Features based Classification of Images using Weighted Feature Support Vector Machines

Amutha A.L.
PG Scholar
Department of Computer Science and Engineering
SSN College of Engineering
Chennai - 603110.

Kavitha S.
Assistant Professor
Department of Computer Science and Engineering
SSN College of Engineering
Chennai – 603110.

ABSTRACT

In the recent research era analyzing and classifying images into meaningful categories using low-level visual features and high-level semantic features is a challenging and important problem. This paper focuses on the classification of COREL dataset images into its specific category using Weighted Feature Support Vector Machines (WFSVM) and the results are compared with Support Vector Machine (SVM) for validation. In WFSVM, the kernel function is precomputed by assigning more weight to relevant features using the principle of maximizing deviations. Initially, any two classes of COREL dataset is divided into training and test set and segmented using Fuzzy C Means clustering. Then from each segment of the image, color and texture features are extracted. The extracted features of the training dataset are used to construct the weighted features and precomputed linear kernel for training the WFSVM and its model file is created. Using this model file the features of test samples are classified into its specific category. Overall accuracy of classification using WFSVM is 99%, and the number of support vectors created is 6 whereas the accuracy of traditional SVM is 97% and the number of support vectors created is 12, justifies the performance of the proposed method with the existing methods.

Keywords

Texture features; Color features; Kernel Function; Support Vector Machine; Weighted Feature Support Vector Machine

1. INTRODUCTION

Fields such as medicine, film, video production, photography, remote sensing and security monitoring used analog imaging previously are switching to digital imaging, for their flexibility and affordability. These sources produce huge volumes of digital image data every day. Processing and analyzing these huge volumes of images manually is a tedious job. Techniques for manipulating these images with little or no human intervention are required. Image analysis is defined as the “act of examining images for the purpose of identifying objects and judging their significance”. Image analyst studies the image data and tries to detect, identify, classify measure and evaluates the significance of objects, their patterns, spatial and temporal relationship. There are many different techniques used in analyzing images automatically: image segmentation, image classification, image understanding and pattern recognition.

To analyse or interpret an image automatically, there must be a way of identifying unambiguously the pixels that correspond to particular features of interest. The process of identifying these pixels is known as segmentation. Segmentation techniques are widely used in many applications involving the detection, recognition, classification and measurement of objects in images. The success or failure of these tasks is a direct consequence of the success or failure of segmentation. The common segmentation method is clustering. The K-means algorithm is an iterative technique that is used to partition an image into K clusters. A drawback of the K-means algorithm is, the number of clusters K is an input parameter. An inappropriate choice of K may yield poor results. This can be resolved by Fuzzy C Means (FCM) algorithm [8]. The FCM employs fuzzy partitioning such that a pixel can belong to all groups with different membership grades between 0 and 1. In this paper, FCM algorithm is used for image segmentation. Quality of clustering is important together with increasing importance of clustering. Therefore validity functions are required to identify the best clustering, namely Partition Coefficient (PC), Classification Entropy (CE), Partition Exponent (PE), Compact and Separate Clustering (CSC) and Index (S) [2] [8] [12]. PC, CE and PE validity measures are lacking direct connection to geometrical property. But S validity function includes geometrical properties [4] [5] and it is a proportion of compactness to separation. Here S measure is used to validate the clustering.

In [1] organizing images into (semantically) meaningful categories is addressed as a challenging problem and it refers to the problem of semantic gap which is the key hindrance in all applications. The supervised machine learning techniques such as support vector machine (SVM), Bayesian classifier are often used to reduce the semantic gap in image classification [10] [11]. Their principal advantage is their good generalization capability. But, the traditional SVM algorithm on image classification have not distinguished the differences of different features for object classification and assigns the same weight to all low-level features. For high dimensional image data there are many features but not all the features are relevant to classification. A Weighted Support Vector Machine (WSVM) [13], assigns different weights to samples in different classes using Kernel-based Possibilistic C-Means (KPCM) algorithm. In [6] Weighted Feature Support Vector Machine (WFSVM) is proposed, where the relevant feature is determined using the degree of discrete. In [3] weights of the features are calculated using the principle of maximizing deviations between categories. In [13] and [6] the feature's weights are calculated but they neglect the relative importance of each feature with respect to classification task. In this paper weights of the features are calculated using principle of maximizing deviations between categories. From the extracted features and feature's weight the precomputed linear kernel is constructed and trained. Then the test samples are validated with the test features.

This paper elaborates on the following sections. In Section II, the system design of the proposed methodology is explained; Section III describes about the experimental results and performance of the SVM and WFSVM approaches based on the color and texture features from an image. Conclusion and Future work are summarised at the end.

2. SYSTEM DESIGN

In this system the digital images of COREL dataset are considered for analysis and evaluation in classification. The images are segmented using Fuzzy C Means clustering. From the segmented image, the color and texture features are extracted. From the feature set of training samples, the weighted features are computed and which is used for the construction of precomputed kernel in WFSVM. Then the kernel matrix of test features is given as input to the WFSVM classifier for classifying the images into their specific category. The system design is shown in Figure 1.

2.1 Dataset

The dataset of digital images are collected from the COREL database [15]. There are 1000 images in the data set. The dataset has 10 thematically diverse image categories. Each category has 100 images. All images are in JPEG format with size 384×256 or 256×384. A keyword is assigned to describe each image category. The categories are:

Africa

Beach

Building

Bus

Dinosaur

Elephant

Rose

Food

Horse

Mountain

The images of any two categories of COREL dataset are used for classification. The images are separated as training and testing set each having 100 images. For WFSVM and SVM the training and testing images are divided equally.

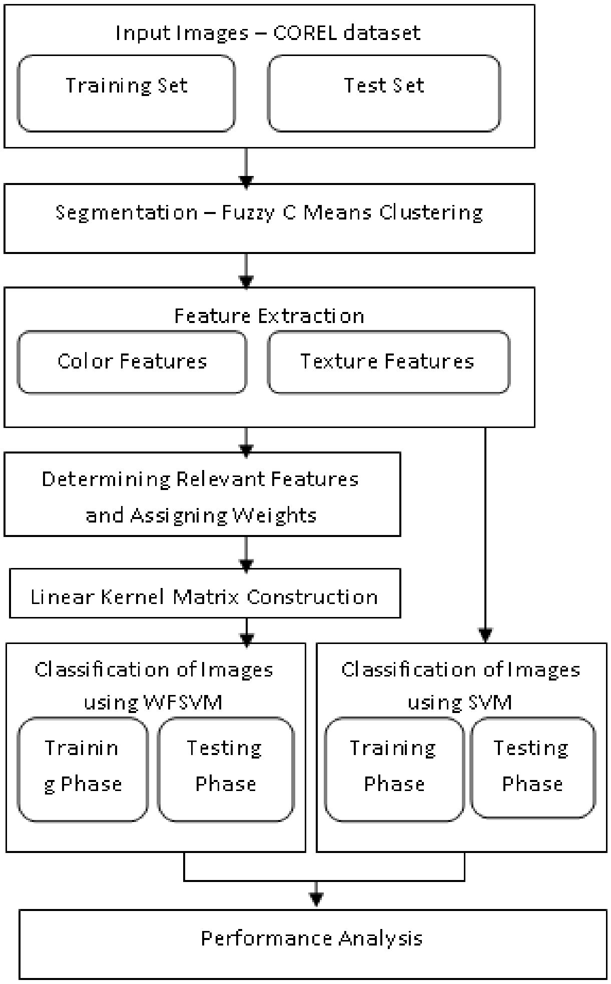
Input Images – COREL dataset
Training Set
Test Set
Segmentation – Fuzzy C Means Clustering
Feature Extraction
Color Features
Texture Features
Determining Relevant Features and Assigning Weights
Linear Kernel Matrix Construction
Classification of Images using WFSVM
Trainin g Phase
Testing Phase
Classification of Images using SVM
Training Phase
Testing Phase
Performance Analysis

Figure 1: System Design

2.2 Image Segmentation

Image segmentation is carried out using fuzzy c means algorithm [9]. The fuzzy c-means algorithm is based on minimization of the objective function given in Equation (1), with respect to μ, a fuzzy c-partition of the data set, and to *v*, a set of *c* centroids.

$$J_{FCM} = \frac{1}{2}\sum_{x=1}^{N}\sum_{i=1}^{c}\mu_{x,i}^{m}\, d^2(z_x, v_i) \qquad (1)$$

where, $\mu_{x,i}$ *(x=1,2,3,....N, i =1,2,....c)* is membership value, it denotes fuzzy membership of data point *x* belonging to class *i*, v_i *(i=1,2,3....c)* is centroid of each cluster, z_x *(x=1,2,3.....N)* is data set(pixel values in image), *m* is the fuzzification parameter, $d^2(z_x,v_i)$ is the Euclidean distance between z_x and v_i, *N* is the number of data points and *c* is the number of clusters. Fuzzy partition is carried out through an iterative optimization of the objective function. The sequence of steps is:

i. Choose primary centroids v_i (prototypes).
ii. Compute the degree of membership of all data set in all the clusters using Equation (2).

$$\mu_{x,i} = \frac{\left(\frac{1}{d^2(z_x, v_i)}\right)^{1/(m-1)}}{\sum_{i=1}^{c}\left(\frac{1}{d^2(z_x, v_i)}\right)^{1/(m-1)}} \qquad (2)$$

iii. Compute new centroids v'_i using Equation (3).

$$v'_i = \frac{\sum_{x=1}^{N} \mu_{x,i}^{m} z_x}{\sum_{x=1}^{N} \mu_{x,i}^{m}} \qquad (3)$$

and update the degree of membership according to ii.

iv. If $max_{x,i}[|\mu_{x,i} - \mu'_{x,i}|] < \varepsilon$ stop, otherwise goto step iii. where ε is the termination criterion. $d^2(z_x,v_i)$ is the Euclidean distance as defined in Equation (4).

$$d^2(z_x, v_i) = ||z_x - v_i||^2 \qquad (4)$$

2.2.1 Validation of Clustered Output

The segmented image is validated using the validity function, index(S) [9]. A smaller S indicates a partition in which all the clusters are overall compact and separate to each other. S is given in Equation (5).

$$S = \frac{compactness}{separation} = \frac{\pi}{s} \qquad (5)$$

The compactness of fuzzy cluster c_i is computed using Equation (6),

$$compactness = \pi = \frac{\sigma}{N} \qquad (6)$$

$$\sigma = \sum \sigma_i\,, i = 1,..c \qquad (7)$$

where N is the number of data points. The variation of fuzzy cluster i is defined in Equation (8).

$$\sigma_i = \sum (d_{x,i})^2\,, x = 1, \ldots, N \qquad (8)$$

$d_{x,i}$ is called the fuzzy deviation of z_x from class i and it is given in Equation (9).

$$d_{(x,i)} = \mu_{x,i}||v_i - z_x|| \qquad (9)$$

s is separation of the fuzzy c-partition as defined in Equation (10), where d_{min} is minimum distance between cluster centroids as given in Equation (11).

$$s = (d_{\min})^2 \qquad (10)$$

$$d_{min} = min_{i,t}||v_i - v_t|| \qquad (11)$$

The compactness and separation validity function S is defined as the ratio of compactness to separation as given in Equation (12), and partition index is obtained by summing up this ratio over all clusters as defined in Equation (13).

$$S = \frac{\pi}{s} = \frac{\sigma/N}{(d_{\min})^2} \qquad (12)$$

$$S = \frac{\sum_{i=1}^{c}\sum_{x=1}^{N}\mu_{x,i}^2||v_i - z_x||^2}{N_{min_{i,t}}||v_i - v_t||^2} \qquad (13)$$

2.3 Feature Extraction

From the segmented image 24 low-level features are extracted [7]. They are

i. Color Features (12)

ii. Texture Features (12)

2.3.1 Color Features

The 12 color features include:

- Average of R, G, B values(3)
- Standard deviation of R, G, B values(3)
- Average of L*, a*, b* values(3)
- Standard deviation of L*, a*, b* values(3)

2.3.2 Texture Features

Texture features are extracted using the 2-D Gabor filter, as given in Equation (14).

$h(x,y) = \exp(-\alpha 2\omega(x^2 + y^2)/2\ ) \exp(\omega\pi\alpha\omega(x \cos\theta + y \sin\theta))$ (14)
where $\alpha=1/2^{(1/2)}$, $\omega=0,1,2,\ldots$, $\theta=[0,2\pi]$

On a trial and error basis, it is found that the filter provide consistent and effective results for values of $\omega = 2$ and $\theta = 5\pi/3$. The different choices of scale and orientation components construct a set of filters. The set of 6 filters is constructed using three scales and two orientations. To reduce the computational load, the filter-banks should be made as small as possible. The following is a sample cell array of the Gabor filter bank in three scales and two orientations that is a 3 x 2 array, computed using the above formula{[1.2682 0.1991] [16.8892 2.6517] [288.7979 45.3932]} .

2.3.3 Convolution

Once a series of Gabor filters have been chosen, image features at different locations, frequencies and orientations can be extracted by convolving the image i(x,y) with the filters using the formula in Equation (15)

$$m(x,y) = L_h\,(i(x,y)) = i(x,y) * h(x,y) \qquad (15)$$

The filter bank is applied to the input image and the mean and variance of the filtered image is obtained.

2.4 Relevant Features and Assigning Weights

After extracting low level features the next step is to assign weight to the features that are relevant to classification. Some of the image features are more relevant to the class than the others. It is necessary to identify the relevant features so that the calculation of kernel function of the support vector machine is not dominated by the irrelevant features. The weights of the features are calculated using the principle of maximizing deviations.

2.4.1 Principle of Maximizing Deviations

The weight of a feature with respect to class is calculated as illustrated in [3]. Consider two classes A and B. The feature vector of a sample that belongs to A and B are given as:

$A=(a_1, a_2,\ldots\ldots a_n)$
$B=(b_1, b_2,\ldots\ldots b_n)$

If the difference between p^{th} (p=1,2,….n) feature a_p and b_p of two samples that belongs to A and B is more, then that feature plays an important role in classification. So, that feature is assigned greater weight than the features which has less difference. So, the feature with greater deviation should be given greater weight than the feature with smaller deviation. Each feature is a random variable. So, the deviation of random variables a_p and b_p is given in Equation (16)

$$d(a_p, b_p) = \iint_{-\infty}^{+\infty}|a_p - b_p|\, f_p(a_p, b_p)\,da_p\,db_p \qquad (16)$$

where $f_p(a_p,b_p)$ is joint probability density function of random variables a_p and b_p. The same feature's values of different samples are independent, thus we have

$$f_p(a_p,b_p)=f_p(a_p)f_p(b_p) \qquad (17)$$

Now Equation (16) becomes

$$d(a_p, b_p) = \iint\limits_{-\infty}^{+\infty} |a_p - b_p| f_p(a_p) f_p(b_p) da_p db_p \quad (18)$$

The deviation between categories of samples is given in Equation (19)

$$D(\lambda_p) = \sum_{p=1}^{n} \lambda_p d(a_p, b_p) \quad (19)$$

Structure the model of maximizing deviations between categories as given in Equation (20):

$$\max D(\lambda_p) = \sum_{p=1}^{n} \lambda_p d(a_p, b_p) \quad (20)$$

$$\sum_{p=1}^{n} \lambda_p^2 = 1, \lambda_p \geq 0, p = 1,2,..n. \quad (21)$$

Using Lagrangian function method to solve this model, weight of each feature is obtained as given in Equation (22)

$$\lambda_p = \frac{d(a_p, b_p)}{\sum_{p=1}^{n} d(a_p, b_p)} \quad (22)$$

2.4.2 Precomputation of Linear Kernel Matrix

In precomputed kernel, the kernel values are computed using the kernel function, here linear kernel function is used in analysis [16]. The precomputed kernel matrix is used in training and testing files. In that case, the SVM does not need the original training and testing files. Assume there are L training instances x_1, x_2, …x_L. Let $K(x,y)$ be the kernel value of two instances x and y. The input formats of training and testing files are:

New training instances for x_i:

<label> 0:i 1:$K(x_i,x_1)$……………L:$K(x_i,x_L)$

New testing instances for any x:

? 0:1 1:$K(x,x_1)$……………L:$K(x,x_L)$

That is, in the training file the first column must be the class label of x_i. In testing, ? can be any value. All kernel values including zeros must be explicitly provided. Any permutation or random subsets of the training/testing files are also valid. The calculation of precomputed kernel is explained with an example.

Example

Assume the original training data has three four feature instances (x_1, x_2 and x_3) and testing data has one instance.

Training set:

15	1:1	2:1	3:1	4:1
45		2:3		4:3
25			3:1	

Testing set:

15	1:1		3:1	

If the linear kernel, $K(x_i,x_j) = x_i^T.x_j$ is used, then the new training/testing sets will be calculated as:

15	0:1	1:$K(x_1,x_1)$	2:$K(x_1,x_2)$	3:$K(x_1,x_3)$
45	0:2	1:$K(x_2,x_1)$	2:$K(x_2,x_2)$	3:$K(x_2,x_3)$
25	0:3	1:$K(x_3,x_1)$	2:$K(x_3,x_2)$	3:$K(x_3,x_3)$

The individual parameters of kernel matrix for the instance x_1 are computed as:

$K(x_1,x_1)$=1*1 + 1*1 + 1*1 + 1*1 = 4

$K(x_1,x_2)$=1*0 + 1*3 + 1*0 + 1*3 = 6

$K(x_1,x_3)$= 1*0 +1*0 + 1*1 + 1*0 = 1

Kernel matrix constructed using linear kernel for all the three instances of the training set:

15	0:1	1:4	2:6	3:1
45	0:2	1:6	2:18	3:0
25	0:3	1:1	2:0	3:1

For Precomputed kernel without weights, the kernel matrix calculated as above is used as such for training and testing but for WFSVM the diagonal of the kernel matrix i.e. $K(x_1,x_1)$, $K(x_2,x_2)$ and $K(x_3,x_3)$ is replaced with the weights of the features calculated using Equation (22).

2.5 Classification

The classification of images is implemented and validated with:

i. Traditional SVM using different kernel types
ii. WFSVM using precomputed Linear Kernel with relevant weights in the diagonal of kernel matrix and without weights substitution.

For both approaches, the difference lies in choosing the kernel function or constructing the kernel function only. But the classification algorithm given below remains same. The classification of images into their category includes two phases: Training phase and Testing phase.

2.5.1 Training Phase

In this phase, from the training images, the low level features are extracted. Using the features extracted from the training images kernel matrix is constructed. Linear kernel is used to construct the kernel matrix. Binary SVM is used for classification. The algorithm for SVM is as follows:

i. Input sample set $T = \{ (x_i, y_i) \}_{i=1 \text{ to } l}$ where x_i is the feature vector and y_i is the classes.
ii. Construct the kernel matrix using the features.
iii. Select appropriate penalty parameter and positive component.
iv. Structure decision function using Equation (23)

$$f(x) = sgn\left(\sum_{i=1 \text{ to } l} y_i \alpha_i^* K(x_i, x) + b^* \right) \quad (23)$$

where b^* is the positive component.

The algorithm for SVM based on weighted feature (WFSVM) [5] is the same as Traditional SVM but the diagonal of the kernel matrix is replaced with the weights of the features.

2.5.2 Testing Phase

The trained WFSVM and SVM are tested with the testing set. From the testing set 24 features are extracted, kernel matrix is constructed without weights in the diagonal of the matrix and then it is given as input to the WFSVM and SVM for classification. For traditional SVM 2/3 of the data is used for training and 1/3 is used for testing but here 1/2 of the data is used for training and testing as the SVM is trained using weighted features.

2.6 Performance Evaluation

The performance of the system is measured using the quantitative metrics such as Sensitivity, Specificity and Accuracy defined in Equations (24-26)

$$\text{Sensitivity} = TP / (TP + FN) \quad (24)$$

$$\text{Specificity} = TN / (TN+TP) \quad (25)$$

$$\text{Accuracy} = (TP+TN) / (TP + TN + FP + FN) \quad (26)$$

where

TP (True Positive) - correctly classified positive cases.
TN (True Negative) - correctly classified negative cases.
FP (False Positive) - incorrectly classified negative cases.
FN (False Negative) - incorrectly classified positive cases.

3. RESULTS AND PERFORMANCE EVALUATION

The result of classification techniques such as SVM and WFSVM are analyzed from the feature set of color and texture features for categories Africa and Beach with quantitative measures. The sample image from categories Africa and Beach is shown in Figure 2. The input images are segmented using Fuzzy C Means clustering and the segmentation is validated using the validity function index (S). The sample image after segmentation is shown in Figure 3. After segmentation, twelve color features and twelve texture features are extracted from the segmented image. From the feature space SVM and WFSVM are performed.

Africa **Beach**

Figure 2: COREL Image Classes

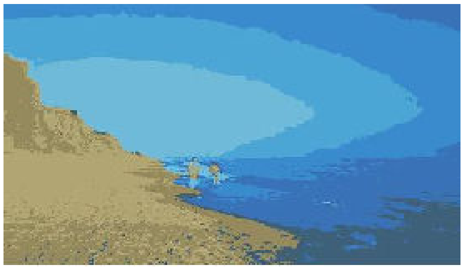

Figure 3: Segmented image from Beach category

For traditional SVM, the training feature space is used to build a train model file and then the test features are validated. This analysis is carried out for different kernel functions where 1/2 of the samples are used for training and testing as given in Table 1.

From the results, it can be inferred that the accuracy of RBF kernel and linear kernel for color and texture features is greater than the accuracy obtained from polynomial kernel.

For WFSVM, the constructed kernel matrix from the training and test feature space with equal number of samples are used in validating the metrics such as sensitivity, specificity and accuracy, as given in Table 2. From the results, it is understood that the Precomputed kernel with color and texture features resulted with high classification accuracy.

In Table 3 the values of parameters (C, γ) are chosen to validate the number of support vectors created for the feature space which resulted high accuracy. In Table 4, the classification result of another binary class (Elephant Vs Horse) with color and texture features is considered for experiment and it has been proved that the number of support vectors for WFSVM is less than the traditional SVM. Also, the accuracy for WFSVM is higher when compared to traditional SVM.

Table 1. Performance Analysis of SVM

Feature type	Kernel type	Testing(100) misclassified samples		Accuracy in %
		A(50)	B (50)	
Color features	Lin	0	8	92
	Poly	0	9	91
	RBF	0	8	92
Texture features	Lin	0	4	96
	Poly	0	5	95
	RBF	0	4	96
Color and Texture features	Lin	0	3	97
	Poly	0	3	97
	RBF	0	2	98

Lin – Linear Poly – Polynomial RBF – Radial Basis Function
A – Africa B – Beach

Table 2. Performance Analysis of WFSVM

Feature type	Approach used – Linear kernel	Testing (100) - Misclassified samples- Africa-50 Beach-50				Accuracy	Sensitivity	Specificity
		TP	TN	FP	FN			
Color features	WFSVM without weights	50	11	39	0	61	100	22
	WFSVM with weights	50	46	4	0	96	100	92
Texture features	WFSVM without weights	50	27	23	0	77	100	54
	WFSVM with weights	50	42	8	0	92	100	84
Color and texture features	WFSVM without weights	49	50	0	1	99	98	100
	WFSVM with weights	50	49	1	0	99	100	98

Table 3. Comparison of Traditional SVM and WFSVM for TA and NSV

Table	Traditional SVM		WFSVM	
	TA(%)	NSV	TA(%)	NSV
1	98	22	99	16
2	97	12	99	6
3	97	10	98	4

TA – Testing Accuracy
NSV – Number of Support Vectors

Table 4. Values of (C , γ)

Table	1	2	3
C	1	5	12
γ	100	0.5	0.4

4. CONCLUSION AND FUTURE WORK

In this paper, the efficiency of classifying the COREL images is analyzed using WFSVM and validated with Traditional SVM. The training and test images are segmented and 24 low level features are extracted. From the extracted features of training set, relevant features are determined and weights are assigned using the principal of maximizing deviations between classes. Then the kernel matrix is precomputed using the linear kernel and its diagonal is constructed with the weights of the features. The WFSVM is trained using this precomputed kernel and a model file is created. For testing, the linear kernel matrix is precomputed using the extracted features of the test dataset and which is given as input to the model file for validation. The performance of WFSVM is compared with traditional SVM and it is proved that the WFSVM is better than traditional SVM. In future research, the WFSVM can be extended to resolve a multiclass classification problem. With SVM other learning techniques can also be incorporated to achieve the task of ensemble learning and automatic understanding in images.

5. REFERENCES

[1] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta and R. Jain, “Content-based Image Retrieval at the end of the Early Years”, on Pattern Analysis and Machine Intelligence, IEEE Trans. volume 12, pp. 1349-1380, 2000.

[2] Amine M. Bensaid, Lawrence O. Hall, James C. Bezdek, Laurence P. Clarke, Martin L. Silbiger, John A.Arrington and Reed F. Murtagth, “Validity-Guided (Re)Clustering with Application to Image Segmentation”, IEEE Trans.on Fuzzy Systems, vol. 4, pp. 112-123, 1996.

[3] Bo Sun, Shi-Ji Song and Cheng Wu, “A New Algorithm of Support Vector Machine based on Weighted Feature”, In Machine Learning and Cybernetics, 2009 International Conference on vol. 3, pp. 1616-1620, 2009.

[4] I. Gath and A.B. Geva, “Unsupervised Optimal Fuzzy Clustering”, IEEE Trans. Pattern Anal. Machine Intell., vol. 11, No. 7, pp. 773-781, 1989.

[5] II Hong Suh, Jae-Hyun Kim and Frank Chung-Hoon Rhee, “Convex-Set-Based Fuzzy Clustering”, IEEE Trans. on Fuzzy Systems vol. 7, No. 3, pp. 271-285, 1999.

[6] Keping Wang, Xiaojie Wang and Yixin Zhong, “A Weighted Feature Support Vector Machines Method for Semantic Image Classification”, in Measuring Technology and Mechatronics Automation (ICMTMA), 2010 International Conference on vol.1, pp. 377-380, 2010.

[7] L. Wang, L. Khan, L. Liu and W. Wu, “Automatic Image Annotation and Retrieval using Weighted Feature Selection”, in Multimedia Software Engineering, 2004 Proceedings on IEEE Sixth International Symposium, pp. 435-442, 2004.

[8] M. Sugeno and T. Yasukawa, “A Fuzzy-Logic-Based Approach to Qualitative Modeling", IEEE Trans. Fuzzy Syst., vol. 1, pp. 7-31, 1993.

[9] Metin KAYA , “An Algorithm for Image Clustering and Compression,” Turk J Elec Engin, vol. 13, 2005.

[10] Simon Tong and Edward Chang, “Support Vector Machine Active Learning for Image Retrieval”, in Proceedings of the ninth ACM International Conference on Multimedia, vol.1, pp. 107-118, New York, USA, 2001.

[11] Stuart Andrews, Ioannis Tsochantaridis and Thomas Hofmann, “Support Vector Machines for Multiple-Instance Learning”, in Advances in Neural Information Processing Systems, vol. 15, pp. 561-568. MIT Press, 2003.

[12] Tianxia Gong, Shimiao Li and Chew Lim Tan, “A Semantic Similarity Language Model to Improve Automatic Image Annotation”, in Tools with Artificial Intelligence (ICTAI), 2010 22nd IEEE International Conference on, vol. 1, pp. 197-203, 2010.

[13] Xuanli Lisa Xie and Gerardo Beni, “A Validity Measure for Fuzzy Clustering”, IEEE Trans. on Pattern Anal. Machine Intell., vol. 13, No.8, pp. 841-847, 1991.

[14] Zhang Qilong, Shan Ganlin and Duan Xiusheng, “Weighted Support Vector Machine based Clustering Vector”, Computer Science and Software Engineering, International Conference on vol. 1, pp. 819-822, 2008.

[15] COREL database http://corel.digitalriver.com/

[16] LibSVM www.csie.ntu.edu.te/~cjlin/libsvm/