



Content-based image retrieval using feature weighting and C-means clustering in a multi-label classification framework

Samaneh Ghodrathnama¹ · Hamid Abrishami Moghaddam²

Received: 27 January 2018 / Accepted: 8 June 2020 / Published online: 18 June 2020
© Springer-Verlag London Ltd., part of Springer Nature 2020

Abstract

In this paper, a novel learning algorithm based on feature weighting is proposed to improve the performance of image classification or retrieval systems in a multi-label framework. The goal is to exploit maximally the beneficial properties of each feature in the system. Since each feature can separate more effectively some of the image classes, it is hypothesized that the weights of various features at some states can be traded off against each other. The training phase of the suggested algorithm is performed in two stages: (1) The input images are clustered using a supervised C-means method iteratively; (2) image features are weighted using a local feature weighting method in each cluster. These weights are determined by considering the importance of each feature in minimizing the classification error on each cluster. In the testing phase, the cluster corresponding to the query is found first. Then, the most similar images are retrieved in the multi-label framework using the feature weights assigned to that cluster. Experimental results on three well-known, public and international image datasets demonstrate that our proposed method leads to significant performance gains over existing methods.

Keywords CBIR · Multi-label · Feature weighting · C-means clustering · KNN classification

1 Introduction

Content-based image retrieval (CBIR) is a method to retrieve the most similar images to a query image. This system searches similar images using features such as color, texture, and shape. One of the most common methods that can be used for this purpose is the K-nearest neighbors algorithm (KNN) [1–3]. This algorithm returns the most similar image using a distance function within feature space.

Retrieving similar images using only low-level features is one of the main shortcomings of the conventional content-based image retrieval (CBIR) systems. Among various techniques, machine learning such as deep learning [4–10], sparse coding for bag-of-words (BoW)-based approach [11, 12], Fisher vectors [13], etc., have been actively investigated as possible directions to bridge the semantic gap in the long term. This paper presents an effort to overcome

this drawback and proposes a CBIR approach in which the retrieved labels of images in the multi-label classification framework satisfy user expectations.

It is also noteworthy to consider that images are in different classes and each of them would be more separable based on one particular kind of feature. For instance, assume a color feature would be more discriminant in one image, so this image is more distinguishable based on the color feature, while another one would be more discernible based on texture or shape features. Therefore, to have an effective search with more accurate results, one way is to search images based on a combination of features using the distance function. Besides, for the purpose of using the discriminative power of each feature concurrently and more purposefully, it is advantageous to train the weights for features. For example, when color features are more likely to be discriminant than the other features, more weight must be assigned to the color features and vice versa [14–16].

Then, if weighting is supposed to be applied to the whole data, considering that the data scope is large and dispersed, it would be difficult to find an ideal set of weights for features. Contrastingly, if the weights are found within subspaces, it will be more probable to achieve weights which yield more accurate retrieval results. Thus, we suggest dividing feature

✉ Hamid Abrishami Moghaddam
moghaddam@kntu.ac.ir

¹ Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

² Faculty of Electrical and Computer Engineering, K. N. Toosi University of Technology, Tehran, Iran

space into some regions in order to do a more intelligent search within smaller regions to retrieve similar images to the query. This strategy makes searching faster and more accurate [17–19]. The clustering algorithm is an intelligent method to divide the space into regions containing more similar data. Obviously, finding samples similar to the query image would be easier in such regions [20].

Based on the aforementioned discussion, in this paper a local feature weighting algorithm [21] is used for CBIR application. This algorithm first clusters the data and simultaneously trains the features in each cluster to have the appropriate weights for such features based on minimizing the classification error rate. In other words, both clustering and classification methods are used conjointly to create a powerful framework for retrieving similar images to the query.

The rest of the paper is organized as follows. In Sect. 2, we review some related studies in our multi-label learning framework such as C-means clustering, KNN and a feature extraction algorithm, which are used in this paper. In Sect. 3, we explain our method for multi-label classification using feature weighting algorithms and C-means clustering based on classification error rate minimization (MLC-FWC). In Sect. 4, we present experimental results and show the superiority of our algorithm over its counterparts which do not use either feature weighting or a multi-label framework. Finally, conclusions are presented in Sect. 5.

2 Related works

2.1 Multi-label classification

Multi-label classification has become popular in many areas [22]. For instance, in text categorization, a document may belong to multiple classes simultaneously [23], or in an automatic image annotation scenario, a scene may be linked to multiple concepts [24]. Similarly, different labels can be assigned to each video in a video indexing domain [25]. This situation also happens in functional genomics where multiple functions may be represented by a single gene [26]. In all the above examples, each instance is associated with multiple labels, and this situation is different from traditional single-label classification. Figure 1 shows a sample image from the Corel1000 database that can be classified as beach, sea, sky, tree and even mountain by considering the semantic meaning of the image.

There are some methods that do CBIR in multi-label framework. For example, in [27, 28] the authors created a visual dictionary for each group in the dataset. Then, they proposed a system for image retrieval based on local features using a BoW model that considered mid-level representations. The methods based on BoW create a codebook of visual discriminating



Fig. 1 Sample multi-label data from Corel1000: beach, sea, sky, tree and mountain

patches (visual words) and then compute statistics (using the codebook) on the visual word occurrences in the test image. The system tries to bring more accuracy with the option to use local rather than global features to reduce the semantic gap issue and enhance the performance of the CBIR by performing image annotation.

In what follows, we provide a formal definition of the multi-label classification problem [29]. Let \mathbf{X} denote the instance space and \mathbf{Y} the set of class labels. Then, the goal is to learn a function $f : \mathbf{X} \rightarrow 2^{\mathbf{Y}}$ which is defined on a given dataset $\{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \dots, (\mathbf{x}_N, \mathbf{y}_N)\}$, where $\mathbf{x}_i \in \mathbf{X}, i = 1 \dots N$ is an instance and $\mathbf{y}_i \subseteq \mathbf{Y}$ is a set of labels $\{y_{i1}, y_{i2}, \dots, y_{il_i}\}$, $y_{ik} \in \mathbf{Y}, k = 1, 2, \dots, l_i, l_i = 1 \dots H$. Here, H is the number of labels, N is the number of data samples and l_i denotes the number of labels in \mathbf{y}_i .

2.2 KNN classification

The KNN classification rule has been used successfully in many pattern recognition applications. This method is simple, and the distance measure in this algorithm can be various kinds of distance function. We use Euclidean distance, mostly used for dissimilarity measurement in image retrieval due to its effectiveness. It measures the distance between two vectors of images by $d(\mathbf{x}, \mathbf{x}') = \sqrt{\sum_{j=1}^n (x_j - x'_j)^2}$, where \mathbf{x} and \mathbf{x}' are two different vectors and n is the feature dimension. Since the purpose of our algorithm is to assign weights to the features, in this framework we use the weighted Euclidean distance, defined as:

$$d_w(\mathbf{x}, \mathbf{x}') = \sqrt{\sum_{j=1}^n w_j (x_j - x'_j)^2} \quad (1)$$

where w_j is the weight corresponding to the j th feature.

2.3 C-means clustering

C-means [30] is one of the most reliable and widely used clustering algorithms. The procedure follows a simple and quick way to cluster a dataset to a certain number of clusters. The main idea is to define C centroids, one for each cluster as a prototype. Then, a binding is created between each data point and one of these C cluster centroids. First, the cluster centroids are set randomly [31]. This algorithm minimizes iteratively the total squared Euclidean distance between the data points in feature space and the cluster centroids. In each iteration, the centroids change their location with respect to the data points in each cluster. After changing the centers, the data points associated with each cluster will change based on the new centroids. These steps will be repeated until no more changes happen or centroids move too slightly (less than a threshold).

Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ be a set of data points and $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_C\}$ represent the C cluster centers. C -partitioning $\{\theta_1 \dots \theta_C\}$ is created by the Euclidean C-means algorithm with the aim to minimize the following objective function:

$$J_1(\mathbf{V}, \mathbf{W}) = \sum_{k=1}^C \sum_{i=1}^{N_k} \|\mathbf{x}_i - \mathbf{v}_k\|_{\mathbf{w}_k}^2 \tag{2}$$

where $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_C\}$ is the set of local feature weights in each cluster and $\|\mathbf{x}_i - \mathbf{v}_k\|_{\mathbf{w}_k}^2$ is a weighted Euclidean distance measure between a data point \mathbf{x}_i and the cluster center \mathbf{v}_k . In the k th cluster, N_k is the number of samples and $\mathbf{w}_k = \{w_{kj}, 1 \leq j \leq n\}$, where n is the number of features (dimensions).

2.4 Feature weighting

Feature weighting is a technique used to approximate the optimal degree of influence of individual features using a training set. When successfully applied, highly relevant features are attributed a high weight value, whereas irrelevant features are given a weight value close to zero [32]. Feature weighting can be used not only to improve the classification accuracy but also to discard features with weights below a certain threshold value and thereby increase the resource efficiency of the classifier [33, 34].

2.5 Feature extraction

Feature extraction has an undeniable role in the efficiency of the classification process within the image retrieval system. In the proposed method, we use low-level features such as color and texture. For texture features, the wavelet correlogram method [35] and, for color, the method based

on a color correlogram with vector quantization [36] are used, because they have shown a good individual ability to classify images.

2.5.1 Wavelet correlogram

Wavelet correlogram is a CBIR system based on texture. It applies the color correlogram method [37] on quantized wavelet coefficients and creates texture features. As a result, this method gains from the multiscale–multiresolution characteristic of the wavelet transform and inherits the translation–rotation invariance property from the color correlogram.

According to the enhanced Gabor wavelet correlogram (EGWC) indexing algorithm, three different scales of wavelet transform are calculated.

In each scale, the non-maximum suppression block is aimed to reduce the data redundancy as it is explained in [35]. Then, the coefficients are discretized using three sets of quantization thresholds. Thanks to optimal selection of thresholds, improved retrieval performance is guaranteed. Finally, the text feature is calculated based on horizontal and vertical auto-correlogram of quantized coefficients.

2.5.2 Vector-quantized color correlogram

A vector-quantized color correlogram (VQCC) is a compound CBIR system designed to utilize both color correlogram and vector quantization methods [36]. The VQCC algorithm quantizes the RGB color of the input image using a vector quantizer. The resulting pixel vector represents a cluster of points in the three-dimensional RGB space. Then, for each pixel inside a cluster, based on its Euclidean distance from the cluster center, a binary code is specified. In other words, the pixel is labeled with the code in the codebook corresponding to the closest cluster center. Finally, an index vector is created by computing the auto-correlogram of these binary codes.

Image retrieval is performed using a comparison between indices. The codebook is constructed by the following three steps:

- a. Some images are selected randomly from the dataset. (The number of images is determined experimentally based on the total number of images and image categories.)
- b. C-means clustering extracts pixel clusters for this population.
- c. The centers of clusters make code vectors in the codebook for quantization.

3 Multi-label classification using feature weighting and C-means clustering (MLC-FWC)

The goal of the learning algorithm is to minimize the total error rate of the KNN classifier [32]. To achieve this goal, we use C-means clustering and local feature weighting in each cluster. The weights of features as parameters of the distance function are learnt by our proposed method that attempts to minimize the average error rate per cluster.

The schematic diagram of the proposed method is shown in Fig. 2. In the training phase, first, texture and color features of database images are extracted. Then, in an iterative process, a supervised C-means clustering, which is explained later, is run to divide the input space into proper subspaces, and then a KNN classifier tries to minimize the classification error on each cluster by means of local feature weighting. This procedure is executed till the feature weights converge. In the test phase, first the nearest cluster to the query is found based on its distance from the center of each cluster. The nearest cluster together with the weights (found in the training phase) represents a new input space within which the most similar images to the input query are retrieved.

3.1 Training phase

In the training phase of this algorithm, clustering and classification are combined to construct a model that benefits from the advantages of both methods. In other words, clusters are improved by taking the classifier feedback, and classification is improved by using clustering information. The feedback from classification is used for relocating misclassified samples into other clusters and finding a new partitioning of the data, so a new labeling is assigned accordingly. In other words, a quasi-supervised clustering is used in this framework to classify more accurately. In contrast, classification can be improved by the clustering information, since it divides the space into some related subspaces. This process continues till no sample is displaced between clusters. For this process, a method is also proposed to weight features locally based on minimizing the error of each cluster [32].

So in our framework, clustering and classification are incorporated in a single objective function in order to partition the input space into related subspaces. It makes classification and retrieval simpler and faster. Our purpose is to construct related clusters and minimize the error rate of the NN classifier within each cluster using leave-one-out (LOO) error rate by doing feature weighting. These weights can be represented as a weight matrix: $\mathbf{W} = \{\mathbf{w}_k, 1 \leq k \leq C\}$, where C is the number of clusters. It is noteworthy that this definition assigns distinct weights to the different features in each cluster.

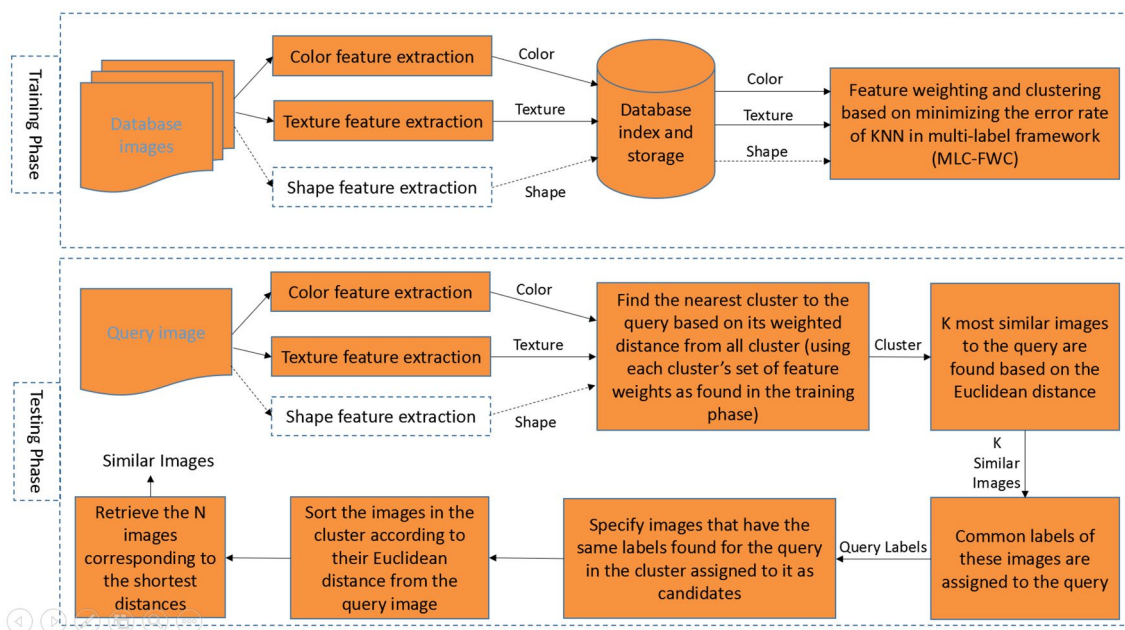


Fig. 2 The schematic diagram of multi-label classification with features weighting and C-means clustering (MLC-FWC). The classification error is minimized within each cluster. (Among images with the

same label, retrieve N similar images according to their distance from the query.) The dashed boxes show how shape features can be integrated to MLC-FWC

MLC-FWC first uses an approximation of the NN classification error as introduced in [32] to define a classification objective function. Then, it tries to minimize the objective function such that the samples are assigned to their corresponding cluster.

In the training phase, first the data are randomly clustered. Then, the idea is to learn the weights of the features in each cluster such that, for each sample \mathbf{x}_i , the nearest same-class sample seems closer, while the nearest different-class sample appears farther. This strategy reduces the weighted distance between same-class samples. The LOO NN classification error as a function of feature weights can be written as [32]:

$$J_2(\mathbf{W}) = \frac{1}{N} \sum_{k=1}^C \sum_{i=1}^{N_k} S_\beta \left(\frac{d_w(\mathbf{x}_i, \mathbf{x}_=)}{d_w(\mathbf{x}_i, \mathbf{x}_\neq)} \right) \tag{3}$$

where N is the number of samples, $\mathbf{x}_=, \mathbf{x}_\neq$ are the nearest same-class and different-class samples of \mathbf{x}_i in the k th cluster and d_w is the weighted Euclidean distance as defined in (2). As the framework performs on the multi-label images, the same-class sample $\mathbf{x}_=$ denotes one which has the largest number of common labels with \mathbf{x}_i , and different-class sample \mathbf{x}_\neq indicates one that has the fewest common labels with \mathbf{x}_i . If there are more than one sample with this condition, the nearest and farthest ones are selected as the same-class and different-class samples, respectively. Let S_β be the sigmoid function with slope β , centered at $z = 1$:

$$S_\beta(z) = \frac{1}{1 + e^{\beta(1-z)}} \tag{4}$$

For large β , if \mathbf{x}_i is closer to some prototypes of its own class than to any other from a different class, then $d_w(\mathbf{x}_i, \mathbf{x}_=) < d_w(\mathbf{x}_i, \mathbf{x}_\neq)$ and the argument of S_β will be smaller than 1 and S_β approaches 0. On the contrary, if \mathbf{x}_i is closer to some prototypes of a different class than to any other from its own class, the argument of S_β will be greater than 1 and S_β approaches 1. Accordingly, $J_2(\mathbf{W})$ is in fact the LOO NN estimate of the misclassification probability over the training set \mathbf{X} .

In this paper, gradient descent optimization is employed, so it requires the function $J_2(\mathbf{W})$ to be differentiable with respect to the corresponding parameters \mathbf{w}_k and \mathbf{v}_k to be minimized, which is why using the sigmoid function is adopted here.

Now, we combine the two objective functions ($J_1(\mathbf{V}, \mathbf{W})$ and $J_2(\mathbf{W})$) which both try to minimize the error, one for classification and the other for clustering. So the overall objective function is defined as:

$$J(\mathbf{V}, \mathbf{W}) = J_1(\mathbf{V}, \mathbf{W}) + \gamma J_2(\mathbf{W}) \tag{5}$$

where γ is a weight which determines the trade-off between the two objective functions. Substituting $J_1(\mathbf{V}, \mathbf{W})$ and $J_2(\mathbf{W})$ yields to:

$$J(\mathbf{V}, \mathbf{W}) = \left(\sum_{k=1}^C \sum_{i=1}^{N_k} \|\mathbf{x}_i - \mathbf{v}_k\|_{\mathbf{w}_k}^2 + \frac{1}{N} \sum_{k=1}^C \sum_{i=1}^{N_k} S_\beta \left(\frac{d_w(\mathbf{x}_i, \mathbf{x}_=)}{d_w(\mathbf{x}_i, \mathbf{x}_\neq)} \right) \right) \tag{6}$$

In this equation, the first term specifies the clustering error rate, while the second term designates the classification error and γ is set to 1. Within each cluster, the weight of features corresponding to $(\mathbf{x}_=)$ should be modified to make this sample appear closer to \mathbf{x}_i in a feature-dependent manner, while those corresponding to (\mathbf{x}_\neq) should be modified such that it appears farther.

Obviously, (6) is differentiable. To solve the objective function, a gradient descent procedure is proposed which guarantees convergence. The gradient descent evolution equation is:

$$\theta^{t+1} = \theta^t - \alpha \left(\frac{\partial J(\theta)}{\partial \theta} \right) \tag{7}$$

where θ' and α denote the optimal parameters (here \mathbf{v} and \mathbf{w}) in the t th iteration and the step size, respectively. Before replacing $J(\theta)$ with the objective function in (6), we define:

$$R(\mathbf{x}_i) = \left(\frac{d_w(\mathbf{x}_i, \mathbf{x}_=)}{d_w(\mathbf{x}_i, \mathbf{x}_\neq)} \right) \tag{8}$$

The partial derivative of $J(\mathbf{V}, \mathbf{W})$ with respect to \mathbf{w}_k is calculated as:

$$\frac{\partial J(\mathbf{V}, \mathbf{W})}{\partial \mathbf{w}_k} \cong \sum_{i=1}^{N_k} \|\mathbf{x}_i - \mathbf{v}_k\|^2 + \frac{1}{N} \sum_{i=1}^{N_k} S'_\beta(R(\mathbf{x}_i)) \frac{\partial R(\mathbf{x}_i)}{\partial \mathbf{w}_k} \tag{9}$$

where the derivative of $S'_\beta(z)$ and $\frac{\partial R(\mathbf{x}_i)}{\partial \mathbf{w}_k}$ is:

$$S'_\beta(z) = \frac{dS_\beta(z)}{dz} = \frac{\beta e^{\beta(1-z)}}{(1 + e^{\beta(1-z)})^2} \tag{10}$$

$$\frac{\partial R(\mathbf{x}_i)}{\partial \mathbf{w}_k} = \frac{1}{2d_w^2(\mathbf{x}_i, \mathbf{x}_\neq)} \left(\frac{1}{R(\mathbf{x}_i)} \|\mathbf{x}_i - \mathbf{x}_=\|^2 - R(\mathbf{x}_i) \|\mathbf{x}_i - \mathbf{x}_\neq\|^2 \right) \tag{11}$$

And the partial derivation of $J(\mathbf{V}, \mathbf{W})$ with respect to \mathbf{v}_k is calculated as:

$$\frac{\partial J(\mathbf{V}, \mathbf{W})}{\partial \mathbf{v}_k} \cong \sum_{i=1}^{N_k} -2\mathbf{w}_k \odot (\mathbf{x}_i - \mathbf{v}_k) \tag{12}$$

Therefore, we have two parameters \mathbf{w}_k and \mathbf{v}_k that should be optimized. We use the gradient descent algorithm to optimize these parameters. In this approach, the number of clusters starts from an initial value and then is learnt dynamically by the algorithm; we first update \mathbf{W} in each cluster and then based on the modified \mathbf{W} , the cluster

centers will be updated. Samples are clustered again with the new weights and centers.

The gradient descent algorithm minimizes $J(\mathbf{V}, \mathbf{W})$ via an iterative procedure. At each iteration, the weights \mathbf{w}_k and \mathbf{v}_k are updated in the opposite direction to $\nabla J(\mathbf{V}, \mathbf{W})$ using the step sizes of α and δ :

$$\mathbf{w}_k^{(t+1)} = \mathbf{w}_k^{(t)} - \alpha \frac{\partial J(\mathbf{V}, \mathbf{W})}{\partial \mathbf{w}_k} \quad (13)$$

$$\mathbf{v}_k^{(t+1)} = \mathbf{v}_k^{(t)} - \delta \frac{\partial J(\mathbf{V}, \mathbf{W})}{\partial \mathbf{v}_k} \quad (14)$$

The values of α and δ are referred to as learning rates or learning step factors.

A pseudo-code of our training algorithm is presented in Fig. 3. It is noteworthy that all the parameters are initialized randomly. The maximum number of iterations is set empirically. The other parameter α , δ and β are selected experimentally for each dataset.

The computational complexity of the proposed algorithm as we can see in the pseudo-code of MLC-FWC algorithm in Fig. 3 is $O(I \times K \times (N_K + N))$, where I is the number of iterations, K is the number of clusters, N_K is the number of samples in each cluster and N is the number of all training samples. Based on the experiments, the algorithm converges within less than 10 iterations, which is not a large number, K is a constant much smaller than the number of samples, and therefore negligible. N_K is the number of samples in

```

Training phase:
X = Trainset
 $\alpha$  and  $\delta$  = Learning rate;  $\beta$  = Sigmoid slope;
Max_Iter = Maximum number of iterations;
#Clusters=Number of clusters;
W = Initial weight matrix;
V = Initial Centers;
(Clusters, V) = C-means(X, V, W);
Wnew = W;
for t = 1: Max_Iter
    for k = 1: #Clusters
        for i = 1: Nk
             $\mathbf{x}_s$  = same-class NN of  $\mathbf{x}_i$  in the kth cluster
             $\mathbf{x}_d$  = different-class NN of  $\mathbf{x}_i$  in the kth cluster
             $\frac{\partial J}{\partial \mathbf{w}_k}$  = Calculate  $\frac{\partial J}{\partial \mathbf{w}_k}$  using (9)
            Wnew = Update W using (13)
        end
    end
    W = Wnew;
    for j = 1: all train samples
         $\frac{\partial J}{\partial \mathbf{v}_k}$  = Calculate  $\frac{\partial J}{\partial \mathbf{v}_k}$  using (12)
        Vnew = Update V using (14)
    end
    (Clusters, V) = C-means(X, V, W);
end
return (V, W);

```

Fig. 3 A pseudo-code of the MLC-FWC algorithm

each cluster, which in the worst case is equal to N (for only one cluster case). Therefore, the overall complexity is $O(N)$.

3.2 Test phase

In the test phase, for each query, its distance to all cluster centers is calculated based on local feature weights in each cluster. The nearest one is considered as the cluster corresponding to the query image. In other words, we find the proper cluster k which minimizes $\sqrt{\sum_{j=1}^n w_{kj}(x_{ij} - v_{kj})^2}$ among all clusters ($k = 1, \dots, C$). Based on the K value, the common labels among KNN will be assigned to the query as labels. If $K = 1$, all labels of NN will be considered as the query label or if $K = 5$, the labels that are repeated two times or more between all 5NN will be assigned to the query image.

Then, M similar images within that cluster with the same labels are assigned to the query and are retrieved as the similar images in the retrieval framework based on their distance to the query image.

3.2.1 Experimental results

The proposed method was evaluated on the Corel1000 database of images. The database consists of 1000 images from 10 categories in which each category has 100 images. The categories are African people, beaches, buildings, buses, dinosaurs, elephants, roses, horses, mountains and food. All these categories were used for the experiments. This dataset is most often used for CBIR systems as a single-label dataset. Since our algorithm is introduced in a multi-label classification framework, we labeled manually this dataset with 1 to maximum 5 labels based on the related semantics of the image. We did the same for the Corel5000 dataset [35]. We also evaluated our proposed method using the Scene multi-label dataset [38] which is a benchmark for multi-label image classification containing 2407 natural scene images. Their characteristics are summarized in Table 1.

The performance of the proposed algorithm can be evaluated within two frameworks: first, based on criteria which are used for evaluating multi-label classification algorithms; second, by criteria that are used for evaluation of image

Table 1 Characteristics of data sets

Name	# of examples	# of features	# of labels	Feature type
Corel1000	1000	256 color 192 texture	5	Numeric (real)
Corel5000	5000	256 color 192 texture	5	Numeric (real)
Scene	2407	294	6	Numeric (real)

retrieval systems. In the following, various criteria of both approaches are discussed.

3.3 Evaluation criteria

The performance of any retrieval system can be measured in terms of its precision and recall. Precision measures the ability of the system to retrieve only models that are relevant, while recall measures the ability of the system to retrieve all models that are relevant. In this paper, we use the criteria for a multi-label framework as defined in [23] and for image retrieval as defined in [35].

3.3.1 Multi-label classification framework

Let $D = \{x_1, x_2, \dots, x_N\}$ be a multi-label evaluation dataset containing N labeled examples. Let $\hat{Y}_i = H(Q_i)$ be the predicted label set for the pattern Q_i as query image, while Y_i is the ground truth label set for x_i . A first metric called the averaged accuracy gives an average degree of similarity between the predicted and the ground truth label sets of all test examples:

$$AA(H, D) = \frac{1}{N} \sum_{i=1}^N \frac{|Y_i \cap \hat{Y}_i|}{|Y_i \cup \hat{Y}_i|} \tag{15}$$

where $|S|$ represents the cardinality of the set S . Two other metrics called average precision and average recall are also used in the literature [39] to evaluate a multi-label learning system. The former computes the proportion of correct positive predictions, while the latter calculates the proportion of true labels that have been predicted as positives:

$$AP(H, D) = \frac{1}{N} \sum_{i=1}^N \frac{|Y_i \cap \hat{Y}_i|}{|\hat{Y}_i|} \tag{16}$$

$$AR(H, D) = \frac{1}{N} \sum_{i=1}^N \frac{|Y_i \cap \hat{Y}_i|}{|Y_i|} \tag{17}$$

Another evaluation criterion is the F1 measure that is defined as the harmonic mean of the AP and AR metrics [40]:

$$F1 = \frac{2}{\frac{1}{AP} + \frac{1}{AR}} \tag{18}$$

The values of these evaluation criteria are in the interval $[0, 1]$. Larger values of these metrics correspond to higher classification quality.

3.3.2 Image retrieval framework

Let $Y(Q_i)$ be the set of retrieved images which are matched to the query image Q_i :

$$Y(Q_i) = \{I_i | \text{rank}(I_i) \leq M, I_i \in A\} \tag{19}$$

where $\text{rank}(I_i)$ is the rank of I_i among M retrieved images and A is the subset of the image database having the same cluster as the query image. Retrieval precision was used for evaluation of the image retrieval. Precision is defined as the ratio of the number of retrieved relevant images to the total number of retrieved images:

$$P(Q_i) = \frac{|Y(Q_i)|}{M} \tag{20}$$

where $|Y(Q_i)|$ represents the size of $Y(Q_i)$. The average precision per cluster is defined as:

$$P(\text{cluster}_k) = \sum_{i=1}^{N_k} P(Q_i) / N_k \tag{21}$$

where N_k is the number of images in the k th cluster. Finally, the average precision is given by:

$$P_{\text{avg}} = \sum_{k=1}^C P(\text{cluster}_k) / C \tag{22}$$

where C is the number of clusters.

4 Results and discussion

There are several approaches to improve retrieval effectivity by feature weighting, and also plenty of methods which address the multi-label classification problem. However, we could not find any algorithm which uses feature weighting in a multi-label classification and retrieval framework. Consequently, we decided to evaluate our algorithm in the four following ways. First, we compare our results before and after the MLC-FWC algorithm within the multi-label classification framework. Second, we compare the MLC-FWC method to methods in [41–47] which use feature weighting (FW) or feature fusion (FF) algorithms to combine features in a single-label image retrieval framework. Third, we compare our algorithm within a multi-label classification framework with the algorithms in [24, 29] which perform multi-label classification without feature weighting on the Scene multi-label dataset. Finally, the retrieval accuracy of the algorithm is evaluated by comparing it to two CBIR methods [35, 36] on the Corel1000 dataset within a multi-label image retrieval framework. For these

Table 2 Average values of accuracy, precision, recall and F1 measure resulting from applying MLC-FWC to Corel1000 database with/without feature combining by FW

Corel1000	Texture feature: EGWC Color feature: 64×4 VQCC			
	AA	AP	AR	F1
Color ($k=1$)	63.69	72.31	71.77	72.04
Weighted color ($K=3$)	57.74	75.05	62.41	66.05
Texture ($k=1$)	58.33	65.39	65.39	65.39
Weighted texture ($K=3$)	54.94	74.88	59.28	63.54
Color texture ($k=1$)	60.56	74.12	72.63	73.37
Weighted color texture ($K=3$)	66.65	74.72	75.42	75.06

Table 3 Average values of accuracy, precision, recall and F1 measure resulting from applying MLC-FWC to Corel5000 database with/without feature combining by FW

Corel5000	Texture feature: EGWC Color feature: 64×4 VQCC $K=30$			
	AA	AP	AR	F1
Color	39.28	41.42	46.50	43.81
Weighted color	43.29	49.01	45.22	46.26
Texture	40.33	42.51	47.23	44.02
Weighted texture	32.85	37.14	34.79	35.44
Color texture	37.72	49.38	40.91	44.80
Weighted color texture	44.75	49.41	45.34	47.28

results, all images are selected as query from the same dataset. In particular, a retrieved image is considered a match if and only if it is in the same category as the query in a single-label framework.

Tables 2 and 3 present the average values of accuracy, precision, recall and F1 measure for various groups of images in the Corel1000 and Corel5000 databases. Here, we used VQCC [36] as color feature and EGWC [35] as texture feature. Tables 2 and 3 demonstrate that the proposed MLC-FWC method provides better results when the EGWC and VQCC features are fused.

To evaluate our proposed method in comparison with other algorithms, the average precision results of the MLC-FWC algorithm and various state-of-the-art methods are shown in Table 4. In these algorithms [41–48], features are weighted or fused in a single-label image retrieval framework to retrieve more similar images and improve their result by feature weighting. All of these algorithms evaluated their methods on the Corel1000 dataset.

In Table 5, we also present the average precision and F1 measure results of our algorithm in comparison with some new multi-label classification algorithms [24, 29] on the Scene multi-label dataset with its original features within

Table 4 Average precision of some new feature combination (FW and FF) methods in comparison with MLC-FWC with $k=3$ on Corel1000 dataset within single-label image retrieval framework

Authors	Method	P_{avg} (%)
Puviarasan et al. [43]	Euclidean distance	63.4
	Hausdorff distance	66.9
Singha and Hemachandran [44]	WBCH	76.2
	Ch	74.2
Lin et al. [45]	–	72.6
Raghupathi et al. [46]	–	70.4
Hiremath and Pujari [47]	–	58.5
Rao et al. [48]	–	70.48
Guldogan and Gabbouj [41]	FSRL	66
Huang et al. [42]	Combined (CMR + GTF)	61
Li et al. [49]	CLHBP	78.50
Ali et al. [50]	–	74.80
Ali et al. [51]	WA SIFT-SURF	70.58
Montazer et al. [52]	RBFNN	70.31
Tian et al. [53]	Color SIFT	72.67
Abrishami Moghaddam and Ghodratnama [54]	MLFF-DS	82.64
Proposed method	MLC-FWC with single label	78.56

Table 5 Average precision of some new multi-label algorithms in comparison with MLC-FWC with $k=3$ on scene dataset and within multi-label classification framework

Authors	Method	AP (%)	F1 (%)	
Zhou et al. [29]	MimlBoost	77.9	55.6	
	MimlSvm	76.5	64.4	
	MimlSvmmi	78.3	67.1	
	MimlNn	77.1	61.3	
	AdtBoost.MH	71.8	N/A	
	RankSvm	74.6	62	
	MISvm	71.2	13.2	
	ML-KNN	75.9	52.9	
	Pedrycz and Yu [24]	BR	71	57.3
		RAkELd	72.9	57.5
BPMLL		67.4	48.8	
ML-KNN		86.6	66.1	
BRKNN		84.9	62.7	
ECC		84.9	65.3	
MLRS		85.2	73.3	
Wu et al. [55]	MLRS-LC	82.5	72.0	
	MMIB	–	72.48	
Abrishami Moghaddam and Ghodratnama [54]	MLFF-DS	87.78	78.63	
Proposed method	MLC-FWC	81.61	79.76	

Table 6 Average precision for each category of EGWC and VQCC in comparison with MLC-FWC on Corel1000 dataset within multi-label image retrieval framework

Category	EGWC [35]	Methods		
		VQCC [36]	MLFF-DS [54]	MLC-FWC
Africans	42.20	65.6	75.37	71.70
Beaches	40.10	35.5	41.46	55.80
Buildings	41.60	56.10	85.20	73.51
Buses	85.30	62.00	82.20	87.55
Dinosaurs	95.20	95.70	97.50	100
Elephants	64.20	66.70	89.60	93.00
Flowers	72.10	71.30	86.30	89.82
Horses	74.20	91.20	91.60	97.00
Mountains	27.10	38.80	60.30	60.00
Foods	57.20	78.80	82.00	80.50
P_{avg}	59.92	66.10	79.15	80.89

a multi-label classification framework. All the results are reported directly from their corresponding papers.

In Table 6, we evaluate the performance of the MLC-FWC algorithm in the retrieval framework. In the current implementation of MLC-FWC, the dataset images are indexed separately with the EGWC [35] and VQCC [36] algorithms. Therefore, we compared the results of the MLC-FWC with retrieval results of these two individual algorithms in terms of average precision (Table 6).

5 Conclusion

In this paper, we present a novel approach for image retrieval in a multi-label framework. The idea of this approach is to make a framework using clustering and classification together to reveal the structure of data and divide the input space into meaningful parts. Also, we fuse features by feature weighting learning algorithm to improve the retrieval results by purposefully using the ability of each feature to separate data.

Compared to the majority of papers that use deep learning architectures to perform CBIR [4–10], the proposed method simplifies the retrieval process and decreases the problem of “semantic gap,” by finding the labels of the query in the multi-label framework before doing retrieval, and also improves the performance through feature weighting. The main advantages of using a deep neural network are providing high accuracy and reducing the efforts in handcrafted feature design. The deep neural network automatically enables feature learning from raw data [10]. Nonetheless, it requires a huge computing power to train, so it is a costly and time-consuming process. It also suffers from difficulty in configuration and has not interpretable results. So if the

features are ready to use with a satisfactory performance, it is not valuable to use that as a training model.

Experimental results on the Corel1000 and Corel5000 datasets demonstrate a significant supremacy of the proposed CBIR system. The results are evaluated by four statistical measures, all of which confirm the significant improvement in the proposed framework.

References

1. Irawan C, Listyaningsih W, Sari CA, Rachmawanto EH (2018) CBIR for herbs root using color histogram and GLCM based on K-nearest neighbor. In: 2018 International seminar on application for technology of information and communication. IEEE, pp 509–514
2. Singh S, Rajput ER (2015) Content based image retrieval using SVM, NN and KNN classification. *Int J Adv Res Comput Commun Eng* 4(6):549–552
3. Dharani T, Aroquiaraj IL (2013) Content based image retrieval system using feature classification with modified KNN algorithm. arXiv preprint [arXiv:1307.4717](https://arxiv.org/abs/1307.4717)
4. Zhou W, Li H, Tian Q (2017) Recent advance in content-based image retrieval: a literature survey. arXiv preprint [arXiv:1706.06064](https://arxiv.org/abs/1706.06064)
5. Tzelepi M, Tefas A (2018) Deep convolutional learning for content based image retrieval. *Neurocomputing* 275:2467–2478
6. Wan J, Wang D, Hoi SCH, Wu P, Zhu J, Zhang Y, Li J (2014). Deep learning for content-based image retrieval: a comprehensive study. In: Proceedings of the 22nd ACM international conference on multimedia. ACM, pp 157–166
7. Yu J, Yang X, Gao F, Tao D (2016) Deep multimodal distance metric learning using click constraints for image ranking. *IEEE Trans Cybern* 47(12):4014–4024
8. Yu J, Tao D, Wang M, Rui Y (2014) Learning to rank using user clicks and visual features for image retrieval. *IEEE Trans Cybern* 45(4):767–779
9. Qayyum A, Anwar SM, Awais M, Majid M (2017) Medical image retrieval using deep convolutional neural network. *Neurocomputing* 266:8–20
10. Sadeghi-Tehran P, Angelov P, Virlet N, Hawkesford MJ (2019) Scalable database indexing and fast image retrieval based on deep learning and hierarchically nested structure applied to remote sensing and plant biology. *J Imaging* 5(3):33
11. Sarwar A, Mehmood Z, Saba T, Qazi KA, Adnan A, Jamal H (2019) A novel method for content-based image retrieval to improve the effectiveness of the bag-of-words model using a support vector machine. *J Inf Sci* 45(1):117–135
12. Tsai CF (2012) Bag-of-words representation in image annotation: a review. *ISRN Artif Intell* 2:1–19
13. Xu D, Yan S, Tao D, Lin S, Zhang HJ (2007) Marginal fisher analysis and its variants for human gait recognition and content-based image retrieval. *IEEE Trans Image Process* 16(11):2811–2821
14. Da Silva SF, Avalhais LP, Batista MA, Barcelos CA, Traina AJ (2014) Findings on ranking evaluation functions for feature weighting in image retrieval. *J Braz Comput Soc* 20(1):7
15. Chaturani NWUD, Geva S, Chandran V, Rajapaksha P (2016) Image retrieval based on multi-feature fusion for heterogeneous image databases. *Int J Comput Inf Eng* 10(10):1797–1802
16. Cordeiro De Amorim R, Mirkin B (2012) Minkowski metric, feature weighting and anomalous cluster initialisation in K-means clustering. *Pattern Recognit* 45(3):1061–1075

17. Modha DS, Spangler WS (2003) Feature weighting in k-means clustering. *Mach Learn* 52(3):217–237
18. Saha A, Das S (2015) Automated feature weighting in clustering with separable distances and inner product induced norms: a theoretical generalization. *Pattern Recognit Lett* 63:50–58
19. Chen X, Ye Y, Xu X, Huang JZ (2012) A feature group weighting method for subspace clustering of high-dimensional data. *Pattern Recognit* 45(1):434–446
20. Magesan E, Gambetta JM, Córcoles AD, Chow JM (2015) Machine learning for discriminating quantum measurement trajectories and improving readout. *Phys Rev Lett* 114(20):200501
21. Ghodrattama S, Boostani R (2015) An efficient strategy to handle complex datasets having multimodal distribution. In: *ISCS 2014: interdisciplinary symposium on complex systems*. Springer, Cham, pp 153–163
22. Tsoumakas G, Katakis I, Vlahavas I (2010) Random k-labelsets for multilabel classification. *IEEE Trans Knowl Data Eng* 23(7):1079–1089
23. Younes Z, Abdallah F, Dencœur T (2009) An evidence-theoretic k-nearest neighbor rule for multi-label classification. In: *International conference on scalable uncertainty management*. Springer, Berlin, pp 297–308
24. Yu Y, Pedrycz W, Miao D (2014) Multi-label classification by exploiting label correlations. *Expert Syst Appl* 41(6):2989–3004
25. Jiang JY, Tsai SC, Lee SJ (2012) FSKNN: multi-label text categorization based on fuzzy similarity and k nearest neighbors. *Expert Syst Appl* 39(3):2813–2821
26. Vens C, Struyf J, Schietgat L, Džeroski S, Blockeel H (2008) Decision trees for hierarchical multi-label classification. *Mach Learn* 73(2):185
27. Wang M, Zhou X, Chua TS (2008) Automatic image annotation via local multi-label classification. In: *Proceedings of the 2008 international conference on Content-based image and video retrieval*. ACM, pp 17–26
28. Lin Z, Ding G, Hu M, Wang J (2014) Multi-label classification via feature-aware implicit label space encoding. In: *International conference on machine learning*, pp 325–333
29. Zhou ZH, Zhang ML, Huang SJ, Li YF (2012) Multi-instance multi-label learning. *Artif Intell* 176(1):2291–2320
30. Duda RO, Hart PE, Stork DG (1973) *Pattern classification and scene analysis*, vol 3. Wiley, New York
31. Celebi ME, Kingravi HA, Vela PA (2013) A comparative study of efficient initialization methods for the k-means clustering algorithm. *Expert Syst Appl* 40(1):200–210
32. Paredes R, Vidal E (2006) Learning weighted metrics to minimize nearest-neighbor classification error. *IEEE Trans Pattern Anal Mach Intell* 7:1100–1110
33. Sharma A, Dey S (2012) A comparative study of feature selection and machine learning techniques for sentiment analysis. In: *Proceedings of the 2012 ACM research in applied computation symposium*. ACM, pp 1–7
34. Bouaguel W, Mufti GB, Limam M (2013) A fusion approach based on wrapper and filter feature selection methods using majority vote and feature weighting. In *2013 International conference on computer applications technology (ICCAT)*. IEEE, pp 1–6
35. Moghaddam HA, Dehaji MN (2013) Enhanced Gabor wavelet correlogram feature for image indexing and retrieval. *Pattern Anal Appl* 16(2):163–177
36. Shad SM (2011). *Color image indexing and retrieval using wavelet correlogram*. M.Sc. thesis in artificial intelligence and robotics, faculty of computer engineering, K.N. Toosi University of Technology, Tehran, Iran
37. Huang J, Ravi Kumar S, Mitra M, Zhu WJ, Zabih R (1997) Image indexing using color correlograms. *Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit* 1:762–768
38. Boutell MR, Luo J, Shen X, Brown CM (2004) Learning multi-label scene classification. *Pattern Recognit* 37(9):1757–1771
39. Tsoumakas G, Katakis I (2007) Multi-label classification: an overview. *Int J Data Warehous Min* 3(3):1–13
40. Yang Y (1999) An evaluation of statistical approaches to text categorization. *Inf Retrieval* 1(1–2):69–90
41. Guldogan E, Gabbouj M (2008) Feature selection for content-based image retrieval. *SIViP* 2(3):241–250
42. Huang ZC, Chan PP, Ng WW, Yeung DS (2010) Content-based image retrieval using color moment and Gabor texture feature. In: *2010 International conference on machine learning and cybernetics*, vol 2. IEEE, pp 719–724
43. Puviarasan N, Bhavani R, Vasanthi A (2014) Image retrieval using combination of texture and shape features. *Int J Adv Res Comput Commun Eng* 3(3)
44. Singha M, Hemachandran K (2012) Content based image retrieval using color and texture. *Signal Image Process* 3(1):39–57
45. Lin CH, Chen RT, Chan YK (2009) A smart content-based image retrieval system based on color and texture feature. *Image Vis Comput* 27(6):658–665
46. Raghupathi G, Anand RS, Dewal ML (2010) Color and texture features for content based image retrieval. In: *Second international conference on multimedia and content based image retrieval*
47. Hiremath PS, Pujari J (2007) Content based image retrieval based on color, texture and shape features using image and its complement. *Int J Comput Sci Secur* 1(4):25–35
48. Rao MB, Rao BP, Govardhan A (2011) CTDCIRS: content based image retrieval system based on dominant color and texture features. *Int J Comput Appl* 18(6):40–46
49. Li L, Feng L, Yu L, Wu J, Liu S (2016) Fusion framework for color image retrieval based on bag-of-words model and color local Haar binary patterns. *J Electron Imaging* 25(2):023022
50. Ali N, Mazhar DA, Iqbal Z, Ashraf R, Ahmed J, Khan FZ (2017) Content-based image retrieval based on late fusion of binary and local descriptors. *arXiv preprint arXiv:1703.08492*
51. Ali N, Bajwa KB, Sablatnig R, Chatzichristofis SA, Iqbal Z, Rashid M, Habib HA (2016) A novel image retrieval based on visual words integration of SIFT and SURF. *PLoS ONE* 11(6):e0157428
52. Montazer GA, Giveki D (2015) An improved radial basis function neural network for object image retrieval. *Neurocomputing* 168:221–233
53. Tian X, Jiao L, Liu X, Zhang X (2014) Feature integration of EODH and Color-SIFT: application to image retrieval based on codebook. *Sig Process Image Commun* 29(4):530–545
54. Moghaddam HA, Ghodrattama S (2017) Toward semantic content-based image retrieval using Dempster-Shafer theory in multi-label classification framework. *Int J Multimedia Inf Retr* 6(4):317–326
55. Wu B, Lyu S, Ghanem B (2016) Constrained submodular minimization for missing labels and class imbalance in multi-label learning. In: *Thirtieth AAAI conference on artificial intelligence*

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.