Proceedings of the Federated Conference on
Computer Science and Information Systems pp. 763–769
ISBN 978-83-60810-51-4

Incorporating Random Forest Trees with Particle Swarm Optimization for Automatic Image Annotation

Mohamed Sami and Aboul Ella Hassanien
Faculty of Computers
and Information - Cairo University
Scientific Research Group
in Egypt (SRGE)
http://www.egyptscience.net/
Cairo - Egypt
Email: mohamed.sami@egyptscience.net
abo@egyptscience.net

Nashwa El-Bendary
Arab Academy for Science,
Technology, and Maritime Transport
Scientific Research Group
in Egypt (SRGE)
http://www.egyptscience.net/
Cairo - Egypt
Email: nashwa.elbendary@ieee.org

Robert C. Berwick
Massachusetts Institute of
Technology (MIT)
USA
Email: berwick@csail.mit.edu

***Abstract*—This paper presents an automatic image annotation approach that integrates the random forest classifier with particle swarm optimization algorithm for classes' scores weighting. The proposed hybrid approach refines the output of multi-class classification that is based on the usage of random forest classifier for automatically labeling images with a number of words. Each input image is segmented using the normalized cuts segmentation algorithm in order to create a descriptor for each segment. Images feature vectors are clustered into K clusters and a random forest classifier is trained for each cluster. Particle swarm optimization algorithm is employed as a search strategy to identify an optimal weighting for classes' scores from random forest classifiers. The proposed approach has been applied on Corel5K benchmark dataset. Experimental results and comparative performance evaluation, for results obtained from the proposed approach and other related researches, demonstrate that the proposed approach outperforms the performance of other approaches, considering annotation accuracy, for the experimented dataset.**

I. Introduction

IN THE recent years, the size of digital image libraries has increased rapidly due to the development of multimedia and network technologies. Content based image retrieval (CBIR) [1] is a well-known technology for analyzing and controlling these image resources. In CBIR, the user has to enter low level visual features, such as color, shape, or texture, leaving a semantic gap in the user query results. In addition to that, keyword-based search is more user friendly than the visual features based search. So, automatic image annotation (AIA) techniques, which are technologies for assigning keywords in order to describe images context, has become more significant. AIA builds a bridge between the high level semantic and low level features, which is considered as an approach to solve the semantic gap problem. Many contributions have been done to the AIA field. Mori et al. [2] developed a co-occurrence model to establish the association between words and images. This model is for region labeling and it has involved four main steps. The first step is grid segmentation for the images, where they divided each image into equal rectangles. Authors selected this type of segmentation because it is fast and simple. The second step is feature extraction for regions, the third step is clustering the features vectors using vector quantization, and the last step is creating a probability model that links each word to a given cluster. On the other hand, Duygulu et al. [3] proposed a model based on machine translation. They treated AIA problem as learning lexicon. Moreover, Barnard and Forsyth [4] proposed a hierarchical model based on statistical clustering. They represented the words and the blobs as a distribution over the hierarchy nodes. This model is a hierarchical combination of the asymmetric and symmetric clustering models. In their experiments, they clustered about 3000 Corel images into 64 clusters. Also, Jeon et al. [5] proposed AIA model for annotating and retrieving images. In that model, images were segmented and features were extracted from each region. They used the same segmentation algorithm and the same 33 features as in Duygulu et al. [3] and Lavrenko et al. [6]. However, for Jeon et al.'s work, they proposed a continuous relevance model (CRM) instead of the previously discussed discrete model. They assumed that each image region is represented by continuous valued feature vector. Furthermore, in [7], multiple-bernoulli relevance model was proposed to improve CRM and cross-media relevance model (CMRM). Shunle and Xiaoqiu [8] proposed an AIA model based on multi instance learning. The proposed approach in this paper is based on the random forest classifier with particle swarm optimization algorithm for classes' scores weighting. Many experiments have been done to Corel5k benchmark dataset and the results have been compared to previous related works. The rest of this paper is organized as follows. Section II introduces a brief description of random forest classifier and particle swarm optimization algorithm used in the proposed approach.

Section III describes in details the proposed image annotation approach. Section IV presents experimental results. Finally, section V addresses conclusions and discusses future work.

II. Particle Swarm Optimization (PSO) and Random Forest Classifier : Preliminaries

Due to space limitations we provide only a brief explanation of the basic framework of particle swarm optimization algorithm and random forest classifier, along with some of the key definitions. A more comprehensive review can be found in sources such as [9]–[14].

A. Particle swarm optimization

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 (PSO) [9]–[11], does not require any gradient information of the function to be optimized. It 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 is 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 $\vec{x}_i$ (i is the index of the particle) and a velocity represented by a velocity-vector $\vec{v}_i$. Each particle remembers its own best position so far in a vector $\vec{x}_i^{\#}$ and its j-th dimensional value is $x_{ij}^{\#}$. The best position-vector among the swarm so far is then stored in a vector $\vec{x}^*$ 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, and the new position is then determined by the sum of the previous position and the new velocity, for more details refer to our published work in [15].

B. Random forest classifier

Random forest classifier [16]–[18] is an ensemble classifier that consists of several decision trees [19]. The output of this classifier is the class number that most frequently occurs individually in the output of decision trees classifiers. The main idea of decision trees is to predicate a target based on a group of input data. Decision trees also named classification trees, where the tree leaves represent the class labels and the branches represent the conjunction of feature vectors that lead to class labels. As depicted in figure 1, each interior node represents an input feature and each node has children of another input feature. The training of decision tree is based on a process called recursive partitioning; this is a recursive process where the input dataset is split into subsets. Recursion stopping condition is when all the tree nodes have the same output targets. For the approach proposed in this paper, the targets are the words.

Fig. 1. Decision Tree to differentiate between melon and orange classes

Generally, there are two types of decision trees:

1) **Regression tree:** where the proposed targets are real numbers (for example: price of a car).
2) **Classification tree:** where the proposed targets are specific classes (for example: is a Female, is a Male).

Classification decision tree is the type used for the proposed approach in this paper. Also, there is an important concept for decision tree learning, which is called *"decision tree pruning"*. Tree pruning is a process that aims to reduce the size of decision tree by removing parts of the tree that give small voting for classifications. This technique has advantage of reducing the size and the complexity of the produced tree with addition to reduction of over-fitting in some cases. However, random forest algorithm doesn't use this technique. Instead, random forest classifier takes n_{tree} as parameter that corresponds to the number of decision trees that will be created in the ensemble bagged forest classifier. Algorithm (1) shows random forests training for each decision tree.

Algorithm 1 Random forests training

```
Set: Number of classes = N, Number of features = M
Let: m determine the number of features at a node of
  decision tree, (m < M)
for each decision tree do
  Select randomly: a subset (with replacement) of training data that represents the N classes and use the rest of data to measure the error of the tree
  for Each node of this tree do
    Select randomly: m features to determine the decision at this node and calculate the best split accordingly. /* No tree pruning used */
  end for
end for
```

The random forest error rate depends on two things:

1) **Correlation:** represents correlation between any two trees in the forest. Error increases as the correlation increases.

2) **Strength:** represents the strength of each tree in the forest. The strength is measured by the error rate; a tree with low error is a strong tree. The forest error rate decreases as the decision tree's strength increases.

One of the advantages of random forest classifier is that it is one of the highly accurate classifiers. On the other hand, it has been observed to over-fit for some datasets with noisy classification tasks.

III. The proposed AIA approach

For the proposed approach, random forest classifier has been used. Figure 2 shows the model of the proposed automatic image annotation approach. The proposed AIA approach is consisted of two phases, which are training phase and testing phase.

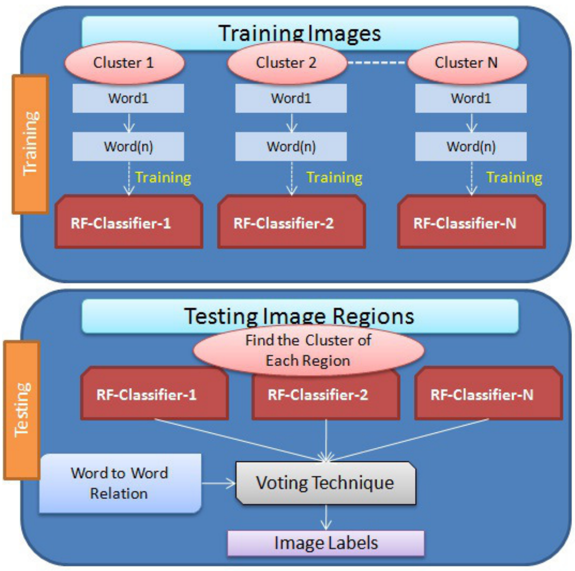


Fig. 2. Random forest classifier based model

Firstly, in the training phase, image regions are clustered into K clusters and for each cluster a random forest classifier has been trained. Thus, K random forest classifiers will be resulted at the end of training phase. Each forest classifies the words (classes) existed per cluster. In the proposed model, we have taken the advantage of the clustering well known good accuracy. To find the words existed in any cluster, the regions in this cluster and accordingly the related images were found using Euclidean distance. Any label for an image means that all the image regions are contributing to this label word (class). In other words, we aimed to find the images that exist in each cluster and their related words. Consequently, each word has been trained from its related images in that cluster. All the forest per cluster have an equal number of decision trees. Secondly, in the testing phase, for each unlabeled image, every image region feature vector is clustered using Euclidian distance to the nearest cluster, then the region feature vector is classified using cluster's forest classifier. The output of the region forest classification contains scores per

TABLE I
PARAMETERS FOR ALGORITHM (2)

Parameter	Description
T	Total number of test images
N	Total number of regions per image
W	Total number of test data
ClstrNumber (CN)	The region vector features cluster number
ClssScores (CS)	The output scores for a region vectors
CorrletionLabels (CL)	The word to word repeats
WrdCorr (WC)	The correlation words for the class word with highest score
FreqTrainWords (FTW)	the frequencies per word for each training dataset
WindowSize (WZ)	the window size controls the size of part taken from the sorted labels per region
ComVotes (CV)	variable contains cumulated votes for each class
NewLabels (NL)	The output labels for unknown image

word (class). The sorted words enter a voting process for the image labeling. The voting takes into consideration different parameters including the regions size, words frequents, and words correlations. The region classification classes sorted in descending order using the forest scores per class using the parameter $WindowSize$. The words frequencies is calculated using equation (1).

$$FreqTrainWords(n) = \frac{TotolTrainingAnnotation}{WordRepeats(n) * C} \quad (1)$$

Where $WordRepeats$ is array for the number of annotations per class in the training dataset, $TotolTrainingAnnotation$ is total number of annotations in the training dataset, and C is a constant. The main propose of this constant is to decrease the affect $FreqTrainWords$ in voting process as the $TotolTrainingAnnotation$ is a large number. The $FreqTrainWords$ is used in algorithm (2), which shows the details of the annotation steps for unlabeled image. Table I presents description for the parameters used in algorithm (2).

In algorithm(2), for each image, each region closest cluster index is found using Euclidean distance and stored at CN, number in CN is used in order to find the index of cluster classifier $RndmTree$. Then, the selected classifier is used to get the classes' scores of the current features vector. After that, descending sort using associated random forest scores is applied. For WC variable, we store the word correlations for the highest scored word (class) and we use it later in voting calculations. As well, CV variable will represent all the votes sum for current image and it will be reset to zero for each new image. Finally, based on the CV content, any new image will be annotated with $NumOfLabels$ words (classes) those have highest votes.

PSO algorithm has been applied to random forest classifiers

Algorithm 2 Annotation using RFC algorithm

for x=1 to T **do**
  Set $CV = 0$
  for y=1 to N **do**
    Compute CN = $FindCluster(RgnVctr)$
    Compute $CS(y)$ = $RndmTree(CN).classify(RgnVctr)$
    Compute $SC(y)$ = $SortByScore(CS)$
    Compute WC = $CL(SrtdCls(1))$
    for s=1 to WZ **do**
      Compute $CV(SC(s))$= $CV(SC(s))$+$regionSize(i)$ + $FreqTrainWords(SrtdCls(s))$
      Compute $CV(WC(SrtdCls(s)))$ = $CV(WC(SrtdCls(s)))$+ $regionSize(i)$ + $FreqTrainWords(WC(SrtdCls(s)))$
    end for
  end for
  $SortedComulatedLabels$ =Sort(CV)
  $NewLabels$=$SortedComulatedLabels(1 : NumOfLabels)$
end for

in order to weight the classes' scores. Algorithm (3) shows the updated part of algorithm(2), which has been used for calculating the PSO fitness function.

Algorithm 3 Fitness RFC with scores algorithm

Set: $ComVotes = 0$
for s=1 to WindowSize **do**
  Compute: $ComVotes(SrtdCls(s))$ = $ComVotes(SrtdCls(s))$ + $Scores(SrtdCls(s))$ * $ClassWeight(SrtdCls(s))$
end for

Where, $Scores$ variable contains all the random forest's scores for the classes those exist in a specific cluster. $ClassWeight$ variable includes the generated PSO weights. The average precision measure has been used as the fitness value and the classes' scores has been used for voting for simplicity reasons. PSO classes' weights are multiplied to each regions' features vectors classifications output scores.

IV. Experimental results

In our experiments, we used Corel5k [20] dataset. The Corel dataset consists of 5000 images from 50 Corel Stock Photo CDs. Each cd includes 100 images on the same topic and each image is also associated with 1-5 keywords. This dataset is divided into 4500 images for training and 500 images for testing. In the training dataset there are 371 words. We consider each word as a class, as previously explained in section III. Each Image is segmented using normalized cuts segmentation algorithm, then the region with size larger than a certain threshold is selected. Each image has a number of regions between 5 to 10. There are 42379 regions for all the training dataset. For each region, a 33 features vector is extracted and the regions are clustered into 500 clusters. These features include segment size, location, convexity, first moment, region color, and region average orientation energy. The dimension of each feature vector is 36. The size of testing data is 500 images and includes only 263 words.

In order to measure our experiments, we used the same measures applied in previous works on Corel5k benchmark dataset. These measures are well known in the field of automatic image annotation. the first measure is the precision, which is referred as the ratio of the counter of correct annotation in relation to all the times of annotation. The second measure is the recall, which is referred as the ratio of the times of correct annotation in relation to all the positive annotated samples. Equations (2) and (3) show the calculations of precision and recall measures, respectively.

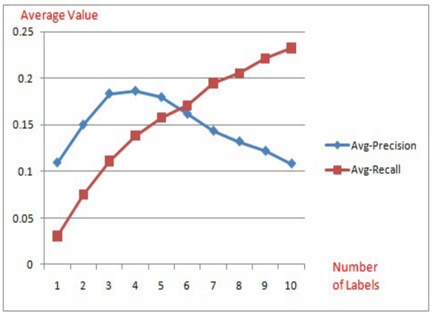
$$Precision = \frac{B}{A} \tag{2}$$

$$Recall = \frac{B}{C} \tag{3}$$

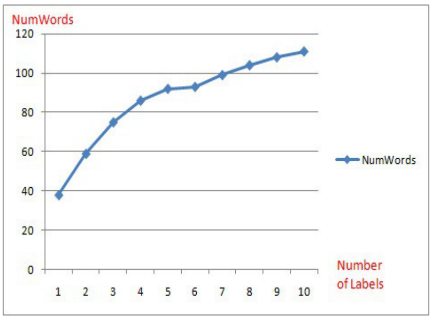
Where A is the number of images annotated by some keyword, B is the number of images annotated correctly, and C is the number of images annotated by some keyword in the whole dataset. Another measure is $NumWords$, which is statistics of the number of correctly annotated keywords that are used to correctly annotate at least one image. This statistical measure reflects the coverage of keywords in the different proposed methods. Figures 3 and 4 show the results for the random forest AIA model on the 500 Corel images, considering that most of these testing images are annotated with four labels at maximum. For all the clusters, n_{tree} equals to 50 has been used. As the number of decision trees per forest increases, classification accuracy increases accordingly. The n_{tree} has not been increased to be more than 50 due to memory limitation. As a results of experiments conducted on Corel5k benchmark, figures 3 (a) and (b) show some statistics for the random forest AIA based model.

In figure 3, the average precision and average recall against the number of labels at random forest approach are depicted for using voting window size equals to 9. The best average precision value which is achieved in this case is 0.1862 when using 4 labels. The recall curve increase as the number of labels increases and it is intersected with the precision curve when the number of labels equals 5. At number of labels equals to 5, the average precision is equal to 0.1795, the average recall is equal to 0.1577, and the "NumWords" is equal to 92. These resulted were obtained considering that the test image is labeled with 1 to 4 words.

Figure 4 illustrates the change in the average precision and recall values when applying changes in the voting technique per image. In figure 4, we compared six different cases. (1) "No Correlation" case, where the relation of words is



(a) Avg-Precision, Avg-Recall curve



(b) NumWords results curve

Fig. 3. Avg-Precision, Avg-Recall and NumWords results for forest AIA with windowSize=9

removed from the voting part in forest AIA algorithm. (2) ”Local Correlations” case, which means using the words correlations within each cluster instead of the correlation in overall the training images. (3) ”Region only” case, which means to use only the region size in voting technique. (4) ”Region+Correlations” case, which means to remove the part of words frequents in the voting algorithm. (5) ”divide by 4000” case, which means to divide all the frequent words array with C=4000, taking into consideration that in all the previous figures and tests the value was 10000. Finally, (6) ”Vote+1”, which means adding just 1 for each occurrence of a class without using region values or words frequent values. Applying normalization to the frequent words array achieved lower accuracy than using constant division.

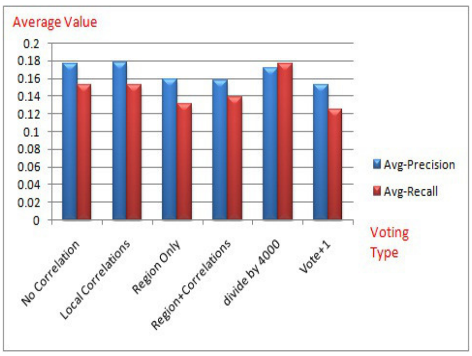


Fig. 4. Different voting techniques for random forest with clustering (RFC)

TABLE II
THE PERFORMANCES OF VARIOUS ANNOTATION MODELS ON COREL5K VS RFC

Model	Average Precision	Average Recall	NumWords
COM	0.03	0.02	19
TM	0.06	0.04	49
CMRM	0.10	0.09	66
CRM	0.16	0.19	107
MBRM	0.24	0.25	122
MIL	0.20	0.22	124
RFC	0.18	0.18	104

TABLE III
THE ACCURACY RESULTS FOR OPTIMIZED FOREST AIA APPROACH

Models	Average Precision	Average Recall	NumWords
RFC (w=5)	0.1386	0.1407	72
RFC (w=9)	0.1494	0.1335	70
RFC (w=15)	0.1482	0.1356	69
RFC+PSO-1(w=15)	0.2207	0.1437	86
RFC+PSO-2(w=15)	0.2068	0.2052	100
RFC+PSO(Trela1)(w=9)	0.2510	0.2170	108
RFC+PSO(Trela1)(w=15)	0.2571	0.2182	109

From figure 4, one can notice that the best results obtained via applying case (5), ”divide by 4000”, where the average precision is equal to 0.18, average recall is equal to 0.18, and $NumWords$ is equal to 104. The best results have been obtained for the case where $windowSize$ equals to 9. The reason for this is that it is the average of words correlations within clusters. Table II compare the forest based approach proposed in this paper and previous traditional annotation models such as COM [2], TM [3], CMRM [5], CRM [6], MBRM [7], and MIL [8]. The proposed model is marked as RFC that stands for $randomforestwithclustering$.

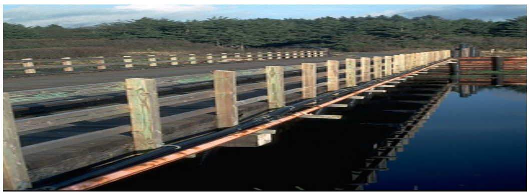
Table II shows that the proposed random forest model is one of the highest accuracy AIA models, however not the best one on Corel5K. For achieving the goal of enhancing the accuracy of RFC, PSO algorithm has been applied to the random forest approach where the classes’ scores are weighted and the average precision acted as the fitness value in the first RFC-PSO experiment and the sum of average precision and average recall in the second RFC-PSO experiment. The $windowSize$ in these two experiments is equals to 9. Table III presents the results before and after using PSO weights. For PSO configuration, number of iterations equals to 400, number of particles equals to 200, and velocity step equals to 2 have been used. RFC with PSO(Trela1) in case of $windowSize$ equals to 9 and 15 achieved the best results. In table III, at RFC case, random forest scores have been used without any weighting, while in case $RFC+PSO-1$ PSO weights have been applied with average precision as fitness value. In $RFC+PSO-2$ case we have used the sum of both the average precision and average recall as fitness value.

It is clear that merging the PSO algorithm with RFC approach has a big affect on the overall performance of RFC approach. Table IV compares our RFC with PSO to the related research models, it appears from the following table that our approach achieved a competitive accuracy. Table V shows a

TABLE IV
THE PERFORMANCES OF VARIOUS ANNOTATION MODEL ON COREL VS RFC WITH PSO (TERLA1) WINDOWSIZE=15

Model	Average Precision	Average Recall	NumWords
COM	0.03	0.02	19
TM	0.06	0.04	49
CMRM	0.10	0.09	66
CRM	0.16	0.19	107
MBRM	0.24	0.25	122
MIL	0.20	0.22	124
RFC-PSO	0.26	0.22	109

TABLE V
ANNOTATION EXAMPLE ON SAMPLES FROM COREL5K BENCHMARK

Test Image
corel-id=130034  -Ground Truth: grass, ground, shore -RFC: tree, sky, ground, head, herd -RFC+PSO(Trela): foals, deer, head, tree, grass
corel-id=163062  -Ground Truth: birds, branch, nest -RFC: tree, butterfly, wings, birds, water -RFC+PSO(Trela): birds, tree, flowers, foals, nest
corel-id=22013  -Ground Truth: bridge, water, wood -RFC: tree, water, whales, sky, cheese -RFC+PSO(Trela): sky, tree, mountain, rocks, coyote

three test images samples with the annotation results for the RFC and RFC with PSO(Trela1).

V. CONCLUSIONS AND FUTURE WORK

In this paper, an automatic image annotation approach, based on random forest classifier and particle swarm optimization algorithm, has been proposed and tested. The proposed approach shows that applying PSO algorithm with random forest increased the average precision from 0.1482 to 0.2207. The best result achieved using PSO (Trela1) with random forest classifier, where precision = 0.26 and recall = 0.22. For the proposed random forest model, the error happens in the clustering stage affects the classification output as a cumulative error. In addition, there are no direct correspondence between the images regions and the classes in Corel5k dataset, that is an image used for the class 'sky' may also used for the class 'tree'. There are minority classes that are represented in some cases with one image in the training Corel5k set, the case that leads to hard classifications for these classes. Creating one tree to classify all the classes was infeasible due to memory limitations. For future work, testing different numbers of clusters may has a noticeable impact on the overall performance. Also, applying features selections and weighting techniques or using different features than the ones generated in the Corel5k is another point of research. Moreover, changing the number of decision trees used in the random forest classifiers should leads to new results.

REFERENCES

[1] W. M. Arnold Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22:1349-1380, 2000.

[2] Y. Mori, H. Takahashi, and R. Oka, "Image-to-word transformation based on dividing and vector quantizing images with words." in *Proc. the International Workshop on Multimedia Intelligent Storage and Retrieval Management*, Orlando, Florida, USA, 1999.

[3] P. Duygulu, K. Barnard, J. Freitas, and D. Forsyth, "Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary." in *Proc. 7th European Conference on Computer Vision*, Copenhagen, Danemark, 2002, pp. 97–112.

[4] K. Barnard and D. Forsyth, "Learning the semantics of words and pictures." in *Proc. 8th IEEE International Conference on Computer Vision (ICCV 2001)*, Vancouver, British Columbia, Canada, 2001, vol. 2, pp. 408–415.

[5] J. Jeon, V. Lavrenko, and R. Manmatha, "Automatic image annotation and retrieval using cross-media relevance models." in *Proc. 26th Annual Int ACM SIGIR2003 Conference on Research and Development in Information Retrieval*, Toronto, Canada, 2003, pp. 119–126.

[6] V. Lavrenko, R. Manmatha, and J. Jeon, "A model for learning the semantics of pictures." in *17th Conference on Neural Information Processing Systems*, MIT, Cambridge, USA, 2003, vol. 16, pp. 553–560.

[7] R. Manmatha, V. Lavrenko, and S. L. Feng, "Multiple bernoulli relevance models for image and video annotation." in *Proc. IEEE Comput. Soc. Conf. on Computer Vision and Pattern Recognition*, Washington, DC, USA, 2004, pp. 1002–1009.

[8] S. Zhu and X. Tan, "A Novel Automatic Image Annotation Method Based on Multi-Instance Learning." *Procedia Engineering*, vol. 15, pp. 3439–3444, 2011.

[9] J. Kennedy and R. Eberhart, "Particle Swarm Optimization." in *Proc. IEEE International Conference on Neural Networks*, Perth, Australia, 1995, vol. 4, pp. 1942–1948.

[10] I. C. Trelea, "The Particle Swarm Optimization Algorithm: convergence analysis and parameter selection." *Information Processing Letters*, vol. 85, no. 6, pp. 317–325, 2003.

[11] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space." *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2003.

[12] P. Viswanath and T. H. Sarma, "An improvement to k-nearest neighbor classifier." in *Proc. IEEE International Conference on Recent Advances in Intelligent Computational Systems (RAICS)*, Kerala, India, 2011, pp. 227-231.

[13] H. Qiang and J. Aibing, "Multiple Real-valued K nearest neighbor classifiers system by feature grouping." in *Proc. IEEE International Conference on Systems Man and Cybernetics (SMC)*, Istanbul, Turkey, 2010, pp. 3922–3925.

[14] L. Lin, L. Qi, L. Jun-yong, and L. Chuan, "An improved particle swarm optimization algorithm." in *Proc. IEEE International Conference on Granular Computing (GrC 2008)*, Hangzhou, China, 2008, pp. 486–490.

[15] M. Sami, N. El-Bendary, A. E. Hassanien, K. Xiao, and A. A. Manaf,"Particle Swarm Optimization based Multi-Class Image Annotation." in *35th IEEE International Conference on Telecommunications and Signal Processing (TSP2012)*, Prague, Czech Republic, July 3-4, 2012.

[16] L. Breiman, "Random Forests." *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[17] T. K. Ho, "Random decision forests." in *Proc. 3rd International Conference on Document Analysis and Recognition*, Montreal, Canada, 1995, vol. 1, pp. 278–282.

[18] T. K. Ho, "The Random Subspace Method for Constructing Decision Forests." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, 1998.

[19] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. 1st ed, New York: Chapman and Hall, 1984.

[20] Corel5k dataset website. [Online]. Available: http://kobus.ca/research/data/eccv_2002/