Available online at www.ijpe-online.com

International Journal of Performability Engineering

Vol. 13, No. 4, July 2017, pp. 519-529
DOI: 10.23940/ijpe.17.04.p19.519529

A New Multiple Instance Learning Algorithm based on Instance-Consistency

Zhize Wu*, Miao Zhang, Shouhong Wan, Lihua Yue

School of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui 233027, P. R. China

Abstract

Multiple-instance learning (MIL) has been successfully utilized in image retrieval. Existing approaches cannot select positive instances correctly from positive bags, which may result in low accuracy. Inspired by the characteristic that consistencies are always among instances and instances, and bags and instances, we propose a new algorithm called multiple instance learning based on instance-consistency (MILIC) to mitigate this issue. First, we select potential positive instances effectively in every positive bag through the minimum cost of instance-consistency. Second, we use the L1-LR to select irrelevant instances from potential positive instances to further improve the retrieval efficiency. Then, we design a novel feature representation scheme based on the irrelevant potential positive instances to convert a bag into a single instance. Band on the feature representations, we finally conduct object-based image retrieval and image categorization by adopting the standard single-instance learning (SIL) strategy, such as the support vector machine (SVM), to verify the effectiveness of our proposal.

Keywords: Multiple instance learning; Image retrieval; Image categorization; Instance consistency; Feature representation;

(Submitted on March 1, 2017; Revised on April 27, 2017; Accepted on June 15, 2017)

1. Introduction

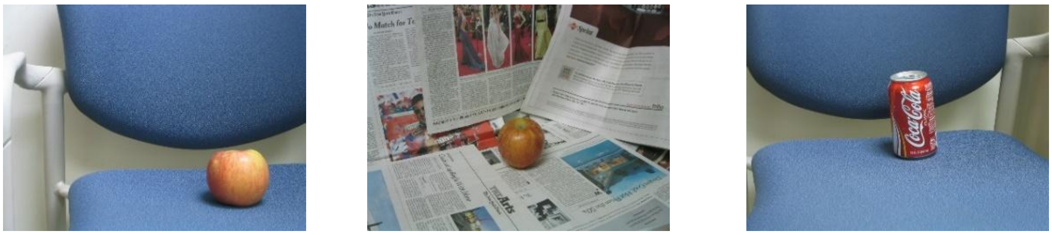
Content-based image retrieval (CBIR) aims to avoid textual descriptions of images and instead retrieve images based on similarities of their contents (textures, colors, shapes etc.) to user-supplied query images or user-specified image features. It has been one on the most vivid research areas in the field of computer vision due to the availability of large and steadily growing amounts of visual and multimedia data, along with the development of the Internet. Traditional studies [1,2] about image retrieval find query results by extracting and comparing their global or local features, like color histograms, color moment, etc., with query images. These methods work very well when the major content of a query image is the interested object. They have also been widely used in traditional CBIR systems. However, when an object occupies a small part in the query image, these approaches include little useful information and the object's features are overridden by the background. What's more, these approaches cannot locate objects which users are interested in.

As shown in Figure 1, the interested object is an apple, but the major part of (a) is a chair, where the chair occupies most of its features and the apple occupies little of its features. We notice that (a) is more similar with (c) than (b) by comparing their extracted features in Figure 1.

Therefore, we can mitigate the influence of background on the object by dividing an image into several instances,

* Corresponding author. Tel.: +86-15156899179.
E-mail address: wuzhize.ustc@gmail.com.

which only interests a portion of an image and is a way of content-based image retrieval (CBIR), that is, object-based image retrieval (OBIR) [2].



(a) An apple in a chair (b) An apple in newspapers (c) A coke-cola in a chair

Figure 1. Problem illustration, the interested object is an apple.

Multiple-instance learning (MIL) has been successfully utilized to address OBIR, where a bag corresponds to an image and an instance corresponds to a region of an image. Under the standard MIL framework, a positive bag contains at least a positive instance, and a negative is only composed of negative instances. As shown in Figure 1, we consider (a) and (b) are the positive bags, and (c) is the negative bag when the interested object is an apple. It should be noted that there will be at least one positive bag and at least one negative bag, and only bags have been labeled while instances haven't in the MIL based image retrieval setting.

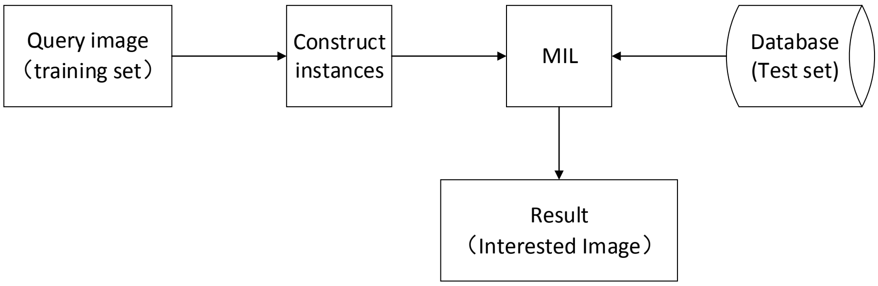


Figure 2. Retrieval process based the MIL

The basic process of the MIL based image retrieval is illustrated in Figure 2. It uses existing segmentation algorithms to segment the images into several regions and sees regions as instances. By measuring the similarities between the query images and database using the MIL, it selects the most similar image as an interested one. Through this process, we can broadly categorize the existing MIL based image retrieval approaches into three main groups:

1) Interested point based approaches [2-4]: These approaches first find an interested point that likely is the center of positive instances, and estimate whether each bag has an instance that is close to the center. In [2], Zhang Q *et al.* are first to perform image retrieval with the MIL method. They simply use the EM-DD to find the interested point, which likely looses some important information when the interested object is divided into several instances. Instead of only considering the feature of the instance itself, Rahmani R *et al.* [3] represents the instance combining itself and its NESW (north, east, south, west) four neighbors. This method performs well when the neighbors of the positive instance are also positive. Otherwise, it will yield a poor performance. Similar to this, Xu YY *et al.* [4] propose a multiple-instance learning based decision neural network, which improves the query accuracy. But, this approach still shares the above weakness, which does not express the relationship of bags and instances.

2) Graph based approaches [5-7]: These approaches estimate whether a bag is positive or not based on the graph model. Rahmani R *et al.* [5] use the bags of training set and test set to construct a graph considering both multiple-instance and semi- supervised properties. Wang C *et al.* [6] use the instances of the labeled set and unlabeled set to construct a graph. Li F *et al.* [7] use the bags and the instances to construct graphs separately. All of these methods consider a bag as positive only if it has a positive instance. Although the graph model can consider the consistency among instances and instances, it doesn't consider the relationship of bags and instances. That is, when the interested object is divided into several regions, the credibility of the query results is low.

3) Space mapping based approaches [8-10]: These approaches consider feature representation schemes to map each bag into an instance space and convert MIL into SIL. In [8], Chen Y *et al.* proposed the approach called DD-SVM, which trained a SVM in feature space constructed from a mapping defined by the local maximizers and minimizers of the DD function. But, this approach doesn't consider instance-consistency and the relationship between instances

and bags. Fan R E *et al.* [9] mapped each bag into a feature space defined by the instances in the training bags using an instance similarity measure. Although this method is more efficient than the DD-SVM, the feature space for representing bags is of high dimension because it contains too many unnecessary features. Li W J *et al.* [10] selected 5 instances from every positive bag while considering the relationship of bags and instances. However, it does not take full advantage of the relationship of instances from different bags, which is consistency among instances, leading to an unsatisfactory accuracy of positive instances selecting.

The aforementioned approaches do not simultaneously take the relationship of instances from different bags and the relationship of bags and instances into account. We argue that these two relationships are very important for the MIL based image retrieval. As space mapping based approaches can express the relationship of bags and instances, in this paper, we use this method and express the relationship of instances from different bags using instance-consistency. We propose a new approach called multiple instance learning based on instance-consistency (MILIC) that takes advantage of the consistencies among instances and instances, and bags and instances. MILIC better selects positive instances using the consideration of the ranking cost of instance-consistency. Then, it selects irrelevant instances with L1-LR to reduce the amount of potential positive instances. Based on the selected irrelevant instances, it can convert MIL to SIL. Finally, it uses the SVM to select interested images.

The main contributions of our proposal are summarized as follows:

1) Propose a potential positive instance-selected method based on the relationship of instances from different bags (instance-consistency).
2) Use the L1-LR to select the irrelevant potential positive instances to reduce the feature dimensions and convert MIL into SIL with the proposed feature mapping algorithm.
3) Compare our method with state-of-the-art methods on two challenging data sets to demonstrate the promising performance of our method with respect to multiple performance metrics, including accuracy and efficiency.

The remainder of this paper is organized as followed. In Section 2, we present the MILIC algorithm. Experimental results on real world data are presented in Section 3. Finally, conclusion and future work are displayed in Section 4.

2. Multi-instance Learning based on Instance-consistency

The framework of the MILIC contains two parts: the training and the retrieval, as shown in Figure 3. During the training phrase, the MILIC first selects potential positive instances and maps each bag into the bag feature space. Through this, the MIL is converted to a normal problem of supervise learning. During the retrieval phrase, the MILIC transforms each bag in the dataset into an element in the bag feature space based on the acquired potential positive instances and scores the bag based on the learned supervise learning model, such as the SVM.

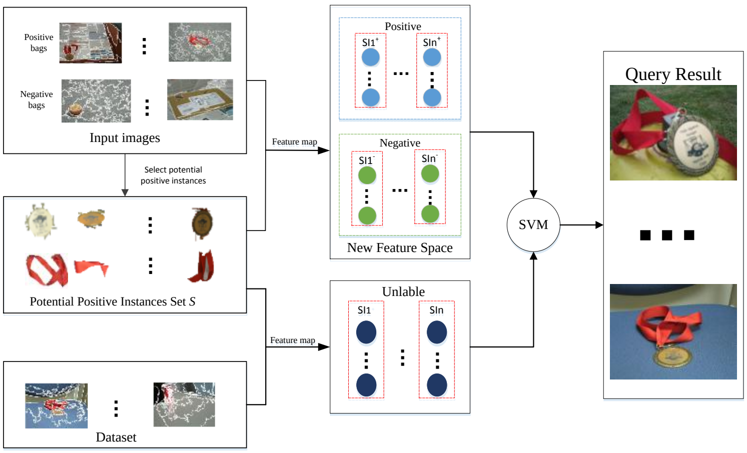


Figure 3. Framework of the MILIC

In Section 2.1, we use the defined cost function based instance-consistency to get potential positive instance set S. With that, we propose a method of the feature mapping to get the new features of the input bags and dataset in Section 2.2. Finally, we can use the SVM to get a classifier and get the query result.

2.1. Potential positive instances selection

A positive bag $P(b^+_{gh},b_{ij})=e^{-dist(b^+_{gh},b_{ij})}$. is represented as a sequence of instances $\left\{b^+_{i1},\ldots,b^+_{in_i^+}\right\}$, where b_{ij} is an instance of the positive bag, n_i^+ is the amount of instances of $P(b^+_{gh},b_{ij})=e^{-dist(b^+_{gh},b_{ij})}$.. Meanwhile, a negative bag B_i^- is represented as a sequence of negative instances $P(b^+_{gh},b_{ij})=e^{-dist(b^+_{gh},b_{ij})}$., where b_{ij}^- is an instance $P(b^+_{gh},b_{ij})=e^{-dist(b^+_{gh},b_{ij})}$.of the negative bag, n_i^- is the amount of instances of B_i^-. We here label the positive bag and the negative one 1 and -1, B_i^+ respectively. And we denote a training set with $\left\{B_1^+,\cdots B_{N^+}^+,B_1^-,\cdots B_{N^-}^-\right\}$, where N^+ is the amount of the positive bags, N^- is the amount of the negative bags.

Based on the Central Limit Theorem, we suppose the consistent probability between two instances follows the Normal Distribution when regarding the distance of two instances as the independent variable. We describe this relationship as follows:

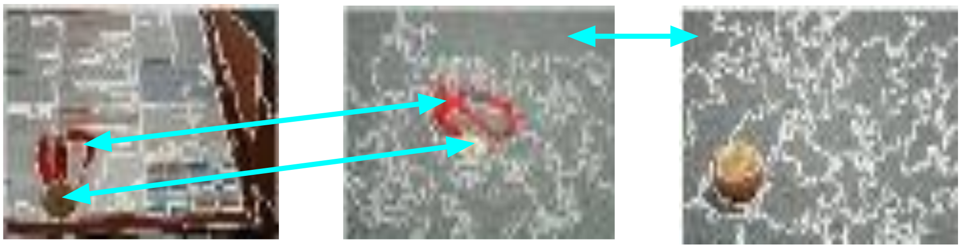
$$P(b^+_{gh},b_{ij})=e^{-dist(b^+_{gh},b_{ij})}. \tag{2.1}$$

Here, $dist(b^+_{gh},b_{ij})$ is defined as follows:

$$dist(b^+_{gh},b_{ij})=\sum_k (b_{ghk}-b_{ijk})^2, \tag{2.2}$$

where k ranges over all the features, b_{ghk} and b_{ijk} refer to the kth features of the corresponding instance. We define the consistent probability between an instance b_{gh} and a bag B_i as the probability of B_i has a consistent instance of b^+_{gh} . Intuitively, the probability between an instance b^+_{gh} and a bag B_i is equal to the maximal consistency probability between instances in B_i and b^+_{gh} , which is defined as followed:

$$P(\mathrm{b}^+_{gh},B_i)=\max_j P(b^+_{gh},b_{ij}). \tag{2.3}$$



(a) Positive bag (b) Positive bag (c) Negative bag

Figure 4. Consistency among instances and the interested object is a gold-medal. There is consistency among instances at the ends of arrow.

It should be noted that an instance in a positive bag is probably positive and an instance in a negative bag is definitely negative. When an instance is positive and it is in a positive bag, there will be a consistent instance in every positive bag and no consistent instances in all negative bags. In Figure 4, we notice the consistency between the positive instances in (a) and one in (b). There probably exists a consistency among negative instances from a positive bag and a negative bag like (b) and (c). Inspired by this observation and the IF-IDF, we define a novel function, called IC function, which can describe the importance of an instance b^+_{gh} on selecting the optional instances, which is defined as followed:

$$IC(b^+_{gh})=\prod_i^{N^+}P(b^+_{gh},B_i^+)\Bigg/\prod_i^{N^-}P(b^+_{gh},B_i^-). \tag{2.4}$$

In the IC function, $IC(b^+_{gh})$ will be high when there is a consistent instance of b^+_{gh} in B_i^+. On the contrary, it will be low when there is a consistent instance of b^+_{gh} in B_i^- . So, it can describe the importance of an instance in retrieval and the probability of an instance is a positive instance. The probability of an instance is a positive instance is high when its IC value is high. In this paper, the amount of the potential positive instances in each positive bag is determined artificially. The procedure is summarized as Algorithm 1.

Algorithm 1. Potential positive instances selection for the MIL

```
Input: $\left\{B_1^+, B_2^+, \cdots, B_{N^+}^+, B_1^-, B_2^-, \cdots, B_{N^-}^-\right\}$;
Output: potential positive instances set $S$;
Initialize potential positive instances set $S=\varphi$;
For $i$=1 to $N^+$ do
    For j=1 to $n_i^+$ do
        Computer $IC(b_{ij}^+)$
    End for
    Sort the instances in $B_i^+$ in ascending order of their IC values;
    Select top K instances into set S;
End for
Return $S$
```

Differing from the EC function proposed in EC-SVM [12], the defined IC function only needs to take into account the similar instances in each bag when computing the importance of an instance. It is helpful for selecting potential positive instances by increasing the influence of the positive instances in the positive bag. However, the EC function accounts for all of the instances in the bag, which yields poor performance in the positive instance selection because of the influence of the negative instances.

2.2. Feature mapping

Based on the selected potential positive instances, we propose a novel feature mapping that maps each bag B_i to a single instance, which has p features:

$$\varphi(B_i, S) = [s(b_1, B_i), s(b_2, B_i), ..., s(b_p, B_i)], \tag{2.5}$$

where b_k is a potential positive instance and p is the amount of the selected instances. $s(b_k, B_i)$ is defined as follows:

$$s(b_k, B_i) = \max_j(s(b_k, b_{ij})), \tag{2.6}$$

which means that the similarity of an instance and a bag is simply equal to the most similar between the selected instance and each instance in the bag, and the similarity of two instances is defined as follow:

$$s(b_k, b_{ij}) = e^{-dist(b_k, b_{ij})}, \tag{2.7}$$

which means the similarity of two instances inverses their distance.

The k-th feature of $\varphi(B_i)$ can be interpreted as the probability of a consistent instance in B_i with the potential positive instance b_k and it ranges from 0 to 1, which is very meaningful because $\varphi(B_i)$ can measure the similarity with all positive instances, not just one positive instance. Thus, based on this feature mapping, we can finely describe the relationship between instances and bags and confirm whether the bag has all interested regions when the interested object is divided into several regions.

2.3. Classification

After the feature mapping, the MIL is converted to a normal problem of supervise learning. Therefore, we can use a supervised model with some specific constraints to perform classification. Because the retrieval result is the score obtained by sorting all of the bags, the outputs are not only -1 and 1, but also continuous values. On the other hand, in order to discriminate and sort the scores, we should make the differences of the scores as large as possible. Finally, the classifier needs to be satisfied with a minor number of samples that can also achieve satisfactory results.

The SVM can obtain excellent results on small sample data and score each bag in the interval of $(-\infty, \infty)$. Based on a proper kernel function, it can also fit any curve. Considering the above factors, we use the SVM to perform such classification, which is similar to some other feature mapping algorithms [9, 12]. The kernel function we use here is the RBF kernel.

2.4. Algorithm of MILIC

The MILIC training steps are summarized in Algorithm 2. In the training phase we first get potential positive instances set S from positive bags; Then, we map the training bags to instances by $\varphi(B_i, S)$. Finally, we train the instance using the SVM to get the classifier. In the test phase, we map test bags to instances and use the trained classifier. According to the ranking results, we can finally get the top queries, which are the interested images.

Algorithm 2. MILIC for retrieval

```
Input: $\{B_1^+, B_2^+, \cdots, B_{N^+}^+, B_1^-, B_2^-, \cdots, B_{N^-}^-\}$;
Dataset: $\{B_1^u, B_2^u, \cdots, B_{N^u}^u\}$;
Output: interested images;
1. Get potential positive instances set S from every positive bags as shown in Algorithm 1.
2. Map the input bags to instances by $\varphi(B_i, S)$ in Section 2.2.
3. Train SVM to get a classification model.
4. Map the dataset to instances by $\varphi(B_i^u, S)$ in Section 2.2.
5. Use the model to get scores for every bag in dataset and rank bags to get the top images
   as the interested images.
```

3. Experimental Results and Analysis

In this section, we carry out the experimental verification and compare our method to several other competitive approaches [2, 3, 8, 11, 12, 13] on two benchmarks including SIAL [2] and Core [9]. The Receiver Operating Characteristic (ROC) curve is insensitive to the ratio of positive and negative examples in the data repository. Therefore, we use the area under the ROC curve (AUC) [13] instead of the Precision and Recall to evaluate the experimental performance like they did in [2, 3, 9, 12]. The experimental protocol is the same as it is in [11]. In Section 3.1, we present the AUC of the proposed MIL framework MILES [11], EC-SVM [12], ACCIO! [2], MI-BDNN [3], MGMIL [8] and our defined MILIC on the SIVAL data set. In Section 3.2, we compare the AUC of MILIC with EC-SVM on the Corel set. We use LIBLINEAR [13] to train all of the L1-LR classifiers to get the irrelevant features and use LIBSVM [14] to train all of the SVM classifiers.

3.1. Evaluation on SIVAL Data Set

Accuracy Evaluation and Analysis: The SIVAL data set contains 1500 images of 25 categories with 60 images in each category. It is obtained by shooting 6 different images (angle, light, location, etc.) in a scene. There are 10 different scenes. These 10 scenes are very different and highly complex, and the target may appear anywhere in the image. In most images the object of interest is only 10% to 15% of the image, but there are a number of images interested in the object area of about 70% of the image area. Each interested object itself is also very complex and will be split into multiple instances.

All the images have been segmented using the Improved Hierarchical Segmentation algorithm (IHS) [15]. The HIS algorithm uses a bottom-up approach. It firstly regards each pixel as a separate area, and then uses the Euclidean algorithm to calculate the similarity of adjacent regions, selecting the most similar areas to merge until the number of areas reaches our specified requirements.

A bag is created for each image *I* with a point according to each segment $s \in I$. We run our algorithm in two-type features: one is represented by a 6 features hold that averages color and texture values for *s* and the other is represented by 30 features where the first 6 features is for *s* and the remaining dimensions hold 6 features for each of the cardinal neighbors *(N,E,S,W)* of *s*. These features hold the difference between the average color and texture values of *s* and its neighbors.
We conduct 30 independent runs for all the categories in the database. In each category, 8 positive and 8 negative images are randomly selected for training, and the remaining 1484 images form the test set. The parameter C for *L1-LR* in LIBLINEAR is set from 2^{-5} to 2^{5} to select the best parameter. We select 2^{-5} as the parameter. The parameter C and Gaussian kernel parameter *r* for SVM in LIBSVM are simply set to 1 and 2^{-4}, respectively.

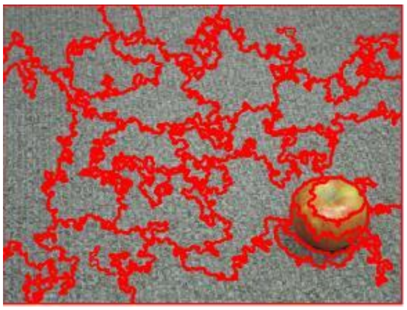
We compare the performance of MILIC with several related and classic type methods, where EC-SVM and MILES are classic type SVM-based multiple learning algorithms, and MGMIL [16] is a classic type graph-based multiple learning algorithm. The ACCIO! method first uses the SIVAL data set. MI-BDNN is the latest multi-learning retrieval algorithm. We list results of the IS-SVM and EC-SVM of the 6 features and all of the 30 features. All of those algorithms except for our algorithm run better with 30 features than 6 features.

Table 1 shows that the overall performance of our proposal is equal to that of the MGMIL [16] and better than those of all the other methods. But the MGMIL [16] uses the graph to solve the problem and create graphs of both instance level and bag level graph. Its time complexity and space complexity is very high. Suppose the number of bags is N. the number of all instance is n and the number of potential instances selected by us is k. The time complexity of MGMIL [16] is $O((n+N)^3)$ and the space complexity of [8] is $O(n^2+N^2)$. The time complexity of our methods all are $O(kn)$ and k is the number of selected instances. Our algorithm is better than [8] in terms of both time and space.

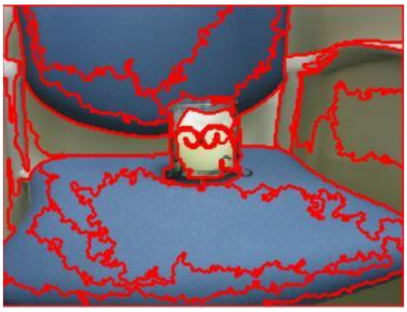
As shown in Table 1, we find that MILIC has an exceptionally better performance with 6 features than with 30 features in some categories like the Apples, Bananas, etc. and exceptionally better performance with 30 features than with 6 features in some other categories like the Candle-With-Holder, Cardboard-Box, etc. Observing the results of image segmentation, we find that the 6 features perform better than 30 features when there is not an interested region whose neighbors are all interested regions. Otherwise, the 30 features perform better than the 6 features. As shown in Figure 4, the interested object of (a) is the apple and all the neighbor instances of the interested instances are few of difference, and so the 6 features is better. On the other hand, the interested object of (b) is the candlestick and all the neighbor instances of the interested instances are very different, and so the 30 features is better. Actually, when the background instances are similar, the extra 24 features are noisy. When the background instance are extremely different, the performance of the 30 features is not far better than that of the 6 features, which is shown in categories such as the orange juice bottle, coke bottle, etc. However, the performance of the 6 features is much better than that of the 30 features when the segmented instances are similar. So, we select 6 features as our final instance expression. It should be noted that the MILIC improves by 5 percent with 6 features compared to the EC-SVM.

Table 1. Average AUC values (in percent) with 95% confidence interval over 30 rounds of test on the SIVAL image set. Unless stated otherwise, the results are reported based on 30 features.

Category	MILIC (6)	EC-SVM	MILIC	EC-SVM	MI-BDNN	MGMIL	MILES	ACCIO!
AjaxOrange	91.0 ±1.9	91.4 ±1.9	96.0 ±0.01	93.8 ±2.1	**97.6 ±0.49**	93.9 ±0.7	90.2 ±2.3	77.0 ±3.4
Apple	**82.6 ±2.1**	75.3 ±3.8	67.2 ±2.5	68.0 ±2.6	66.6 ±4.17	73.5 ±1.3	64.5 ±2.5	63.4 ±3.4
Banana	**82.7 ±2.2**	70.4 ±2.8	68.5 ±2.6	69.1 ±2.9	74.8 ±3.03	77.1 ±0.9	68.1 ±3.1	65.9 ±3.3
BlueScrunge	86.2 ±2.9	**86.9 ±2.1**	69.7 ±2.7	74.1 ±2.4	67.2 ±5.62	80.4 ±1.4	72.6 ±2.5	69.5 ±3.4
CandleWithHolder	77.8 ±2.2	76.0 ±2.6	**88.8 ±1.1**	88.1 ±1.1	68.7 ±3.61	76.0 ±1.3	84.0 ±2.3	68.8 ±2.3
CardboardBox	65.9 ±2.6	63.0 ±2.6	83.0 ±2.8	**85.6 ±1.6**	69.9 ±1.95	81.2 ±1.2	81.2 ±2.7	67.9 ±2.2
CheckeredScarf	95.3 ±1.1	93.5 ±1.3	97.0 ±0.3	96.9 ±0.5	**98.2 ±0.08**	89.3 ±0.9	93.7 ±1.2	90.8 ±1.6
CokeCan	92.0 ±1.1	87.3 ±1.9	95.3 ±0.7	94.6 ±0.8	**95.1 ±0.07**	89.4 ±1.4	92.4 ±0.8	81.5 ±3.5
DataMiningBook	85.9 ±2.3	81.0 ±3.5	80.4 ±2.5	75.0 ±2.4	65.6 ±2.86	**93.0 ±0.9**	71.1 ±3.2	74.7 ±3.4
DirtyRunningShoe	84.2 ±1.3	76.8 ±2.2	90.7 ±1.4	90.3 ±1.3	**91.7 ±0.61**	82.9 ±1.1	85.3 ±1.7	83.7 ±1.9
DirtyWorkGloves	67.4 ±1.6	66.0 ±1.6	79.6 ±2.4	83.0 ±1.3	77.5 ±2.17	**80.1 ±1.2**	77.1 ±3.1	65.3 ±1.5
FabricSoftenerBox	93.1 ±1.1	92.1 ±1.6	**98.0 ±0.6**	97.9 ±0.5	94.8 ±1.01	95.8 ±0.7	97.1 ±0.7	86.6 ±3.0
FeltFlowerRug	88.4 ±1.6	86.6 ±1.6	95.5 ±1.0	94.2 ±0.8	**95.9 ±0.42**	89.2 ±1.2	93.9 ±0.7	86.9 ±1.7
GlazedWoodPot	**87.7 ±2.3**	63.3 ±3.3	72.1 ±2.8	68.0 ±2.8	73.9 ±3.42	74.9 ±1.1	68.2 ±3.1	72.7 ±2.3
GoldMedal	**91.1 ±1.2**	90.3 ±1.4	81.1 ±2.7	87.5 ±1.4	83.4 ±3.42	85.3 ±1.6	80.7 ±2.9	77.7 ±2.6
GreenTeaBox	95.6 ±1.2	86.1 ±2.3	92.8 ±1.8	86.9 ±2.2	**96.1 ±0.57**	90.4 ±0.9	91.2 ±1.7	87.3 ±3.0
JuliesPot	**90.7 ±1.3**	74.6 ±3.8	82.0 ±2.9	67.3 ±3.3	84.8 ±2.62	87.3 ±1.5	78.7 ±2.9	79.2 ±2.6
LargeSpoon	53.7 ±2.0	55.7 ±2.1	62.2 ±1.7	61.3 ±1.8	63.3 ±1.36	**73.8 ±1.3**	58.2 ±1.6	57.6 ±2.3
RapBook	61.7 ±5.1	60.7 ±1.9	70.0 ±2.5	68.6 ±2.3	73.2 ±1.66	**87.0 ±1.0**	61.7 ±2.4	62.8 ±1.7
SmileyFaceDoll	**92.1 ±1.2**	90.3 ±2.1	81.1 ±1.9	84.6 ±1.9	68 ±4.44	83.8 ±1.2	77.5 ±2.6	77.4 ±3.3
SpriteCan	**88.9 ±1.0**	73.9 ±2.9	86.2 ±2.1	85.4 ±1.2	82.1 ±2.44	79.7 ±1.4	80.4 ±2.0	71.9 ±2.5
StripedNoteBook	78.6 ±2.2	72.9 ±2.3	76.6 ±3.3	75.6 ±2.3	**88.5 ±1.75**	73.8 ±1.3	68.7 ±2.4	70.2 ±3.2
TranslucentBowl	**90.4 ±1.2**	85.3 ±2.9	76.4 ±2.2	74.2 ±3.2	83.8 ±3.08	79.3 ±1.7	73.2 ±3.1	77.5 ±2.3
WD40Can	91.9 ±1.0	92.3 ±0.8	**94.8 ±0.7**	94.3 ±0.6	80.5 ±3.58	92.3 ±1.0	88.1 ±2.2	82.0 ±2.4
WoodRollingPin	65.8 ±2.7	61.4 ±2.5	67.4 ±1.7	66.9 ±1.7	67 ±1.34	**70.3 ±1.2**	62.1 ±2.5	66.7 ±1.7
Average	**83.2**	78.1	82.1	81.3	80.3	**83.2**	78.4	74.6



(a) apple



(b) candle with holder

Figure 4. Images segmented by HIS. (a) The object is an apple and the east and west of all interested regions are background region. (b) The object is a candle-With-Holder and there are an interested region whose all neighbors are interested regions.

In Figure 5, we present the average performance over 25 categories of MILIC and EC-SVM when the number of labeled bags is varied. Here, all labeled bags are randomly selected and the number of positive bags and negative bags are equal. All remaining images are placed in the dataset. We can see that the performance of all algorithms is increasing when the labeled bags becomes larger. We see that we can get a higher accuracy when we get enough labeled bags. In addition, the performance of the MILIC is always better than that of the EC-SVM. The 30 features of our algorithm performs better than 6 features. The performance of the EC-SVM is just the opposite.

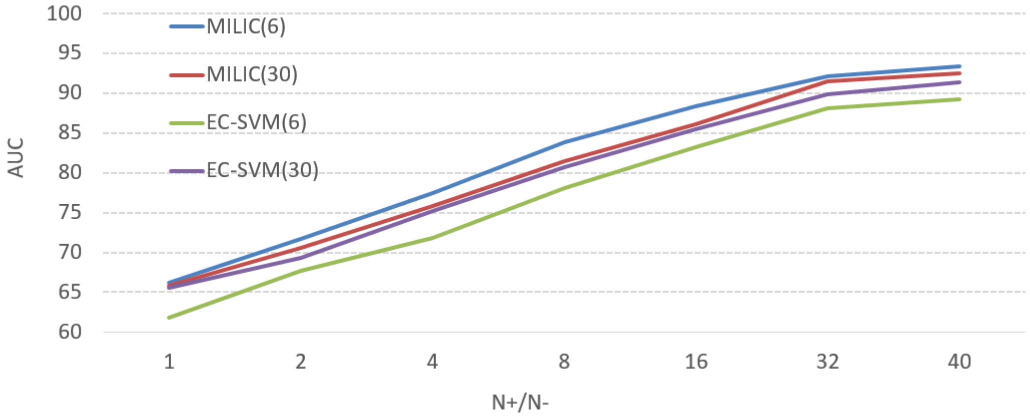


Figure 5. Average AUC values for all 25 categories over 30 rounds of different number of labeled bags on the SIVAL image set and number in brackets is amount of instances' features.

Sensitivity to the Number of Potential positive instances: Figure 6 shows the AUC variation when the value *K*, the number of the potential positive instances, is changed. For each value *K*, each class runs 50 times, and each run selects 4 positive packages and 4 negative packages. The remaining images form the database. In Figure 6, we can see that the mean AUC does not have a significant change when the value of *K* changes on 2 ~ 10, which proves the robustness of our algorithm to *K* value change. When *K* is 5 or 6, the AUC reaches the highest level.

Computation Cost: We conduct the experiment on a 3.30 GHz PC with 8G memory. In [12], the EC-SVM is better than the DD-SVM and MILES on run time, so we only compare with the EC-SVM. Table 2 lists the training time required by EC-SVM and MILIC. In this table, we can see that the run-time of the EC-SVM is double that of the MILIC. Because we use a 6-dimensional feature, and the EC-SVM is a 30-dimensional feature, the MILIC is faster than the EC-SVM when calculating distance. The efficiency of the EC-SVM and the number of selected feature examples are related. Therefore, the MILIC is four times faster than the EC-SVM in feature mapping. However, the efficiency of their approach is similar when calculating the efficiency of the classifier model and the final calculations of similarity values. On the whole, the run-time of the EC-SVM is not four times that of the MILIC.

Table 2 Computation time comparison (in second)

MILIC	EC-SVM
2.43±0.045	4.98±0.045

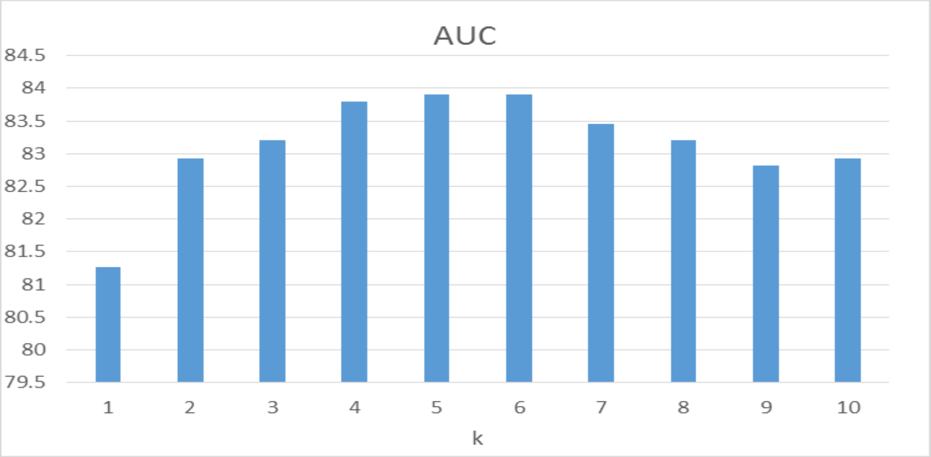


Figure 6. Average AUC values for all 25 categories over 30 rounds on the SIVAL data set against K.

3.2. Evaluation on Corel Data Set

The COREL 2000 dataset contains 20 images, 20 classes, and 100 images per category. Each category represents an obvious theme, including villages, sandy beaches, historic buildings, buses, dinosaurs, elephants, flowers, horses, mountains and glaciers, food, dogs, lizards, fashion models, sunset scenes, cars, waterfalls, classical furniture, battleships, skiing, desserts, and so on.

Similar to the EC-SVM, we chose 2000 images from the 20 categories. Each category contains 100 images. We use the same image segmentation and feature representation with the EC-SVM to construct the corresponding bags and instances. The K is set to 6. The parameter C and Gaussian kernel parameter r for SVM in LIBSVM are simply set to 1 and 2^{-3} respectively, similar to that of EC-SVM. For each category, we use the “one-versus–the rest” strategy to evaluate the performance. In each round, 4 randomly selected positive images and 4 randomly selected negative images are chosen to form the training and the remaining 1992 images form the test set. The results are reported based on 50 rounds of independent testing. Table 3 lists the results of the average AUC values (in percent) for all categories with 95% confidence interval over 50 rounds of testing. Once again, the MILIC achieves better results than the EC-SVM.

Table 3. Average AUC values (in percent) for all 20 categories with 95% confidence interval over 50 rounds of test on the COREL image set.

MILIC	EC-SVM
84.2 ±0.2	83.2 ±0.3

We further study the sensitivity to the number of the selected potential positive instances, K. Figure 7 illustrates the change in the average performance over 20 categories and when K is varied on Corel dataset. We can also find that there is no significant difference among the performances when K ranges from 2 to 10 in two datasets. It further proves that our method is robust to the variation of the number of potential positive instances.

3.3. Experiment on Image Categorization

DD-SVM, MILES, MILIS and EC-SVM are all the space mapping based approaches, so we perform comparisons with them on image categorization. As in Section 3.2, the K is set to 6, the parameter C and Gaussian kernel parameter r for the SVM in LIBSVM are simply set to 1 and 2^{-3}, respectively. Similar to the MILES, for each category, we randomly select 50 images from this category as positive bags and select 50 images from other categories as negative bags to form the training set. The remaining images form the test set. The average classification accuracies are over five random test sets in Table 4. We can see that the MILIC achieves the best performance. It shows that the MILIC is not only suitable to the image retrieval, but also the image categorization application.

Table 4. Comparing the Image Categorization accuracy obtained using MILIC with other methods on Corel-2000 dataset

Algorithm	MILIC	MILIS	EC-SVM	MILES	DD-SVM
Accuracy	87.10 ±2.0	70.1 ±1.7	85.96 ±2.0	68.7 ±1.4	67.5 ±1.4

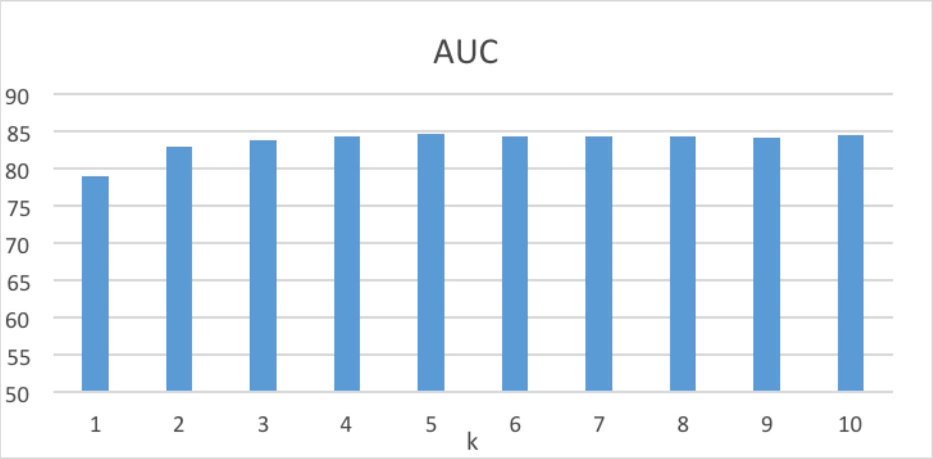


Figure 7. Average AUC values for all 20 categories over 50 rounds on the Corel data set against K.

4. Conclusion and Future Work

In this paper, we propose an efficient object-based image retrieval approach based on the multiple instance learning framework (MIL). Previous works on MIL based image retrieval usually do not consider full use of the relationship of the instances and the relationship of the instances and bags. They have poor performance on the query accuracy. Instance-consistency information is first studied by the proposed IC function. It can offer meaningful features for selecting the potential positive instance. Then, we use the L1-LR to select irrelevant instances from potential positive instances and design the feature representation scheme to convert a bag into a single instance. Experiments using the SIVAL dataset and the Corel dataset show that our proposal achieves superior results in terms of its query accuracy and run time, and it is robust to the variation of the number of the selected potential positive instances. It also gets a pretty good result when conducting image categorization on the Corel-2000 dataset.

Although very promising performances of both AUC and rum-time have been achieved by our method, we do not consider the spatial relation of instances. Hence, we will focus on how to consider the spatial relation in our future work. Furthermore, we are shaping the characteristic, combining the color and texture in features.

References

1. Smeulders, Arnold WM, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. "Content-based image retrieval at the end of the early years." *IEEE Transactions on pattern analysis and machine intelligence* 22, no. 12 (2000): 1349-1380.
2. Rahmani, Rouhollah, Sally A. Goldman, Hui Zhang, John Krettek, and Jason E. Fritts. "Localized content based image retrieval." In *Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval*, pp. 227-236. ACM, 2005.
3. Zhang, Qi, Sally A. Goldman, Wei Yu, and Jason E. Fritts. "Content-based image retrieval using multiple-instance learning." In *ICML*, vol. 2, pp. 682-689. 2002.
4. Xu, Yeong-Yuh. "Multiple-instance learning based decision neural networks for image retrieval and classification." *Neurocomputing* 171 (2016): 826-836.
5. Wang, Changhu, Lei Zhang, and Hong-Jiang Zhang. "Graph-based multiple-instance learning for object-based image retrieval." In *Proceedings of the 1st ACM international conference on Multimedia information retrieval*, pp. 156-163. ACM, 2008.
6. Rahmani, Rouhollah, and Sally A. Goldman. "MISSL: Multiple-instance semi-supervised learning." In *Proceedings of the 23rd international conference on Machine learning*, pp. 705-712. ACM, 2006.
7. Li, Fei, and Rujie Liu. "Graph-based multiple-instance learning with instance weighting for image retrieval." In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pp. 2453-2456. IEEE, 2011.
8. Li, Fei, and Rujie Liu. "Multi-graph multi-instance learning for object-based image and video retrieval." In *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval*, p. 35. ACM, 2012.
9. Chen, Yixin, and James Z. Wang. "Image categorization by learning and reasoning with regions." *Journal of Machine Learning Research* 5, no. Aug (2004): 913-939.
10. Wang, Jie, Jiayu Zhou, Peter Wonka, and Jieping Ye. "Advances in neural information processing systems." In *Neural information processing systems foundation*. 2013.
11. Chen, Yixin, Jinbo Bi, and James Ze Wang. "MILES: Multiple-instance learning via embedded instance selection." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28, no. 12 (2006): 1931-1947.
12. Li, Wu-Jun, and Dit-Yan Yeung. "Localized content-based image retrieval through evidence region identification." In *Computer*

Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pp. 1666-1673. IEEE, 2009.

13. Fan, Rong-En, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. "LIBLINEAR: A library for large linear classification." *Journal of machine learning research* 9, no. Aug (2008): 1871-1874.
14. Chang, Chih-Chung, and Chih-Jen Lin. "LIBSVM: a library for support vector machines." *ACM Transactions on Intelligent Systems and Technology (TIST)*2, no. 3 (2011): 27.
15. Zhang, Hui, Jason Fritts, and Sally Goldman. "An improved fine-grain hierarchical method of image segmentation." *Washington University CSE Technical Report* (2005).
16. Li Fei, and Rujie Liu. "Multi-graph multi-instance learning with soft label consistency for object-based image retrieval." In *Multimedia and Expo (ICME), 2015 IEEE International Conference on*, pp. 1-6. IEEE, 2015.