stolen. Operation of the BoW is based on operating on histograms, which are treated as image representation. Histograms do not in any way allow even a part of the original image to be reproduced. Query image, which is subject to the process of classification or search, must also have its own image representation. Such storage of image data allows for transmission between the requesting system and the database. In this case transmission does not have to be additionally encrypted.

The idea of not having to store original images is used for algorithms that run on the fingerprint [Voloshynovskiy, 10] or that use hash methods [Koval, 09]. In this case, a database stores only properly prepared digital fingerprints of the examined objects. In [Beekhof, 08] this counterfeit detection solution can be put into practice. A mobile device is used to take a photo of the surface of an object. A digital fingerprint is generated from this image and sent to the server. The database compares it with the other fingerprints and decides on the authenticity or falsification of the test item. The article [Farhadzadeh, 12] presents information on the performance of content-based identification using binary fingerprints, the impact of codeword length on identification accuracy, and the probability of errors occurring.

# 3 Description of the algorithms used in the proposed Bag-of-Words approach

## 3.1 Adaptive *k*-means algorithm

The task of this algorithm is to automatically select the number of clusters depending on the number of data. The only parameter requiring initial initiation is threshold value $\tau_{\max}$. As a result of its operation the algorithm generates a set of centers in points $\mathbf{v}_j$, $j = 1, \dots, c$, where $c$ is the number of clusters selected automatically. The paper [Gabryel, 16-2] provides a detailed description of this method.

## 3.2 Differential Evolution algorithm

Differential Evolution (DE) is a most efficient and fast evolutionary algorithm which was proposed by [Storn, 97]. This method has a prerequisite of three initial parameters, i.e. the population size (*NP*) and two coefficients: one for mutation control (*F*) and the other for crossover probability (*CR*). The following steps are performed under this algorithm:

1. Initiate the algorithm:
    a. Set the initial parameter values: *NP*, *F* and *CR*.
    b. Determine the fitness function that returns the results for each individual in the population.
    c. Select the initial values of the individuals in the population.
    d. Set the max number of *generation*, and the actual generation number $t = 0$.
2. For each vector from the population:
    a. Generate the mutant vector.
    b. Crossover vectors within the population.
3. Next generation $t = t + 1$.
4. If $t \leq generation$, go to point 2.
5. The individual with the highest fitness function value from the last population is treated as an optimal solution.

## 3.3 Speeded Up Robust Features algorithm

The presented Bag-of-Words algorithm works on characteristic features of an image. Several algorithms can be used to find them. One of the most popular algorithms is the Speeded-Up Robust Features (SURF) algorithm [Bay, 06]. It is a modification of another algorithm operating on a similar basis, i.e. the SIFT algorithm (Scale-Invariant Feature Transform). However, the advantage of the SURF is its faster operation. As a result of its operation, a collection of descriptors describing the surroundings of the located key points is obtained. These points are generated independently of the scale of an image, image rotation or changes in illumination.

### 3.4 Fast Non Dominated Sort algorithm

In the Bag-of-Words method being described, each image has at least two histograms describing the occurrence of different image characteristics. Two images are considered to be similar when the distance between the histograms is minimal. However, with distances between several histograms several similarity criteria are obtained. On their basis, a set of non-dominated solutions is created, which can be presented using the following formula:

$$x \succ y \iff \exists i\ f_i(x) \leq f_i(y), \tag{1}$$

where $x$ and $y$ are Pareto-optimal solutions and $x$ dominates over $y$ if and only if the value of the objective function for $x$ is not greater than the value for the objective function for $y$. The presented Bag-of-Words algorithm uses a fast version of Pareto front calculation using the Fast Non-Dominated Sort algorithm [Deb, 02].

## 4 Modified Bag-of-Words algorithm

The proposed search and image classification algorithm has three parts: (i) the initiating part, during which images are represented in the form of histograms, (ii) the analytical part, responsible for histogram optimization and modification, and (iii) the part, which searches for similar images and classifies the query image.

The initiating part is designed to create a representation of images in the form of histograms. Each type of image characteristic feature is supposed to have one histogram.

The Bag-of-Words algorithm operates on $I_L$ set of images denoted as $\boldsymbol{I}_i$, where $i = 1, \ldots, I_L$. The set of all $C$ classes to which images belong is designated as $\Omega$ and $\Omega = \{\omega_1, \ldots, \omega_C\}$. Each $\boldsymbol{I}_i$ image belongs to a $\omega_j$ class so that $c(\boldsymbol{I}_i) = \omega_j$. Different $T$ types of image characteristic features are selected.

1. Selecting randomly $J$ images from among all $I_L$ images so that:
$$J = \sum_{i=1}^{C} L_i. \tag{2}$$
where $L_i$ is the number of randomly selected images for each $\omega_i$ class.
2. Generating image characteristic features. Find all the $T$ types of characteristic feature for all images $\boldsymbol{x}_i^t = [x_{i1}^t, \ldots, x_{iK_t}^t]$, where $i = 1, \ldots, L_t$, t=1,…,T, $L_t$ - the number of all generated characteristic features, $K_t$ – size of the vector generated in order to describe $t$ feature.
3. Creating a dictionary. Group points $\boldsymbol{x}_i^t$ with the use of the adaptive $k$-means algorithm for each type $t$ separately. Obtain group centres $\boldsymbol{w}_j^t$ of clusters $N_j$, $j = 1, \ldots, N_c^t$ and $N_c^t$ is the number of clusters produced automatically during the operation of the $k$-means algorithm for each characteristic feature $t$.
4. Creating histograms for each image:
    a. Create histograms $\boldsymbol{h}_i^t = [h_{i1}^t, \ldots, h_{iN_c}^t]$ for image $i$, $i = 1, \ldots, I_L$, $t = 1, \ldots, T$, where

$$h_{ik}^{t} = \sum_{n=1}^{L} \delta_{nk}^{t}(i), k = 1, \dots, N_c^t, \tag{3}$$

$$\delta_{nk}^{t}(i) = \begin{cases} 1 & \text{if } \|\boldsymbol{w}_k^t - \boldsymbol{x}_n^t\| \le \|\boldsymbol{w}_j^t - \boldsymbol{x}_n^t\| \text{ for } \boldsymbol{x}_n^t \in \boldsymbol{I}_i, \\ & j = 1, \dots, N_c^t, j \neq k \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

If $\boldsymbol{w}_k^t$ cluster is the closest to $\boldsymbol{x}_n^t$ vector from $\boldsymbol{I}_i$ image, then indicator $\delta_{nk}^{t}(i)$ is 1.

b. Save the image representation $\boldsymbol{I}_i$ in the form of histograms $\boldsymbol{h}_i^t$ in the database along with the label of the class $c(\boldsymbol{I}_i)$ to which it belongs.

The analytical part of the algorithm is designed to analyze the obtained histograms in terms of information about the most numerous and thus significant clusters only. Two versions of this algorithm have been proposed. Each of them consists of two stages and they differ in the number of parameters determined.

In the first step the clusters' activity of visual words $\alpha_{jk}^t$ for every class is calculated:

$$\alpha_{jk}^{t} = \sum_{\substack{i=1 \\ c(\boldsymbol{I}_i)=\omega_k}}^{I_L} \sum_{n=1}^{L} \delta_{nk}^{t}(i) \tag{5}$$

for $c(\boldsymbol{I}_i) = \omega_k,\ j = 1, \dots, N_c^t, k = 1, \dots, C, t = 1, \dots, T$. If there occurs inequality:

$$\alpha_{jk}^{t} < \theta^t \tag{6}$$

then

$$h_{ij}^{t} = 0, \tag{7}$$

where $\theta^t$ is the threshold value of clusters' activity. These calculations are performed separately for each particular *t* type of image feature. This first algorithm version concerns setting threshold $\theta^t$ for feature *t* for all images in a database.

The other version of the analytical part of the presented algorithm makes it possible to set individual thresholds $\theta_k^t$ for each classes $\omega_k$ and each feature *t* of an image separately. The way in which the algorithm operates is the same as in the first version; however, formula (6) is replaced by the following inequality.

$$\alpha_{jk}^{t} < \theta_k^t, \tag{8}$$

As before, histograms contain information about the most relevant clusters. In this case, however, each class has its own limit $\theta_k^t$.

The number of possible threshold selections $\theta^t$ or $\theta_k^t$ is rather tiresome and time consuming. However, this problem can be solved by using an evolutional algorithm for automatic selection of threshold value. The DE algorithm described in Section 3.2 has been adapted to the problem by introducing the following changes:

1. Depending on the version of the algorithm, the chromosome vector will take the threshold values $\theta^t$ or $\theta_k^t$respectively. The initial values of the vector are

initiated by random values. The length of the chromosomes are, depending on the algorithm version, $C$ or $C * N_c^t$ respectively.

2. The fitness function value is taken as the accuracy value corresponding to the efficiency of image classification.
3. For each class $L_i = 30$ images are selected for each $i = 1,\dots,C$.
4. When there are no changes to the population during 10 algorithm iterations, then another set of randomly selected images is selected.

The value of accuracy index is the index of the metrics of performance within the scope of classification problems [Olson, 08]. The experimental study, presented in the next section, shows an improvement of the presented BoW method efficiency when the analytical algorithm to the classification and search process is added.

The third part of the Bag-of-Words algorithm presented heroin aims at classifying and searching for similar images. Query image $\boldsymbol{Q}$ is given on the algorithm input. Similar to the images in the database, query image $\boldsymbol{Q}$ needs to generate *T* vectors of $\boldsymbol{x}_q^t$ characteristic features. Next are created histograms $\boldsymbol{h}_Q^t$ according to formula (3). Using the L1 metric the distances for each feature *t* are calculated individually between query image $\boldsymbol{h}_Q^t$ and database image histograms $\boldsymbol{h}_i^t$:

$$d_i^t = \sum_{k=1}^{N_C^t} \left| h_{Qk}^t \cdot m_{c(\boldsymbol{I}_i),k}^t - h_{ik}^t \right|, i = 1,\dots,L \tag{9}$$

where $m_{j,k}^t$ is a mask taking value 0 or 1 whose objective is to disable inactive clusters removed according to formula (6) in the second part of the presented BoW algorithm:

$$m_{j,k}^t = \begin{cases} 1 \text{ if } \alpha_{jk}^t \geq \theta^t \\ 0 \text{ if } \alpha_{jk}^t < \theta^t \end{cases}. \tag{10}$$

For the second version (8), mask $m_{j,k}^t$ takes the form of:

$$m_{j,k}^t = \begin{cases} 1 \text{ if } \alpha_{jk}^t \geq \theta_k^t \\ 0 \text{ if } \alpha_{jk}^t < \theta_k^t \end{cases}. \tag{11}$$

The obtained values are compared independently for a given feature *t* by means of multi-criteria comparison. The algorithm used is the Fast Non Dominated Sort algorithm described in Section 3.4. This algorithm generates multiple fronts, which include images and where distances $d_i^t$ between them and query image are not dominated. For the purpose of searching for similar images, the images on the first two fronts will be considered and marked as $\boldsymbol{I}_p$. These images can serve as a set in response to a search for images similar to query image $\boldsymbol{Q}$. However, in the case of classification, images $\boldsymbol{I}_p$ take part in the majority vote:

$$c(\boldsymbol{Q}) = \omega_{win} \Leftrightarrow \sum_{\boldsymbol{I}_i \in \boldsymbol{I}_p} c_{i,win} = \max_{j=1,\dots,C} \sum_{\boldsymbol{I}_i \in \boldsymbol{I}_p} c_{i,j}, \tag{12}$$

where $\omega_{win}$ is the class to which $\boldsymbol{Q}$ belongs and $c_{i,j}$ is defined according to formula:

$$c_{i,j} = \begin{cases} 1 \text{ if } c(I_i) = \omega_j \\ 0 \text{ if } c(I_i) \neq \omega_j \end{cases}. \tag{13}$$

## 5 Experimental study

This section provides a description of the experimental study designed to present the efficiency of the presented Bag-of-Words algorithm. The study was carried out using the Caltech 101 image database [Fei-Fei, 07] from which six sample classes were chosen ($C = 6$), i.e. airplanes, car_side, leopards, motorbikes, revolvers and wrenches. The software for the study was written in Java using elements from the OpenCV library (mainly used for generating key points with the SURF algorithm). Two image features ($T = 2$) were selected: characteristic points generated with the SURF algorithm ($t = 1$) and the histogram of the number of points for particular greyscale intensity (scale $0-64$, $t = 2$). In the first case a 64-dimensional vectors containing floating-point numbers are generated ($K_1 = 64$), and in the other case a 64-integer vector is generated ($K_2 = 64$).

The first experiment presents the operation of the analytical part of the presented Bag-of-Words in terms of its overall efficiency. In the analytical part two threshold values $\theta^1$ (for the first image features - characteristic points) and $\theta^2$ (for the gray scale histogram) are determined for all the images in the database according to formula (6). The results obtained are presented in the Tables 1–4. Each table provides results for different combinations of parameters $\theta^1$ and $\tau_{\max}$ having the same value $\theta^2$ at the same time. The next cells present the obtained accuracy values given in per cents. Analysis of the tables shows that the best values are obtained for $\theta^1$ from the range of 10-25 and for $\theta^2 = 4$ independently of value $\tau_{\max}$.

The next experiment offers comparison of the BoW algorithms: one without the analytical algorithm (the first case), one with operating the first version of the analytical algorithm (according to formulas (6) and (10) - the second case), and also one with the second version of the analytical algorithm (according to formulas (8) and (11) - the third case). The results are presented in Table 5. In the first case the accuracy values for the test and learning data are provided. In the second case, apart from the results obtained, in the additional columns are given selected threshold values $\theta^t$ for each *t* feature. In the last case the DE algorithm was used to determine the threshold values $\theta_k^t$. The obtained threshold values are presented in Table 5 in separate columns for each class *k* and each type of the image quality *t*. The following parameter values were taken: $NP = 100$ and $G = 100$. Table 5 shows the accuracy values for the three cases under consideration accounting for images from both the test and the learning parts. An analysis of the results provided in the table shows that the analytical part clearly adds to improving the results obtained. The most advantageous classification results are obtained for the third case, where the analytical algorithm specifies particular values $\theta_k^t$ for each class and each image feature.

The subsequent experiments involved comparing the effectiveness of the algorithm presented in this article with the classic BoW method designed to classify images. The BoW algorithm is based on the OpenCV library, the *k*-means algorithm and the SVM classifier (classifier parameters: SVM version – C-Support Vector Classification, kernel – RBF, gamma – 0,50625, C - 312.5, maximum iteration number – 100, epsilon accuracy - 0.000001) in the C++ language. The obtained results of several runs of the algorithms for different numbers of clusters are presented in Table 6. In the following columns there is a set value of parameter $\tau_{\max}$ obtained in the process of the adaptive *k*-means algorithm, the results obtained for the test part of the set of images and the number of clusters. Similar tests for different numbers of clusters are presented in the successive columns. It is clearly seen that our new method together with the analytical algorithm is evidently more efficient than the classical Bag-of-Words algorithm.

| | | Clusters activity thresholding value $\theta^1$ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 5 | 7 | 10 | 15 | 20 | 25 | 30 | 35 |
| Value $\tau_{max}$ | 125 | 71.64 | 71.15 | 72.22 | 72.22 | 72.22 | 71.15 | 71.15 | 69.93 | 69.93 |
| | 250 | 70.12 | 70.95 | 71.15 | 70.95 | 71.93 | 70.22 | 69.93 | 66.43 | 66.43 |
| | 500 | 69.23 | 70.22 | 72.23 | 69.23 | 70.12 | 68.51 | 69.93 | 66.43 | 66.43 |
| | 750 | 66.43 | 66.43 | 64.73 | 65.73 | 66.43 | 66.43 | 65.73 | 65.73 | 64.73 |
| | 1000 | 64.82 | 65.73 | 63.38 | 67.12 | 66.43 | 65.73 | 63.38 | 63.38 | 63.38 |

*Table 1: Classification efficiency for different clusters activity thresholding $\theta^1$ in relation to value $\tau_{max}$ for $\theta^2 = 0$.*

| | | Clusters activity thresholding value $\theta^1$ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 5 | 7 | 10 | 15 | 20 | 25 | 30 | 35 |
| Value $\tau_{max}$ | 125 | 70.12 | 72.22 | 72.22 | 72.02 | 72.43 | 72.73 | 69.93 | 69.93 | 61.24 |
| | 250 | 71.64 | 72.44 | 71.15 | 73.23 | 71.53 | 73.23 | 70.93 | 69.93 | 62.14 |
| | 500 | 75.52 | 75.52 | 75.52 | 75.52 | 75.52 | **77.62** | 75.52 | 74.13 | 69.23 |
| | 750 | 64.23 | 64.23 | 64.23 | 67.23 | 68.87 | 68.87 | **69.93** | 68.87 | 67.23 |
| | 1000 | 63.06 | 64.07 | 65.56 | 67.12 | 68.53 | 65.13 | 65.13 | 64.73 | 64.73 |

*Table 2: Classification efficiency for different clusters activity thresholding $\theta^1$ in relation to value $\tau_{max}$ for $\theta^2 = 2$.*

| | | Clusters activity thresholding value $\theta^1$ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 5 | 7 | 10 | 15 | 20 | 25 | 30 | 35 |
| Value $\tau_{max}$ | 125 | 74.83 | 73.43 | 76.22 | **77.62** | 72.03 | 72.73 | 70.63 | 69.93 | 60.84 |
| | 250 | 73.64 | 74.34 | 73.43 | 76.43 | 78.53 | **79.72** | 74.93 | 69.73 | 63.24 |
| | 500 | 73.76 | 73.76 | 74.89 | 74.89 | 75.52 | 73.58 | 72.13 | 72.13 | 69.23 |
| | 750 | 65.73 | 65.73 | 65.73 | 67.02 | 68.51 | 66.43 | 66.43 | 67.13 | 67.13 |
| | 1000 | 65.13 | 65.13 | 65.83 | 67.83 | **69.93** | 67.18 | 67.18 | 65.76 | 65.76 |

*Table 3: Classification efficiency for different clusters activity thresholding $\theta^1$ in relation to value $\tau_{max}$ for $\theta^2 = 4$.*

| | | Clusters activity thresholding value $\theta^1$ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 5 | 7 | 10 | 15 | 20 | 25 | 30 | 35 |
| Value $\tau_{max}$ | 125 | 69.43 | 70.43 | 72.33 | 73.43 | 72.22 | 70.12 | 69.93 | 65.73 | 61.24 |
| | 250 | 70.22 | 70.22 | 71.93 | 72.22 | 72.22 | 71.93 | 67.23 | 67.23 | 65.21 |
| | 500 | 63.64 | 64.34 | 66.43 | 66.43 | 68.53 | 69.23 | 68.53 | 65.73 | 62.24 |
| | 750 | 65.73 | 65.73 | 65.73 | 66.43 | 66.43 | 66.43 | 65.73 | 65.73 | 64.73 |
| | 1000 | 63.38 | 64.82 | 64.82 | 66.83 | 68.53 | 66.54 | 65.23 | 64.73 | 63.38 |

*Table 4: Classification efficiency for different clusters activity thresholding $\theta^1$ in relation to value $\tau_{max}$ for $\theta^2 = 6$.*

| $\tau_{max}$ | $c$ | Algorithm without optimization | | Accuracy result with thresholding with one value | | | Thresholding values for particular classes ($\theta_k^1/\theta_k^2$) | | | | | | Accuracy result with thresholding for each of the classes | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Train | Test | $\theta^1/\theta^2$ | Train | Test | Airplanes | car_side | leopards | motorbikes | revolver | wrenches | Train | Test |
| 125 | 1176 | 73.33 | 74.13 | 10/4 | 77.77 | 76.92 | 14/2 | 11/4 | 10/2 | 23/2 | 3/2 | 10/0 | 82.22 | 77.62 |
| 250 | 300 | 72.77 | 63.63 | 20/4 | 77.77 | 69.93 | 27/2 | 14/4 | 26/4 | 50/4 | 18/2 | 21/4 | 84.44 | 79.72 |
| 500 | 255 | 77.22 | 75.52 | 20/2 | 76.66 | 77.32 | 37/4 | 4/2 | 34/2 | 50/4 | 22/4 | 15/4 | 82.77 | 77.62 |
| 750 | 104 | 63.88 | 65.73 | 35/2 | 63.88 | 69.93 | 32/2 | 15/2 | 40/4 | 10/2 | 41/0 | 22/2 | 70.55 | 69.93 |
| 1000 | 208 | 70.00 | 68.53 | 15/4 | 69.44 | 68.53 | 45/2 | 44/2 | 25/2 | 28/2 | 22/4 | 25/4 | 70.00 | 69.93 |

*Table 5: Accuracy classification efficiency values [%] obtained as a result of the experiments conducted for image classification for different values $\tau_{max}$: without the analysis algorithm, with the first version of the analysis algorithm, and the second version of the analysis algorithm.*

| $\tau_{max}$ | Proposed algorithm | | Classical BoW | |
|---|---|---|---|---|
| | $c$ | Test | Clusters | Test |
| 125 | 1176 | 77.62 | 1100 | 70.80 |
| 250 | 300 | 79.72 | 300 | 69.30 |
| 500 | 255 | 77.62 | 250 | 67.20 |
| 750 | 104 | 69.93 | 100 | 67.90 |
| 1000 | 208 | 69.93 | 200 | 59.10 |

*Table 6: Comparison of the operation of the proposed algorithm with the classical BoW.*

As already mentioned, the use of multi-criteria comparison allows us to obtain the so-called Pareto front, on which there are a number of images. The algorithm considered the first two Pareto fronts. On the basis of the images selected in this way, a decision was made about the class to which a given query image belongs by majority voting. However, the Pareto front images can be treated at the same time as a result of searching for an image similar to the query image. Several image samples from the first two Pareto fronts are shown in Figure 1. The Figure shows examples of query images from the test set (left) and sets of similar images searched from the first two Pareto-optimal fronts (right).

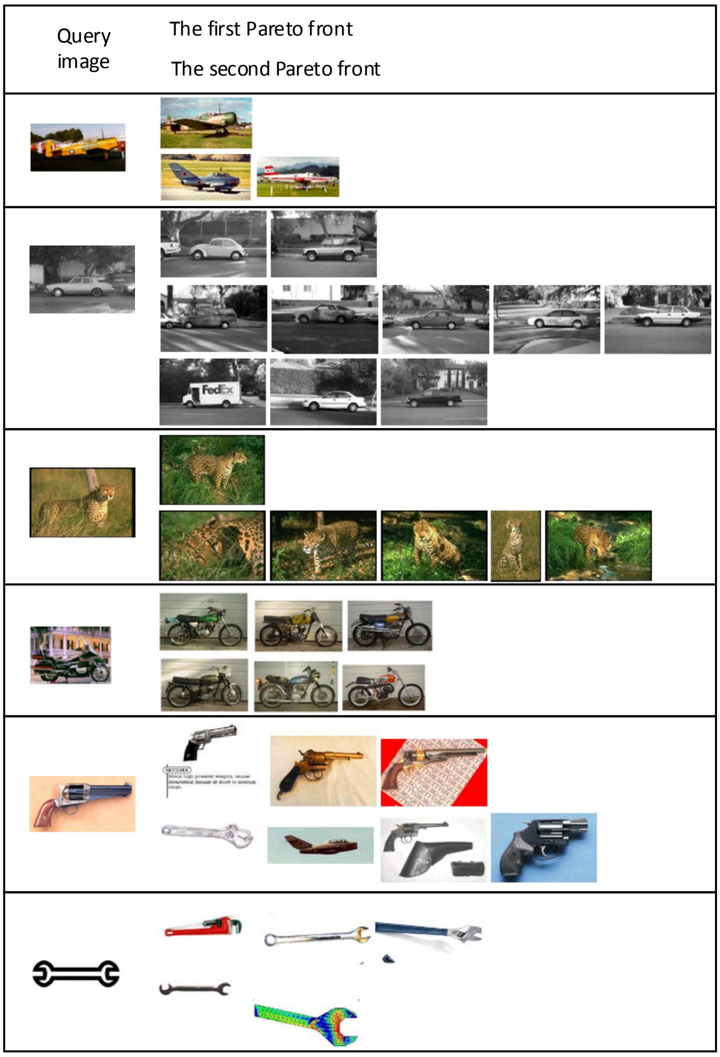


*Figure 1: Sample search results for similar images in a database.*

## 6 Conclusions

This paper presents a new image classification and search algorithm based on the Bag-of-Words algorithm. The algorithm, compared to the classic BoW, takes a much better account of the specific aspect of image representation. Among other things, it allows for multiple types of image characteristic features to be used simultaneously.

When classifying the query image, many types of features allow multi-criteria comparison, which offers a double advantage:

- Images, which belong to the first two Pareto fronts, are treated as a result of a search in a database of similar images to the query image,
- Majority voting, i. e. choosing the most numerous class of the image classes from the Pareto front, allows for a considerable simplification of calculations.

Another novelty is the use of the adaptive *k*-means algorithm. This algorithm enables automatic selection of the appropriate number of clusters. This process depends on the specificity of a particular problem being solved. This is a significant advantage in comparison to the classic *k*-means.

Because of the complexity of histograms storing the number of clusters allocated to a given image, the analytical algorithm allows for filtering them and removing those elements that have a negligible impact on the process of searching for similar histograms. The analysis process is carried out in two ways:

- setting one threshold value for each type of image feature, or
- determining thresholds values for each class and each type of image feature individually.

In both cases, the threshold values are selected by an evolutionary algorithm. The study carried out has shown that using more thresholds produces much better results compared to the BoW algorithm without an analytical algorithm.

Nowadays deep-learning neural networks algorithms [Połap, 17], widely understood parallel processing [Marszalek, 17] and other hybrid systems [Wozniak, 18] are becoming increasingly popular. These algorithms are unrivalled when it comes to image classification, but they require massive computing power. In some cases, however, the presented version of the BoW may prove much more advantageous. The algorithms applied in this method can be successfully implemented in both relational and non-relational databases.

## References

[Bay, 06] Bay H., Tuytelaars T., Van Gool L., Surf: Speeded up robust features, in: A. Leonardis, H. Bischof, A. Pinz (Eds.), Computer Vision ECCV , Vol. 3951 of Lecture Notes in Computer Science, pp. 404-417, Springer Berlin Heidelberg, 2006.

[Beekhof, 08] Beekhof, F., Voloshynovskiy, S., Koval, O., Villán, R., & Pun, T. Secure surface identification codes. In Security, Forensics, Steganography, and Watermarking of Multimedia Contents X (Vol. 6819, p. 68190D). International Society for Optics and Photonics, 2008.

[Chang, 13] Chang B.-M., Tsai H.-H., Chou W.-L., Using visual features to design a content-based image retrieval method optimized by particle swarm optimization algorithm, Engineering Applications of Artificial Intelligence 26 (10) 2372 – 2382, 2013.

[Csurka, 04] Csurka G., Dance C. R., Fan L., Willamowski,J. and Bray C. Visual categorization with bags of keypoints. In In Workshop on Statistical Learning in Computer Vision, ECCV, pages 1-22, 2004.

[Deb, 02] Deb K., Pratap A., Agarwal S., Meyarivan T., A fast and elitist multiobjective genetic algorithm: Nsga-II, Evolutionary Computation, IEEE Transactions on 6 (2) 182-197, 2002.

[Farhadzadeh, 12] Farhadzadeh, F., Voloshynovskiy, S., & Koval, O. Performance analysis of content-based identification using constrained list-based decoding. IEEE Transactions on Information Forensics and Security, 7(5), 1652-1667, 2012.

[Fei-Fei, 05] Fei-Fei, L., & Perona, P. A bayesian hierarchical model for learning natural scene categories. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on (Vol. 2, pp. 524-531). IEEE, 2005.

[Fei-Fei, 07] Fei-Fei, L., Fergus, R., & Perona, P. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. Computer vision and Image understanding, 106(1), 59-70, 2007.

[Gabryel, 17] Gabryel, M., and Capizzi, G. The bag-of-words method with dictionary analysis by evolutionary algorithm. In International Conference on Artificial Intelligence and Soft Computing (pp. 43-51). Springer, Cham, 2017.

[Gabryel, 16] Gabryel, M. The bag-of-features algorithm for practical applications using the MySQL database. In International Conference on Artificial Intelligence and Soft Computing (pp. 635-646). Springer, Cham, 2016

[Gabryel, 16-2] Gabryel, M., A bag-of-features algorithm for applications using a NoSQL database. In International Conference on Information and Software Technologies (pp. 332-343). Springer, Cham, 2016.

[Koval, 09] Koval, O., et al. On security threats for robust perceptual hashing. Media Forensics and Security. Vol. 7254. International Society for Optics and Photonics, 2009.

[Lazebnik, 06] Lazebnik S., Schmid C., and Ponce J. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, volume 2, pages 2169-2178, 2006.

[Marszalek, 17] Marszałek, Z. Parallelization of Modified Merge Sort Algorithm. Symmetry, 9.9: 176, 2017.

[Li, 16] Li W., Dong P., Xiao B., Zhou L., Object recognition based on the region of interest and optimal bag of words model, Neurocomputing 172, 271-280, 2016.

[Nanni, 15] Nanni L., Melucci M., Combination of projectors, standard texture descriptors and bag of features for classifying images, Neurocomputing, 2015.

[Olson, 08] Olson D. L., Delen D., Advanced Data Mining Techniques, 1st Edition, Springer Publishing Company, Incorporated, 2008.

[Połap, 17] Połap D., Kęsik K., Książek K., Woźniak: Obstacle Detection as a Safety Alert in Augmented Reality Models by the Use of Deep Learning Techniques. Sensors, Vol. 17, Issue. 12, MDPI, pp. 2803:1-2803:16, Switzerland, 2017.

[Ramesh, 15] Ramesh B., Xiang C., Lee T. H., Shape classification using invariant features and contextual information in the bag-of-words model, Pattern Recognition 48 (3) 894 – 906, 2015.

[Sivic, 03] Sivic J. and Zisserman A. Video google: a text retrieval approach to object matching in videos. In Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on, pages 1470-1477 vol.2, 2003.

[Storn, 97] Storn R., Price K., Differential evolution a simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization 11 (4) 341-359, 1997.

[Turk, 03] Turk, M., & Kölsch, M. Perceptual interfaces. In Communications of the ACM, 2003.

[Wozniak, 18] Wozniak, M.; Połap, D., Adaptive neuro-heuristic hybrid model for fruit peel defects detection. Neural Networks, 98: 16-33, 2018.

[Voloshynovskiy, 10] Voloshynovskiy S., Koval O., Beekhof F., Farhadzadeh F. and Holotyak T., Information-theoretical analysis of private content identification, 2010 IEEE Information Theory Workshop, Dublin, 2010.

[Yu, 13] Yu J., Qin Z., Wan T., Zhang X., Feature integration analysis of bag-of-features model for image retrieval, Neurocomputing 120, 355-364, image Feature Detection and Description, 2013.

[Yuan, 15] Yuan Y., Hu X., Bag-of-words and object-based classification for cloud extraction from satellite imagery, Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of 8 (8) 4197-4205, 2015.

[Zhao, 15] Zhao Ch., Li X., and Cang Y.. Bisecting k-means clustering based face recognition using block-based bag of words model. Optik - International Journal for Light and Electron Optics, 126(19):1761-1766, 2015.