Using Particle Swarm Optimization for Image Regions Annotation

Mohamed Sami[1,*], Nashwa El-Bendary[2,*], Tai-hoon Kim[3,⋆], and Aboul Ella Hassanien[1,*]

[1] Cairo University, Faculty of Computers and Information, Cairo, Egypt
{mohamed.sami,abo}@egyptscience.net
[2] Arab Academy for Science, Technology, and Maritime Transport, Cairo, Egypt
nashwa.elbendary@ieee.org
[3] School of Information Science, University of Tasmania, Australia
taihoonn@empas.com

Abstract. In this paper, we propose an automatic image annotation approach for region labeling that takes advantage of both context and semantics present in segmented images. The proposed approach is based on multi-class K-nearest neighbor, k-means and particle swarm optimization (PSO) algorithms for feature weighting, in conjunction with normalized cuts-based image segmentation technique. This hybrid approach refines the output of multi-class classification that is based on the usage of K-nearest neighbor classifier for automatically labeling images regions from different classes. Each input image is segmented using the normalized cuts segmentation algorithm then a descriptor created for each segment. The PSO algorithm is employed as a search strategy for identifying an optimal feature subset. Extensive experimental results demonstrate that the proposed approach provides an increase in accuracy of annotation performance by about 40%, via applying PSO models, compared to having no PSO models applied, for the used dataset.

1 Introduction

With advent of digital imaging modalities, such as the increasingly affordable digital cameras and the widespread of personal computers, lead to build up a huge volume of digital images that are available in the digital libraries and on the web. This growth raises the demand to increase the usefulness of these large images archives. There is a growing need to have applications that effectively search and index these images. By using text query, images could be found in a manner that would meet the different needs of many users as most users prefer textual keyword search instead of content based image retrieval systems that ask user to enter image samples or low level features descriptions as query. Therefore, the importance of image annotation techniques for automatically assigning labels to images in order to fill the gap between the visual features search and the semantic means is rapidly increasing.

In this paper, we present an approach based on multi-class K-nearest neighbor (K-NN) classifier, K-means clustering, and particle swarm optimization (PSO) algorithm.

⋆ Scientific Research Group in Egypt (SRGE), http://www.egyptscience.net

T.-h. Kim, Y.-h. Lee, and W.-c. Fang (Eds.): FGIT 2012, LNCS 7709, pp. 241–250, 2012.

The common PSO [1], Trelea [2], and Clerc [3] models have been applied. We used the same normalized cuts-based algorithm for segmentation, visual features, and data-set as in [4]. Particle swarm optimization algorithm is used in our proposed approach for weighting feature vectors. The PSO fitness function includes the usage of K-NN and K-means algorithms as well as training and testing data. We used K-means to find the centroid of each class training data, and we use K-NN to find the closest class for each new feature vector.

The rest of this paper is organized as follows. Section 2 gives an overview of K-NN and particle swarm optimization algorithms. Section 3 presents the different phases of the proposed hybrid image annotation approach. Section 4 shows the obtained experimental results. Finally, Section 5 addresses conclusions and discusses future work.

2 K-Nearest Neighbor (K-NN) and Particle Swarm Optimization (PSO): Preliminaries

Due to space limitations we provide only a brief explanation of the basic framework of K-NN theory and particle swarm optimization algorithm, along with some of the key definitions. A more comprehensive review can be found in sources such as [1–3, 5–7].

2.1 K-Nearest Neighbor (K-NN)

The K-nearest neighbor (K-NN) method was first introduced by Fix and Hodges in 1951, and was one of the most popular nonparametric methods used for classification of new object based on attributes and training samples. The K-NN consists of a supervised learning algorithm where the result of a new instance query is classified based on the majority of the K-nearest neighbor category [8]. The K-NN method has been successfully applied in many areas: statistical estimation, pattern recognition, artificial intelligence, categorical problems, and feature selection. One advantage of the K-NN method is that it is simple and easy to implement.

K-NN is not negatively affected when training data are large, and indifferent to noisy training data. Disadvantages of the K-NN method are the need to determine parameter K (number of nearest neighbors), calculate the distances between the query instance and all the training samples, sort the distances and determine the nearest neighbors based on the K^{th} minimum distance, as well as determine the categories of the nearest neighbors.

2.2 Particle Swarm Optimization (PSO)

The concept of particle swarms, although initially introduced for simulating human social behaviors, has become very popular these days as an efficient search and optimization technique. Particle swarm optimization [1–3], does not require any gradient information of the function to be optimized, uses only primitive mathematical operators and is conceptually very simple. PSO has attracted the attention of a lot of researchers resulting into a large number of variants of the basic algorithm as well as many parameter automation strategies. The canonical PSO model consists of a swarm of particles, which are initialized with a population of random candidate solutions. They move iteratively through the d-dimension problem space to search the new solutions, where the fitness, f, can be calculated as the certain qualities measure. Each particle has a position represented by a position-vector x_i (i is the index of the particle), and a velocity represented by a velocity-vector v_i. Each particle remembers its own best position so far in a vector $x_i^{\#}$, and its j-th dimensional value is $x_{ij}^{\#}$. The best position-vector among the swarm so far is then stored in a vector x^*, and its j-th dimensional value is x_j^*. During the iteration time t, the update of the velocity from the previous velocity to the new velocity is determined by equation (1). The new position is then determined by the sum of the previous position and the new velocity by equation (2).

$$v_{ij}(t+1) = wv_{ij}(t) + c_1 r_1 (x_{ij}^{\#}(t) - x_{ij}(t)) + c_2 r_2 (x_j^*(t) - x_{ij}(t)) \tag{1}$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \tag{2}$$

Where w is called as the inertia factor, which governs how much the previous velocity should be retained from the previous time step, r_1 and r_2 are the random numbers, which are used to maintain the diversity of the population, and are uniformly distributed in the interval [0,1] for the j-th dimension of the i-th particle. c_1 is a positive constant, called as coefficient of the self-recognition component, c_2 is a positive constant, called as coefficient of the social component. From equation (1), a particle decides where to move next, considering its own experience, which is the memory of its best past position, and the experience of its most successful particle in the swarm. In the particle swarm model, the particle searches the solutions in the problem space with a range $[-s,s]$ (if the range is not symmetrical, it can be translated to the corresponding symmetrical range). In order to guide the particles effectively in the search space, the maximum moving distance during one iteration must be clamped in between the maximum velocity $[-v_{max}, v_{max}]$ given in equation (3):

$$v_{ij} = sign(v_{ij}) min(\left|v_{ij}\right|, v_{max}) \tag{3}$$

The value of v_{max} is $p \times s$, with $0.1 \leq p \leq 1.0$ and is usually chosen to be s, i.e. $p = 1$. The end criteria are usually one of the following:

- maximum number of iterations
- number of iterations without improvement
- minimum objective function error

3 Automatic Image Regions Annotation Approach

The intelligent PSO-based image regions annotation approach proposed in this paper is composed of four phases; namely, *segmentation phase*, where the input image is segmented into regions using normalized cuts algorithm, *feature extraction phase*, where a feature vector per segmented region is extracted, *feature optimization phase*, which applies a particle swarm optimization for features weighting per class, and finally *classification and annotation phase*, where the optimized weights are used with K-NN and K-Means for classification. This section describes each of these phases in details. An overview of the proposed approach is depicted in figure 1.

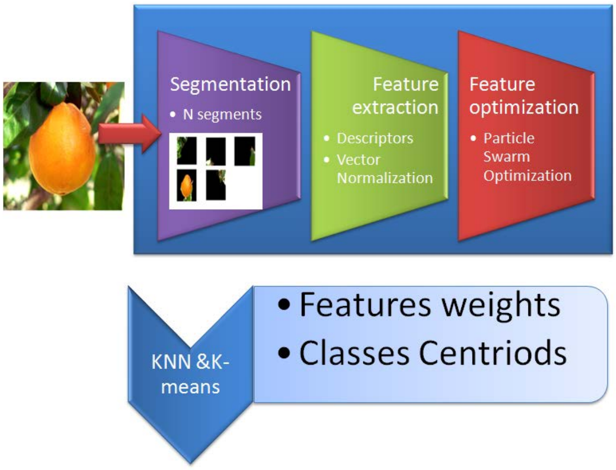


Fig. 1. Intelligent automatic image regions annotation system architecture

3.1 Segmentation Phase

In our proposed approach we aim to automatically annotate images based on regions. Therefore, the need for an image segmentation step is evident. For this, we applied the normalized cuts segmentation algorithm [9, 10], which formulates image segmentation as a graph partitioning problem and uses the normalized cut value between different graph groups.

In the employed graph theoretic formulation of grouping, a set of points in an arbitrary feature space is represented as a weighted undirected graph $G(V,E)$, where the nodes of the graph are the points in feature space, and an edge is formed between every pair of nodes. The weight on each edge $w(i,j)$ is a function of the similarity between nodes i and j. A graph $G(V,E)$ is partitioned into two disjoint sets A and B, where $A \bigcup B = V$ and $A \bigcap B = \phi$, by removing edges and hence performing a cut. The segment cut between A and B is defined as presented in equation (4).

$$cut(A,B) = \sum_{u \in A, v \in B} w(u,v) \tag{4}$$

The optimal partitioning is the one that minimizes the cut value. In order to avoid partitioning out small sets of points, instead of looking at total edge weights between the two parts, a fraction of cut cost for total edge connections to all the nodes in graph is calculated resulting what is called normalized cut given by equation (5).

$$Ncut(A,B) = \frac{Cut(A,B)}{assoc(A,V)} + \frac{Cut(A,B)}{assoc(B,V)} \tag{5}$$

Where, $assoc(A,V) = \sum_{u \in A, t \in V} w(u,t)$ is the total connection from nodes in group A to all nodes in the graph and $assoc(B,V)$ is similarly defined. In our approach, we segment every input image into five regions.

3.2 Feature Extraction Phase

During this phase, each image segment is represented by a visual feature vector characterizing its color, texture, and shape contents. A comprehensive list of the 83 features we have employed is given in Table 1.

Table 1. Region features

Feature Name	Length	Type	Details
Color Mean	3	Color	For RGB
Color StdDev	3	Color	For RGB
Segment size	1	Global	Number of pixels in image
Entropy	1	Global	
Moment	3	Global	For RGB
Dominant Color	4	Color	Color percentage to image with RGB
Eccentricity	1	Shape	Scalar that specifies the eccentricity of the ellipse that has the same second-moments as the region
Orientation	1	Shape	The angle (in degrees ranging from -90 to 90 degrees) between the x-axis and the major axis of the ellipse that has the same second-moments as the region
Convex Area	1	Shape	Scalar that specifies the number of pixels in the binary image that specifies the convex hull
Filled Area	1	Shape	Scalar specifying the number of on pixels in the binary image of the same size as region
Euler Number	1	Shape	Scalar that specifies the number of objects in the region minus the number of holes in those objects
Equiv Diameter	1	Shape	Scalar that specifies the diameter of a circle with the same area as the region
Solidity	1	Shape	Scalar specifying the proportion of the pixels in the convex hull that are also in the region
Perimeter	1	Shape	The distance around the boundary of the region
Gabor Filter	60	Texture	5 scales, 6 orientations
Total	**83**		

3.3 Feature Optimization Phase

A computational iterative PSO method, which works on a population of candidate solutions that are called particles. These particles follow mathematical formula and moving in search space. There are a local best position that influence each particle in this space. This best position is updated if a better position is found during the iterative search process. By this particle's movement the swarm should move toward the best solution available. The K-means algorithm is used to find the centroid of each class based on its belonged feature vectors. For the classification of any new non-labeled feature vector, the K-NN classifier with euclidian distance is used to find the closet class's centroid.

For the proposed approach, we used PSO coupled with K-means in order to optimize the feature set for image annotation per class. We applied real number particle swarm candidate solutions as weight. We have used four models for PSO algorithm, namely; common PSO, Trelea-1, Trelea-2, and Clerc models [1–3]. The length of each particle swarm is calculated according to equation (6).

$$Length = NumberOfClasses * NumberOfDescriptors \tag{6}$$

The fitness function of the PSO is based on K-NN evaluation with instance particle. Each class has its own training data and its own part of particle candidate that is used as feature weighting. After calculating all the centroids based on the training data and the weights, test data is used for evaluating the current weights. A fitness function is used to evaluate every particle using the centroids that have calculated and the K-NN for test data classification. Algorithm (1) describes the fitness value calculation. Table 2 defines the meaning of each parameter used in algorithm (1).

Table 2. Parameters used in Algorithm (1)

Parameter	Description
N	Number of classes including the background class
IndexTrain	Current training data item from certain class
T	Total number of test data
Centroid	Array of all classes centroids
Distance	Array distances between a specific data to all centroids
ExpectedClass	Correct current label of testing data

3.4 Classification/Annotation Phase

In this phase, we experiment particle swarm optimization algorithm with four different models, namely; the common PSO, Trelea-1, Trelea-2, and Clerc models [1–3]. K-means algorithm was used for evaluating the centroids of different data classes and K-NN classifier was used as well for the classification of test data. We assumed that there are *N* classes including a background class. For a new unknown feature vector, it is firstly normalized, then weighted and classified based on the already calculated centroids. The distance for all centroids is calculated and also the minimum distance points to the corresponding class.

4 Experimental Results

The proposed approach has been evaluated using six classes of fruit images: apple, banana, grapes, mango, orange, and watermelon, in addition to a class for background. Each class has in average about 15 training feature vectors and 10 for testing feature vectors, however the background class has the highest number of vectors which is 216 in training and 101 in testing. Each image is segmented, as previously mentioned, into five segments using the normalized cuts algorithm. Each segment was manually annotated to provide a ground truth. Table 3 and table 4 show the number of testing regions in

Algorithm 1. Fitness value calculation algorithm

for i=1 to N **do**
    Compute: $TrainingDataWeighted(j) = TrainingData(i) * Weights(i)$
    Train: Centroid(i)=Caculate-Centroid-KMeans($TrainingDataWeighted$, $TrainingLabels$)
end for
Result = Array of centroids for each class
for x=1 to T **do**
    for y=1 to N **do**
        Compute: $TestingDataWeighted = TestingData(x) * Weights(y)$
        Compute: Distance(y) = KNN.classify(($TestingDataWeighted$),$Centroid$(y))
    end for
    Result =Find index of minimum value in Distances array
    if Result = ExpectedClass **then**
        Increase the total number of correct labels
    end if
end for
Compute: the fitness value of current iteration as:

$$FitnessValue = \frac{TotalNumberOfCorrectLabels}{TotalTestData} \quad (7)$$

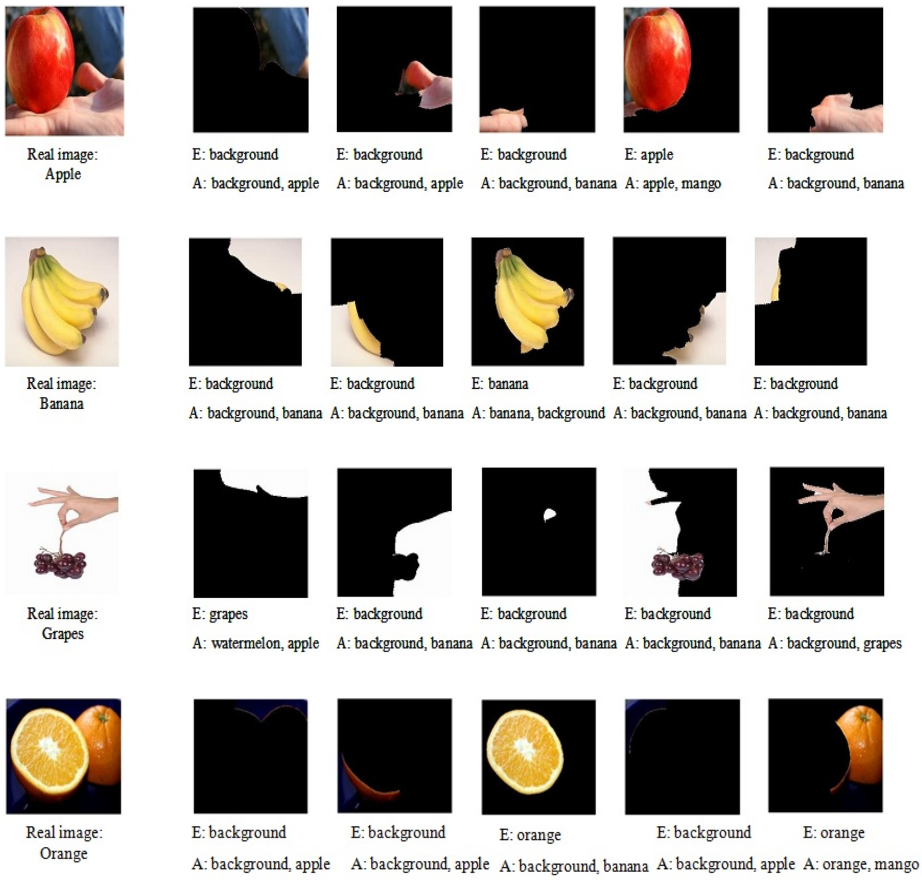
(A:actual) when using Trelea-1 parameter set for PSO and (E:expected) which is the ground truth for each class and the distribution of classes feature vectors, respectively. Figure 2 shows some normalized cuts segmentation results.

Table 3. Number of testing regions in (A:actual) and (E:expected) for each class

Class	A:actual:	E:expected
Grapes	1	9
Apple	6	6
Banana	5	8
Mango	7	12
Orange	5	8
Watermelon	3	6
Background	100	101
Total Number Of regions	127	150
Accuracy	**84.67%**	

For the PSO algorithm we have used a population size of 200, maximum iteration equal 400, inertia weights equal 0.9 and 0.6. acceleration constant equals 2.1, and the lowest error gradient tolerance value is 1e-99. The starting particle positions are random, and the particle values range is -100 to 100.

Table 5 shows obtained accuracy using the proposed approach without using any PSO model and through applying the four previously mentioned PSO models. As illustrated in table 5, the maximum accuracy achieved was via using the Trelea PSO models $\simeq$ 84% compared to $\simeq$ 45% without having any PSO models used.



E: Expected label, A: Actual results (First closest, Second closest)

Fig. 2. Examples of normalized cuts segmentation results

Table 4. Distribution of classes feature vectors

Class	Training vectors	Testing vectors
apple	13	6
banana	13	8
grapes	19	9
mango	11	12
orange	14	8
watermelon	14	6
background	216	101

That obtained annotation accuracy percentage is not optimal due to a number of reasons such as the limited dataset used with addition to mistakes happen sometimes in our manual annotation of training and testing regions. Also, the wrong segmentation of images taking in mind the variation in size and quality of images dataset represent another reason for the accuracy achieved. Moreover, some images are poorly segmented and wrongly classified as the number of regions representing fruits classes is very small compared to background classes.

For watermelon and mango classes, poor labeling results have been obtained due to poor segmentation as segmentation doesn't precisely segment objects and errors are generated due to manual labeling. Also, the number of regions representing background classes is very large compared to other fruits classes. Moreover, the number of test segments for watermelon and the number of training segments for the mango were the lowest. Furthermore, grapes class has the maximum number of training segments and also most of the misclassified mango segments were labeled as grapes, we have to mention that the worst segmentation results are in the grapes class images. These reasons for poor labeling results will be considered for further research.

Table 5. Results for different PSO Models

Models	**Accuracy**
Without-PSO	44.67%
Common PSO	82.00%
Trelea-1	84.67%
Trelea-2	84.00%
Clerc	82.00%

5 Conclusions and Future Work

In this paper, we have presented an automatic image regions annotation approach for region labeling that takes advantage of both context and semantics present in segmented images. The proposed approach is based on KNN classifier, K-means and particle swarm optimization (PSO) algorithms. The PSO was used for weighting the features vectors and this has increased the annotation performance by about 40% with achieving maximum annotation accuracy $\simeq$ 84%. Our focus in the future work is validating the current work proposed in this paper using a larger dataset, and also implementing new techniques for optimizing weighting and features selection. Moreover, we are going to consider using different image segmentation techniques in order to overcome the miss segmentations in our model. Currently, we are working on the imageclef [11] dataset.

References

1. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: Proceedings of IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948 (1995)
2. Trelea, I.C.: The Particle Swarm Optimization Algorithm: convergence analysis and parameter selection. Information Processing Letters 85(6), 317–325 (2003)

3. Clerc, M., Kennedy, J.: The particle swarm - explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation 6(1), 58–73 (2003)
4. Sami, M., El-Bendary, N., Hassanien, A.E., Schaefer, G.: Hybrid Intelligent Automatic Image Annotation Using Machine Learning. In: The 2011 Online Conference on Soft Computing in Industrial Applications WWW (WSC16) (2011)
5. Viswanath, P., Sarma, T.H.: An improvement to k-nearest neighbor classifier. In: Proceedings of IEEE Int. Conference on Recent Advances in Intelligent Computational Systems (RAICS), pp. 227–231 (2011)
6. Qiang, H., Aibing, J., Qiang, H.: Multiple Real-valued K nearest neighbor classifiers system by feature grouping. In: Proceedings of IEEE International Conference on Systems Man and Cybernetics (SMC), pp. 3922–3925 (2010)
7. Lin, L., Qi, L., Jun-yong, L., Chuan, L.: An improved particle swarm optimization algorithm. In: Proceedings of IEEE International Conference on Granular Computing (GrC 2008), pp. 486–490 (2008)
8. Castillo, O., Xu, L., Ao, S.I.: Trends in Intelligent Systems and Computer Engineering. Springer (2008)
9. Cai, W., Wu, J., Chung, A.C.S.: Shape-based image segmentation using normalized cuts. In: IEEE Int. Conference on Image Processing, pp. 1101–1104 (2006)
10. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Analysis and Machine Intelligence 22(8), 888–905 (2000)
11. Imageclef dataset website, `http://www.imageclef.org/datasets`