An Image Is Worth 1000 Words: Exploring Image Search Capabilities Using BoVW and GLCM

Cătălin-Aurelian CIOCÎRLAN, Andrei-Radu DĂNILĂ

Intelligent Systems and Computer Vision
University "Politehnica" of Bucharest, Bucharest, Romania
mail: {catalin.ciocirlan, andrei_radu.danila}@stud.etti.upb.ro

***Abstract*—In this paper we propose two end-to-end fully automatic content-based image retrieving (CBIR) systems. The first one is constructed using the Bag of Visual Words method while the second one is used for comparison, being based on 2^{nd} order probability distributions, more specifically, on gray-level co-occurrence matrix (GLCM) features. For retrieving images, both system are design to use a clustering algorithm. Our solution is trained and tested on *The COREL database for Image Retrieval*, which is an industry standard in this field. We reported the precision for our systems based on the top 1, 3, 5 and 10 returned images in order to compare the two systems and decide on the better solution. Another explored feature of the the proposed systems is their scalability, allowing new images to be indexed, which results in better generalization, a process called self-adjusting continual learning.**

***Index Terms*—image retrieval, bag of visual words, gray-level co-occurrence matrix, k-nearest neighbours**

I. Introduction

Image retrieval is the task of searching for and retrieving images from a large collection based on their visual content. It has become increasingly important in recent years with the growth of the internet and the availability of large image databases. Image retrieval can be done in various ways, such as text-based query, example-based query, or content-based query.

One of the most popular approaches for image retrieval is the use of gray-level co-occurrence matrix (GLCM) which is a method that uses the statistical properties of the image pixels to extract the texture features. It is a matrix that contains the frequency of occurrence of two pixels with a specific gray-level value and with a specific spatial relationship (distance and angle) between them. These features can be used to classify and retrieve images by capturing the texture information of the image.

Another approach for content-based image retrieval is the use of local features, such as SIFT or SURF, and the bag of visual words (BoVW) representation. This approach is based on the idea of creating a *vocabulary* of visual words, which are typically local features extracted from a set of training images. The images are then represented as a histogram of visual words and the histograms are used as a feature representation for image retrieval and classification tasks.

In this article, we will discuss the GLCM method and its usage in image retrieval and its ability to capture texture information, as well as the BoVW method for image retrieval and its limitations. We will also compare the performance of both methods in image retrieval tasks.

The rest of the paper is structured as follows: Section II discusses the main theory behind a CBIR system, Section III introduces the dataset, feature extraction and proposed solution and Section IV shows the obtained results. Finally, Section V presents the general conclusions of the authors.

II. Problem formulation

Given a reference image i_{ref} and a set $I = \{i_1, i_2, \ldots, i_N\}$ of N images, an image retrieval system is capable of searching, identifying and retrieving the most similar images with i_{ref} from the given set.

Most systems available are using metadata, such as captions or keywords, in order to classify images in small batches and retrieve the entire batch when asked for a picture which matches the description. However, more modern approaches are not using these external information. Instead, they use the image content to more accurately describe the subjects in that picture. Some examples of CBIR search engines available on the internet are listed in Table I, where M stand for million (of images) and `N/A` means not available.

Name	Domain	Index size
Google Image Search	General purpose	`N/A`
Yandex Image Search	General purpose	10000M
Baidu Image Search	General purpose	1000M
ID My Pill	Medicine identification	`N/A`
Shopachu	Shopping & Fashion	1M
eBay Image Search	Shopping & Fashion	20M

TABLE I: Examples of available CBIR systems.

A. Machine learning

K-means is a widely used clustering algorithm that aims to partition a dataset into k clusters, where k is a user-specified parameter. The principle of k-means is to define clusters as the set of points that are closest to a particular centroid, which is a point representing the center of a cluster. Given a dataset of N points $X = \{x_1, x_2, x_3, \ldots, x_N\}$, and a specified number of clusters k, the k-means algorithm proceeds in two steps:

1) Initialize k centroids $U = \{u_1, u_2, u_3, \ldots, u_k\}$, where u_i is the centroid of cluster i;
2) For each point x_j in dataset, assign x_j to the cluster i , where i is the closest centroid, using a distance metric such as Euclidean distance, Manhattan distance, Cosine Similarity, Kullback-Leibler divergence, etc., which are described bellow:
   - *Euclidean distance*:
   $$D_E = || \, x - y \, ||_2^2 = \sqrt{\sum_{i=1}^{N} | \, x_i - y_i \, |^2} \quad (1)$$
   where x and y are the two N sized vectors between which the euclidean distance is measured;
   - *Manhattan distance*:
   $$D_M = || \, x - y \, ||_1 = \sum_{i=1}^{N} | \, x_i - y_i \, |; \quad (2)$$
   - *Cosine Similarity*:
   $$S_C(x, y) = \frac{x \cdot y}{|| \, x \, || \, || \, y \, ||} = \frac{\sum_{i=1}^{N} x_i y_i}{\sqrt{\sum_{i=1}^{N} x_i^2} \sqrt{\sum_{i=1}^{N} y_i^2}}; \quad (3)$$
   - *Kullback-Leibler divergence*:
   $$D_{KL}(P \, || \, Q) = \sum_{x \in \mathcal{X}} P(x) \log \left(\frac{P(x)}{Q(x)} \right) \quad (4)$$
   where P and Q are two discrete probability functions defined on the same sample space $\mathcal{X}$.
3) Recompute the centroid of each cluster as the mean of all points assigned to that cluster;
4) Repeat steps 2 and 3 until the centroids stop changing or a maximum number of iterations is reached.

The main advantage of k-means is its simplicity and scalability. It can be applied to large datasets and it is easy to implement. However, the k-means algorithm is sensitive to the initial centroid positions, and it can get stuck in local optima. Additionally, the choice of the k (number of clusters) parameter can also represent a problem.

K-nearest neighbors (k-NN) is a non-parametric method used for classification and regression. The general idea is to find the k training examples that are the most similar to the test example and then use the class labels of these nearest neighbors to predict the class label of the test example. Given a set of training examples X and their corresponding class labels y, and a test example x_{new}, the k-NN algorithm proceeds in three steps:

1) Compute the distance between x_{new} and all training examples in X. This is typically done using a distance metric such as the ones described above.
2) Select the k training examples from X that are closest to x_{new}. These are the k "nearest neighbors" of x_{new}.
3) Finally, use the class labels, from y, of the k nearest neighbors to predict the class label of x_{new}. This is typically done by taking a majority vote among the k nearest neighbors.

Unsupervised nearest neighbors learning is a variation of the k-nearest neighbors algorithm that is used for unsupervised learning tasks, such as clustering or image retrieval. In unsupervised nearest neighbors learning, the goal is to group similar data points together into clusters, without the use of predefined class labels.

The basic idea behind the unsupervised nearest neighbors algorithm is to find the k-nearest neighbors of each data point, and then use these nearest neighbors to define the clusters. The algorithm starts by initializing each data point as its own cluster, and then repeatedly merges the closest clusters together until a stopping criterion is met.

B. Image processing and feature extraction

We chose to use the co-occurrence matrix, which is used to represent the frequency of co-occurring events in a given context. It is commonly used in information retrieval to identify patterns and relationships between structured data. In image processing, co-occurrence matrices are used to analyze the texture of an image by counting the number of times different pairs of pixels appear next to each other. They are introduces by Haralick et al. in [1], and still are one of the main methods used today for describing textures and for feature extraction. Because they are derived from the GLCM and introduced by Haralick, these features are usually called Haralick features. Amongst them, we used the contrast, dissimilarity, homogeneity, energy and correlation. These features are described mathematically in Equations 5 – 9

- *Contrast*:
$$C = \sum_{i=1}^{N} \sum_{j=1}^{N} (i - j)^2 p(i, j) \quad (5)$$
where N is the number of gray levels used and $p(i, j)$ is the normalized GLCM matrix at position (i, j);
- *Dissimilarity*:
$$D = \sum_{i=1}^{N} \sum_{j=1}^{N} | \, i \cdot j \, | \cdot p(i, j); \quad (6)$$
- *Homogeneity*:
$$H = \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{p(i, j)}{1 + (i - j)^2}; \quad (7)$$
- *Energy*:
$$E = \sum_{i=1}^{N} \sum_{j=1}^{N} p(i, j)^2; \quad (8)$$
- *Correlation*:
$$Corr = \sum_{i=1}^{N} \sum_{j=1}^{N} \left(\frac{i - \mu_x}{\sigma_x} \right) \left(\frac{j - \mu_y}{\sigma_y} \right) p(i, j) \quad (9)$$
where $\mu_x = \sum_{i=1}^{N} i \cdot p_x(i)$ and $\sigma_x^2 = \sum_{i=1}^{N} (i - \mu_x)^2 \cdot p_x(i)$ are the mean and standard deviation of the x-axis co-

occurrences probability $p_x = \sum_{j=1}^{N} p(i,j)$, and analogous for the y-axis.

Bag-of-Visual-Words (BoVW) is a method used for image retrieval and classification tasks that allows images to be represented as a histogram of visual words. It is based on the idea of creating a *vocabulary* of visual words, which are typically local features, such as SIFT, extracted from a set of training images [2].

SIFT (Scale-Invariant Feature Transform) [3] is a method used to extract features from images that are invariant to changes in scale, rotation, and affine distortion. Due to its robustness to scale and rotation changes, it is often considered as a standard feature descriptor in the field of computer vision and image processing. Extracting the SIFT descriptors from an image involves several steps:

1) Scale-space extrema detection: The algorithm starts by constructing a scale-space representation of the image by smoothing the image with a Gaussian kernel at multiple scales. The scale-space representation is then used to identify stable keypoints in the image that are invariant to scale changes;
2) Keypoint localization: Once the keypoints have been identified, the algorithm refines their location and scale by fitting a detailed model to the keypoints;
3) Orientation assignment: The algorithm assigns an orientation to each keypoint by analyzing the local image gradient around the keypoint. This step is done to make the features rotation invariant;
4) Feature descriptor: Once the keypoints have been localized and oriented, the algorithm computes a feature descriptor for each keypoint. The descriptor is a vector of local image gradient information that describes the keypoint's appearance.

The BoVW method is efficient and simple to implement, but it has several limitations. One limitation is that it does not take into account the spatial relationships between visual words, which can be important for some image retrieval and classification tasks. Additionally, it requires a large number of visual words to achieve good results, which can increase the computational complexity of the method.

III. Experimental setup

A. Dataset

The dataset selected for this task is *The COREL database for Content based Image Retrieval* [4]. It contains over 10000 images classified in 80 different groups such as: *bonsai*, *cloud*, *dog*, *elephant*, *iceberg*, or *waterfall*. Similar concepts are also available in different groups, which can result in difficulties when evaluating the system. Nonetheless, the sheer size of the dataset makes it useful, by providing the necessary tools for a machine learning algorithm to generalize on any given subject from the database.

COREL has been used in many publications to demonstrate the performance of content-based image retrieval systems and has become a defacto standard in the field [5]. However, many researchers have used different subsets from this collection, making results comparison a difficult task. Some images present in this dataset, alongside their labels are presented in Figure 1 .

Fig. 1: Examples from the COREL dataset with their associated labels

It is important to note that this dataset is a relatively an old one and it might not reflect the current state of the art in terms of image resolution, diversity or complexity.

For this task, we used a split of 85% for training and 15% for testing. Because classes are already relatively balanced, as it can be seen in Figure 2, we do not have to take into account any additional measures in order to balance the dataset.

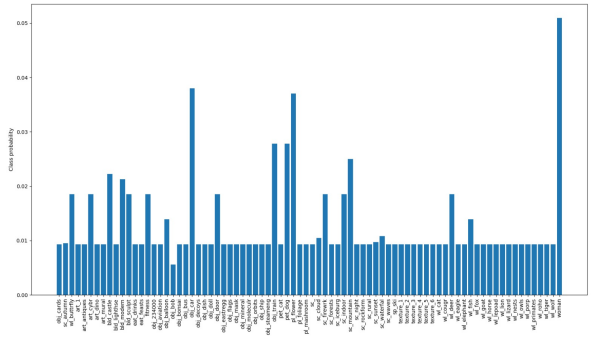

Fig. 2: Class distribution for the COREL dataset

B. First proposed solution – BoVW

For the BoVW implementation, we used the SIFT method to extract local features from all the images of the training set. These local features are extracted near the centroid of the image in order to ensure that the most keypoints are related to the subject of the image and are not extracted from the background [6]. An example of the centroid of an image is illustrated in Figure 3. Using all the classes available, and this method of subject localization, a total number of 684,693 128-dimensional feature vectors were extracted from 9,180 images.

During training, we clustered the descriptors into a *vocabulary* of 400 visual words, using the k-means algorithm. We considered this number of clusters a reasonable trade-off between performance and computational cost.

We again iterated over the training set and for each image computed a histogram which shows the frequency of apparitions of each visual word in the image. These histograms are

Fig. 3: Example for the centroid of an image. The goat, which is the subject of the image is correctly identified while the background is discarded as it does not represent an important part in our image

the base on which we fit an unsupervised k-NN algorithm, in order to gain a measure of the similarity between images and match them according to their local descriptors.

The testing phase encapsulates a similar process, in which from a test image the local features are extracted and categorized into the 400-word vocabulary. A histogram is computed based on the content of the words present in the image and the k images with the closest histogram to the test image are retrieved. A histogram of the whole vocabulary is depicted in Figure 4.

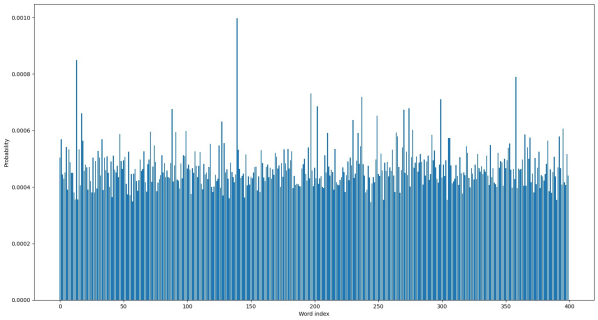

Fig. 4: Histogram obtained after clustering the visual words obtain from the SIFT features using k-means algorithm. We created 400 clusters, which gives us a relatively uniformed distribution.

C. Second proposed solution – GLCM

In order to extract the GLCM features, the first step is to quantize the image into a set of discrete gray levels. Next, the GLCM is constructed by counting the number of times each pair of pixels with a specific gray level appears next to each other in the image. Different directions are considered, each of them resulting in an independent matrix. For each direction, the GLCM is calculated by moving one pixel at a time in the chosen direction and counting the number of times each pair of pixels appears. This process is repeated for each direction and the resulting matrices are used to calculate the Haralick features.

From our test, the best results are obtained using 26 level of quantization, and four directions, as follows: 0, 45, 90 and 135 degrees. These extracted features for these 4 GLCM are then fed as input to a k-NN algorithm. We chose k-NN because it does not require a training phase, only to store the desired feature vectors. At testing, depending on how many top results we want, we *trained* different models: 1-NN, 3-NN, 5-NN and 10-NN. Because the extraction of those features is much more faster then the BoVW alternative, we were able to perform multiple tests using this method.

For the k-NN algorithm, multiple distances were tested in order to obtain better results. Some of the metrics tested are: the mean absolute error (MAE), mean squared error (MSE), euclidean distance, or Kullback-Leibler (KL) divergence. From our test, all of the aforementioned metrics give similar results. We decided on keeping the KL as the main metric and consider the feature vectors as a discrete probability function for each analyzed image.

IV. Performance evaluation

Most CBIR systems use precision (P), explained in Equation 10 in order to determine their overall performance.

$$P = \frac{TP}{TP + FP} \tag{10}$$

where TP stands for true positive and FP for false positive.

We compared our findings with the ones found in literature and presented the results in Table II, where S_1 represents our first proposed solutions, using BOW, and S_2 is the solution using GLCM features.

Method	Subset	Metric	Value
[7]	10 cat.	Avg. P for top 30	**50%**
S_2	10 cat.	Avg P for top 1	49.63%
S_2	10 cat.	Avg P for top 3	47.58%
S_2	10 cat.	Avg P for top 5	43.20%
S_2	10 cat.	Avg P for top 10	33.17%
[8]	Corel-5000	Avg. P for top 12	**55.92%**
S_2	40 cat.	Avg P for top 1	27.46%
S_2	40 cat.	Avg P for top 3	21.00%
S_2	40 cat.	Avg P for top 5	17.38%
S_2	40 cat.	Avg P for top 10	13.28%
S_1	80 cat.	Avg P for top 1	10.31%
S_1	80 cat.	Avg P for top 3	9.89%
S_1	80 cat.	Avg P for top 5	10.12%
S_1	80 cat.	Avg P for top 10	9.74%
S_2	80 cat.	Avg P for top 1	**20.64%**
S_2	80 cat.	Avg P for top 3	16.74%
S_2	80 cat.	Avg P for top 5	13.52%
S_2	80 cat.	Avg P for top 10	9.81%

TABLE II: Results obtained using our proposed method

In their work, Hoiem et al. [7] obtain 50% average precision, for the first 30 retrieved images, by utilizing a Naive Bayes algorithm on the HSV color space, extracting features about hue and saturation. Liu et al. [8] created new micro-descriptors

(MSD) for the images presented in their variant of the COREL dataset. They used the L_1 distance to determine the most similar images using these descriptors.

Our results indicates that, when using GLCM, the first images that are retrieved are better suited than the ones after. This means that if the user requests only one or two matches, they will be close enough but when requesting multiple images, the results tends to diverge from the subject. Some visual result for the GLCM features are depicted in Figure 5.

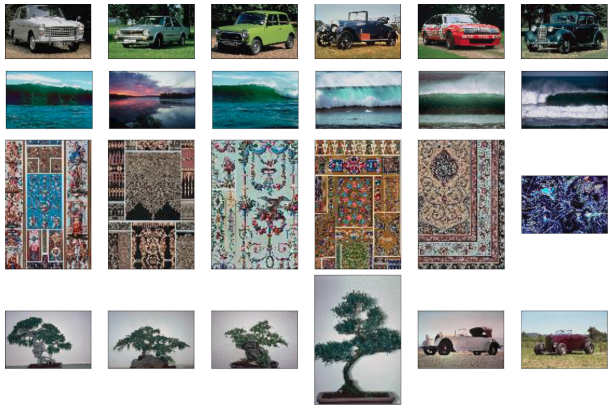

Fig. 5: Results using the GLCM features. For each row, the first image is the search one, followed by the first 5 images retrieved (top 5) by the system.

Even though the precision for the system using BoVW is lower, the number of cases when all the images are retrieved correctly, meaning they are belonging to the same class, is greater than in the case of GLCM. This seems to be the main problem of the BoVW system, because the output can either be very precise or very different from the requested input. We conclude that this system, although very interesting in structure, has a very high variance and is not as useful as the one based on GLCM. For comparison, the results using BoVW features are illustrated in Figure 6.

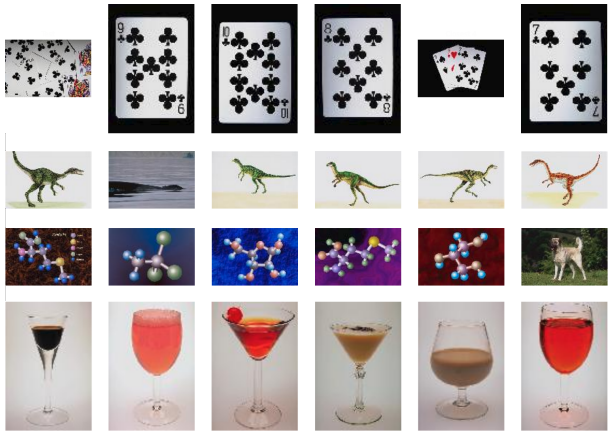

Fig. 6: Results using the BOW features. The same structures is respected as the one presenting the results using GLCM.

All resources used for this paper are available on GitLab [1]. Access to the repository may require obtaining approval from the authors.

V. Conclusion

In this paper, we presented two end-to-end systems for content-based image retrieval, one based on BoVW and the other based on GLCM features. Both systems were evaluated using a benchmark dataset and the results showed that while they performed well, there is still much room for improvement.

One promising approach to improve the performance of the systems is to combine them using ensemble learning. This can be done by using the decision scores from both systems and fusing them to make the final decision, a method called bagging, or by training another classifier to decide which of the system is better in different scenarios, also known as stack learning. Those approaches can leverage the strengths of both systems and potentially improve the overall performance.

Finally, incorporating deep learning techniques can also be an important direction for future. Convolutional neural networks (CNNs) are used for most task in computer vision, including image retrieval ones. CNNs are also good for an end-to-end system because they can be integrated with much easy in a pipeline, being invariant to most of the transformations commonly applied to images such as rotations or translations, as stated by He et al. in [9].

References

[1] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 6, pp. 610–621, 1973.

[2] S. Basak and R. Parekh, "2016 : An improved bag-of-features approach for object recognition from natural images," *International Journal of Computer Applications*, vol. 151, pp. 975–8887, 11 2016.

[3] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2. Ieee, 1999, pp. 1150–1157.

[4] "The corel database for content based image retrieval." [Online]. Available: https://sites.google.com/site/dctresearch/Home/content-based-image-retrieval

[5] H. Müller, S. Marchand-Maillet, and T. Pun, "The truth about corel-evaluation in image retrieval," in *International Conference on Image and Video Retrieval*. Springer, 2002, pp. 38–49.

[6] S. C. Hoi, W. Liu, and S.-F. Chang, "Semi-supervised distance metric learning for collaborative image retrieval and clustering," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 6, no. 3, pp. 1–26, 2010.

[7] D. Hoiem, R. Sukthankar, H. Schneiderman, and L. Huston, "Object-based image retrieval using the statistical structure of images," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 2. IEEE, 2004, pp. II–II.

[8] G.-H. Liu, Z.-Y. Li, L. Zhang, and Y. Xu, "Image retrieval based on micro-structure descriptor," *Pattern Recognition*, vol. 44, no. 9, pp. 2123–2133, 2011, computer Analysis of Images and Patterns.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[1] https://gitlab.com/iom-sign-language/cppsms-proiect-corel