ijournals International Journal of Software & Hardware Research in Engineering
ISSN No: 2347-4890 Volume 2 Issue 5, May 2014

Hybrid Features Extraction Method for Natural Scenes Categorization using ANN and SVM Kernels

Devendran V[1], Nagarajan A[2], P. Babu[3]

[1]Professor, Department of MCA, PSNA College of Engineering and Technology, Dindigul,Tamil Nadu India

[2]Associate Professor, Department of MCA, PSNA College of Engineering and Technology, Dindigul,Tamil Nadu India

[3]Associate professor, Dept of MCA ,PSNA College of Engineering and technology , dindigul

Abstract

Scene classification, the classification of images into semantic categories (e.g., coast, mountains, highways and streets) is a challenging and important problem nowadays. Many different approaches and feature extraction methodologies concerning scene classification have been proposed and applied in the last few years. In real time environments, we prefer a feature extraction method which helps us with minimal data, performing better with less execution time. In this aspect, we are proposing a hybrid feature extraction method which includes statistical and texture features for natural scene categorization problems. The results are proving the efficiency of the proposed feature over the commonly used feature extraction methods such as haar, geometrical moments, statistical moments, texture features, co-occurrence features, Zernike moments. These features are tested using polynomial and radial basis kernel functions in support vector machine. The kernel results are also compared with traditional backpropagation algorithm using artificial neural networks. Images are used, as it is, without any preprocessing, making the system robust to real scene environments. This complete work is experimented in Matlab 6.5 using real world dataset.

Keywords

Artificial Neural Networks, Scene Categorization, Feature Extraction and Support Vector Machines.

1. Introduction

In the machine learning literature [1], the term "natural scene" is usually intended as the one of a semantically coherent, namable human-scaled view of an outdoor real world environment, and the term natural scene categorization refers to the task of grouping images into semantically meaningful categories. Understanding the robustness and rapidness of human scene categorization has been a focus of investigation in the cognitive sciences over the last decades [2][3][4]. At the same time, progress in the area of image understanding has prompted computer vision researchers to design computational systems that are capable of automatic scene categorization. Classification is one of several primary categories of machine learning problems [5]. Papers [6], [7] and [8] give very promising results in the classification of indoor-outdoor scene image and manmade-natural classification. Ian Stefan Martin [9] presents in his doctoral work, the techniques for robust learning and segmentation in scene understanding. In [10], Manuele Bicego et al. give a new approach to scene analysis under unsupervised circumstances. In [11][12], Bosch et al. present a scene description and segmentation system capable of recognizing natural objects (e.g., sky, trees, grass) under different outdoor conditions. Texture is a commonly used feature in the analysis and interpretation of images. Lance M. Kaplan [13] worked on mosaic images using extended self similar model (ESS) and k-means classification algorithm. Arivazhagan *et al.*, [14][15] worked on classification of mosaic images using statistical features from Ridgelet and Curvelet Transformed Images. G. Y. Chen and P. Bhattacharya [16] worked on invariant texture classification using Ridgelet packets, Fourier transformation and db4 with nearest neighbor classifier. Therefore, the research presented in this article focuses on texture analysis for scene categorization using Artificial Neural Networks (ANN) and Support Vector Machines (SVM). In my previous work [17], five levels of wavelet decomposition is applied for classification of images using neural classifier and support vector machines. This proves the efficiency of radial basis kernel function in terms of consist performance. In the previous work [18][19], Invariant Moment Features are tested in the scene classification with Neural classifier and Support Vector

IJournals International Journal of Software & Hardware Research in Engineering
ISSN No: 2347-4890 Volume 2 Issue 5, May 2014

Machines. In our work [20], haar wavelet, invariant moments and co-occurrence matrices features are tested for their consistency over various binary class problems in natural scenes categories. This shows that co-occurrence features are giving average classification rate of 73.5 % over six various binary classification problems. Hence, this paper presents the natural scene classification using co-occurrence matrix features with polynomial and radial basis kernel function using support vector machines. The results are also compared with the traditional neural networks with backpropagation algorithm. The organization of the paper is as follows: Section 2 describes Artificial Neural Networks, Section 3 describes Support Vector Machine, Section 4 deals with Feature Extraction Methods, Section 5 explains the Scene Categorization, Section 6 deals with Discussion, and finally Section 7 concludes with conclusion.

2. Artificial Neural Networks

The Neural networks developed from the theories of how the human brain works [21]. Many modern scientists believe the human brain is a large collection of interconnected neurons. These neurons are connected to both sensory and motor nerves. Scientists believe, that neurons in the brain fire by emitting an electrical impulse across the synapse to other neurons, which then fire or don't depending on certain conditions.

The Artificial neural network is basically having three layers namely input layer, hidden layer and output layer. There will be one or more hidden layers depending upon the number of dimensions of the training samples as shown in Fig 1.

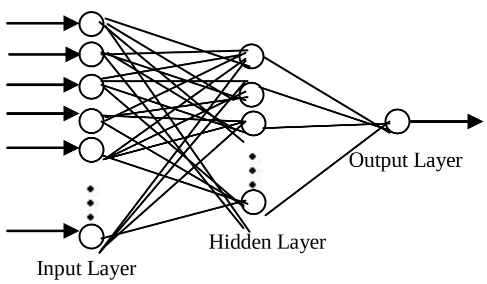


Fig. 1 Simple Neural network Structure

A learning problem with binary outputs (yes / no or 1 / 0) is referred to as binary classification problem whose output layer has only one neuron. A learning problem with finite number of outputs is referred to multi-class classification problem whose output layer has more than one neuron. The examples of input data set (or sets) are referred to as the training data. The algorithm which takes the training data as input and gives the output by selecting best one among hypothetical planes from hypothetical space is referred to as the learning algorithm. The approach of using examples to synthesize programs is known as the learning methodology. When the input data set is represented by its class membership, it is called supervised learning and when the data is not represented by class membership, the learning is known as unsupervised learning. There are two different styles of training .ie, Incremental Training and Batch training. In incremental training the weights and biases of the network are updated each time an input is presented to the network. In batch training the weights and biases are only updated after all of the inputs are presented. In this experimental work; Back propagation algorithm is applied for learning the samples, Tan-sigmoid and log-sigmoid functions are applied in hidden layer and output layer respectively, Gradient descent is used for adjusting the weights as training methodology.

3. Support Vector Machines

Support vector machine is a relatively new pattern classifier introduced by Vapnik [22]. A SVM classifies an input vector into one of two classes, with a decision boundary developed based on the concept of structural risk minimization (of classification error) using the statistical learning theory. The SVM learning algorithm directly seeks a separating hyperplane that is optimal by being a maximal margin classifier with respect to training data. For non-linearly separable data, the SVM uses kernel method to transform the original input space, where the data is non-linearly separable, into a higher dimensional feature space where an optimal linear separating hyperplane is constructed. On the basis of its learning approach, the SVM is believed to have good classification rate for high-dimensional data. Consider the problem of image classification where X is an input vector with 'n' dimensions. The SVM performs the following operation involving a vector $W = (w_1,..., w_n)$ and scalar b:

$$f(X) = \mathrm{sgn}(W \bullet X + b) \qquad (1)$$

Positive sign of f(X) may be taken as '*MIT-street*' images and negative value of f(X) may be regarded as '*MIT-highways*' images. Consider a set of training data with l data points from two classes. Each data is denoted by (X_i, y_i), where i=1, 2,..., l, $X_i = (x_{i1},..., x_{in})$, and $y_i \in \{+1, -1\}$. Note that y_i is a binary value representing the two classes. The task of SVM learning algorithm is to find an optimal hyperplane (defined by W and b) that separates the two classes of data. The hyperplane is defined by the equation:

$$W \bullet X + b = 0 \qquad (2)$$

Where X is the input vector, W is the vector perpendicular to the hyperplane, and b is a constant. The graphical representation for a simple case of two-dimensional input (n=2) is illustrated in Fig. 3. According to this hyperplane, all the training data must satisfy the following constraints:

$$W \bullet X_i + b \geq +1 \text{ for } \forall_i = +1$$

$$W \bullet X_i + b \geq -1 \text{ for } \forall_i = -1 \qquad (3)$$

which is equivalent to :

$$y_i(W \bullet X_i + b) \geq 1 \;\; \forall_i = 1, 2, \ldots, l \qquad (4)$$

There are many possible hyperplanes that separate the training data into two classes. However, the optimal separating hyperplane is the unique one that not only separates the data without error, but also maximizes the margin, i.e., maximizes the distance between the closest vectors in both classes to the hyperplane [23]. As shown in Fig. 2, the margin, ρ, is the sum of the absolute distance between the hyperplane and the closest data points in each class. It is given by:

$$\rho = \min \frac{|W.X_i + b|}{\|W\|} + \min \frac{|W.X_i + b|}{\|W\|} = \frac{2}{\|W\|} \qquad (5)$$

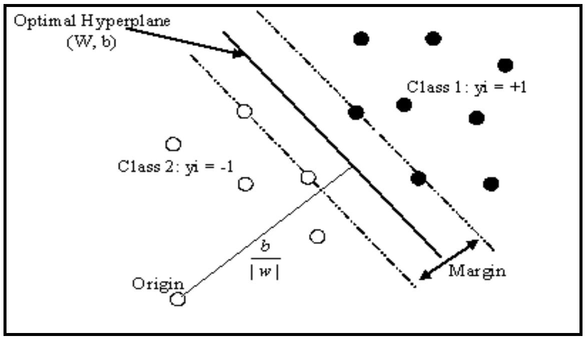


Fig. 2 Optimal separating hyperplane for 2-Dimensional two-class problem

Here, the first min is over X_i of one class and the second min is over X_i of the other class. Therefore, the optimal separating hyperplane is the one that maximizes $2/||W||$, subject to constraints (4). It is mathematically more convenient to replace maximization of $2 / ||W||$ with the equivalent minimization of $||W||^2/ 2$ subject to constraints (4), which can be solved by the Lagrangian formulation:

$$\min \; L = \frac{1}{2}\|W\|^2 - \sum_{i=1}^{l} \alpha_i \left[y_i (W.X_i + b) - 1 \right] \qquad (6)$$

where α_i is the Lagrange multiplier ($\alpha_i >= 0$, i=1,2,.., l). The Lagrangian has to be minimized with respect to W and b, and maximized with respect to α_i. The minimum of the Lagrangian with respect to W and b is given by:

$$\frac{\partial L}{\partial W} = 0 \Rightarrow W = \sum_{i=1}^{l} \alpha_i X_i y_i \qquad (7)$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^{l} \alpha_i y_i = 0 \qquad (8)$$

Substituting (7) and (8) into (6), the primal minimization problem is transformed into its dual optimization problem of maximizing the dual Lagrangian L_D with respect to α_i :

max L_D =

$$\sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j (X_i \bullet X_j) \qquad (9)$$

subject to

$$\sum_{i=1}^{l} \alpha_i y_i = 0 \qquad (10)$$

$$\alpha_i \geq 0 \;\; \forall \;\; i=1,\ldots, l \qquad (11)$$

Thus, the optimal separating hyperplane is constructed by solving the above quadric programming problem defined by (9)-(11). In this solution, those points have non-zero Lagrangian multipliers ($\alpha_i > 0$) are termed support vectors. Support vectors satisfy the equality in the constraint (4) and lie closest to the decision boundary (they are circles in Fig. 2, lying on the dotted lines on either side of the separating hyperplane). Consequently, the optimal hyperplane is only determined by the support vectors in the training data. Based on the α_i values obtained, W can be calculated from (7). b can be obtained by using the Karush-Kuhn-Tucker(KKT) complementary condition for the primal Lagrangian optimization problem:

$$\alpha_i \left[y_i (W.X_i + b) - 1 \right] = 0 \;\; \forall_i = 1,.., l \qquad (12)$$

One b value may be obtained for every support vector (with $\alpha_i > 0$). Burges[16] recommends that the average value of b be used in the classification. With this solution, the SVM classifier becomes

$$f(X) = \operatorname{sgn}(W \bullet X + b) = \operatorname{sgn}\Big(\sum_{\forall i, \alpha_i > 0} y_i \alpha_i (X_i \bullet X) + b \Big)$$

(13) Note that, in (13), one only needs to make use of X_i, y_i and α_i of the support vectors, while X is the input vector to be classified. When a linear boundary is inappropriate (i.e., no hyperplane exists to separate the two classes of data), the extension of above method to a more complex decision boundary is accomplished by mapping the input vectors $X \in R^n$ into a higher dimensional feature space H through a non-linear function $\phi : R^n \rightarrow H$. In H, an optimal separating hyperplane is then constructed using training data in the form of dot products $\phi(X_i) \bullet \phi(X_j)$ instead of the $X_i \bullet X_j$ term in (9). To avoid the expensive computations of $\phi(X_i) \bullet \phi(X_j)$ in the feature space, it is simpler to employ a kernel function such that

$$K(X_i . X_j) = \phi(X_i) \bullet \phi(X_j) \qquad (14)$$

Thus, only the kernel function is used in the training algorithm, and one does not need to know the explicit form of ϕ. The computation in (15) results in some restrictions on the form and parameter values of non-linear functions that can be used as the kernel functions.

Detailed discussions can be found in [22] and [23]. Some commonly used kernel functions are:

Polynomial function: $K(X_i, X_j) = \left(X_i . X_j + 1\right)^d$ (15)

Radial basis function: $K(X_i, X_j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}$ (16)

Sigmoid function $K(X_i, X_j) = \dfrac{1}{1 + e^{[v(X_i . X_j) - \delta]}}$ (17)

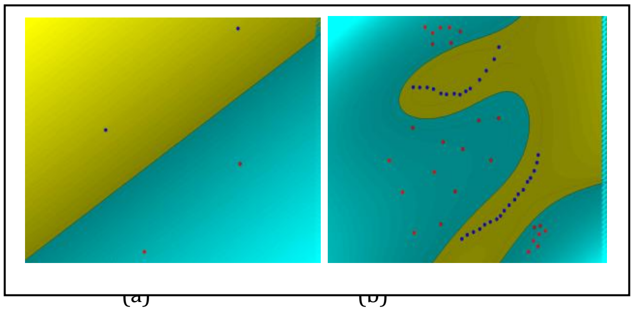


Fig. 3 Representation of (a) linearly separable (b) non-linearly separable

Where d is a positive integer, and, σ , v and δ are real constants. These four parameters must be defined by the user prior to SVM training. With the use of a kernel function, the SVM capable of performing non-linear classification of input X becomes,

$$f(X) = \mathrm{sgn}\left(\sum_{\forall_i, \alpha_i > 0} y_i \alpha_i K(X_i, X) + b \right) \quad (18)$$

The hyperplane and support vectors used to separate the linearly separable data are shown in Fig. 4 (a). And the hyperplane and support vectors used to separate the non-linearly separable data are shown in Fig. 4 (b). Radial basis kernel function with p1=5 used for this non-linear classification. Individual colors represents particular each class of data.

4. Feature Extraction

There are many motivations for using features rather than the pixels directly. The most common reason is that feature extraction is used to reduce the dimension of the input data and in turn helps to minimize the training time taken for network classifier. Image feature detection is a fundamental issue in many intermediate level vision problems such as stereo, motion correspondence, image registration and object recognition as shown in the Fig. 4.

.

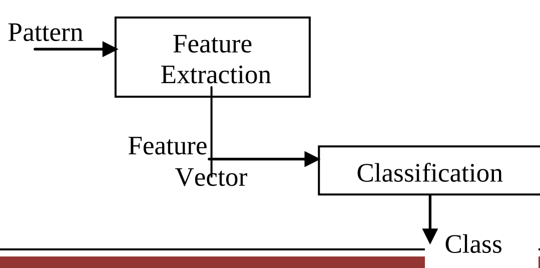


Fig. 4 Basic steps involved in classification

Table 1 Feature Method and the features extracted

Feature Extraction Methodology	Features Extracted (in Nos)
Haar Wavelet	128
Geometrical Moments	28
Statistical Moments	10
Texture Moments	6
GLCM	32
Zernike Moments	576
Hybrid Moments	16

Haar features, Statistical moments, Texture Features, GLCM Features, Invariant Moments, Zernike Moments, Hybrid features (which combines statistical moments and texture moments) are used to extract features from the scenes. The detailed explanation is discussed below in Sections A to F. Number of feature data extracted from the scenes using all the mentioned feature extraction methodologies are given in the Table 1.

A. Haar Wavelet

Haar wavelet [24] is widely used technique for the feature extraction, which is single-level one-dimensional wavelet decomposition and gives both an approximation and detailed coefficients as shown in the Fig. 5. Approximation coefficients which are of size 128x1 are considered for the training of the neural classifier.

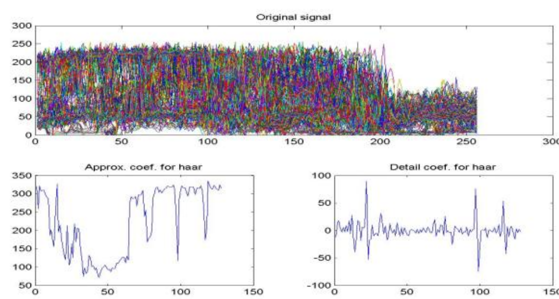


Fig. 5 Original, approximation and detailed coefficients of Haar wavelets

B. Geometrical Moments

Hu [25] defines seven values, computed by normalizing central moments through order three, that are invariant to object scale, position and orientation. In terms of the normalized central moments, the seven moments are given below:

$$\phi_1 = \eta_{20} + \eta_{02}$$
$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}{}^2$$
$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$
$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$
$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$
$$\phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$
$$\phi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

C. Statistical Moments

We compute statistical central moments of image histogram. We compute upto 10 statistical moment of histogram whose components are in vector P. The length of P must equal to 256. The features F1 to F10 are extracted as follows:

F1 = Mean
F2 = Variance
F3 = 3rd Order moment
F4 = 4th Order moment
. . .
F10= 10th Order Moment.

D. Texture Features

We compute statistical measures of texture in an image. The features F1 to F6 are extracted as follows:

F1 = Average gray level
F2 = Average contrast
F3 = Measure of smoothness
F4 = Third Order moment
F5 = Measure of uniformity
F6 = Entropy

E. GLCM Features

We base our texture feature extraction on the spatial gray-level cooccurrence-matrix (SGLCM). More specifically, a SGLCM gives a joint histogram of the quantized gray-level value pairs of two image pixels bearing a certain spatial relationship. Eight texture features F1 to F8 are extracted as follows:

F1= energy
F2=inertia
F3=entropy
F4=homogeneity
F5=maximum probability
F6=contrast
F7=inverse
F8=correlation

All the eight features are extracted on each co-occurrence matrix derived from each of the angular directions, 0^0, 45^0, 90^0 and 135^0 degrees. So, total of 32 features are computed for each image as eight features for each angle is considered.

F. Zernike Moments

Zernike polynomials were first proposed in 1974 by Zernike. Their moment formulation appears to be one of the most popular, outperforming the alternatives (in terms of noise resilience, information redundancy and reconstruction capability). Moments have been widely used in image processing applications through the years. Geometrical, central and normalized moments were for many decades the only family of applied moments. The main disadvantages of these descriptors were their disability to fully describe an object in a way that, using the moments set, the reconstruction of the object could be possible. In other words they weren't orthogonal. The kernel of Zernike moments is a set of orthogonal Zernike polynomials defined over the polar coordinate space inside a unit circle. The computation is followed from [26] and it gives 32 features. We have divided each image into 16 blocks (as the performance is better if the images are divided into 16 blocks) and 32 features are extracted from every block. Hence, 512 features are used to represent an image.

5. Scene Categorization

In the case of computer vision, the examples are representations of photographic images and the task of the classifier is to indicate whether or not a specific object or phenomena of interest is present in the image. In order to successfully accomplish this, the classifier must have sufficient prior knowledge about the appearance of the object. Natural scene classification system is trained to recognize a type of example or differentiate between examples that fall in separate categories. This paper is trying to recognize the scenes of two different categories called '*MIT-street*' and '*MIT-highways*'. Since performance of classification of scenes should robust to real world circumstances, we have not used any preprocessing techniques such as noise removal, enhancements, smoothening. Feature extraction techniques are applied directly with the raw images. Various feature extraction methods are tested in our proposed work of natural scene categorization problem. Haar features, Statistical moments, Texture Features, GLCM Features, Invariant Moments, Zernike Moments are used to extract features from the scenes. Polynomial kernel function and radial basis kernel functions are used for classification of natural scenes. The result is compared with the traditional neural networks with backpropagation algorithms. On each classification, 200 samples are used for the training phase and another set of 200 samples are used for testing phase separately without overlapping including 100 samples from 'MIT-Street' as positive sample and 100 samples from 'MIT-highways' as negative sample. Normalization is then applied using Zero-mean normalization method in order to maintain the data within the specified range and to improve the performance of the classifier. The detailed description of our proposed work is shown in Fig.6.

ijournals International Journal of Software & Hardware Research in Engineering
ISSN No: 2347-4890 Volume 2 Issue 5, May 2014

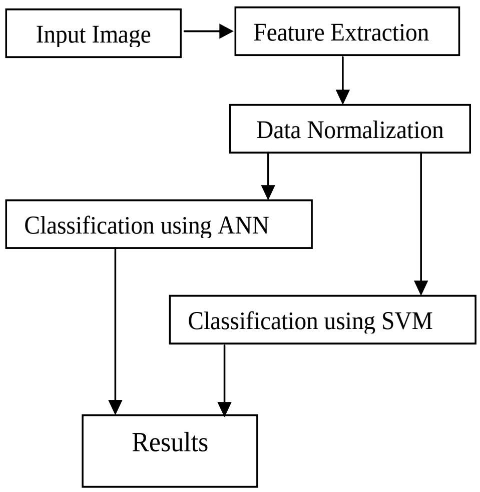


Fig. 6 Detailed description of proposed work

A. The Database

The sample images of scenes are taken from the Ponce Research Group [27]. This database contains 15 different scene categories, containing 250 samples in each categories. All the images are gray scale and of size (256x256) pixels. The sample images of natural scene categories are given in Fig. 7, 8, 9, 10 and 11.

B. Data Normalization

Normalization is then applied using Zero-mean normalization method in order to maintain the data within the specified range and also found suitable to improve the performance of the classifier.

Zero-Mean Normalization: By using this type of normalization, the mean of the transformed set of data points is reduced to zero. For this, the mean and standard deviation of the initial set of data values are required. The transformation formula is v' = (v - meanA) / std_devA where meanA and std_devA are the mean and standard deviation of the initial data values.

The following are the other techniques normally used by our research community for normalization of data.

Fig. 7 Sample images of '*MIT-highway*' category

Fig. 8 Sample images of 'MIT-Street' category

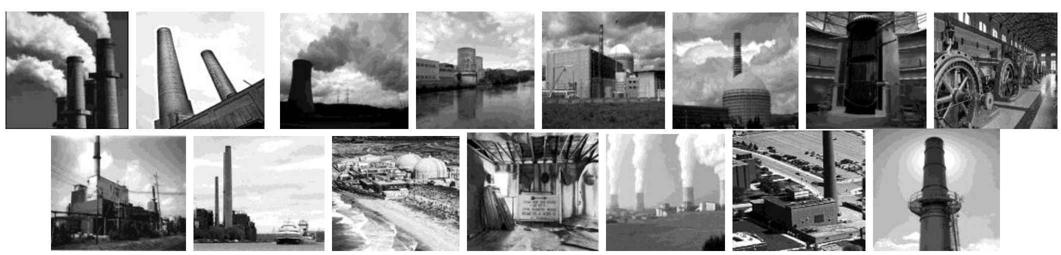
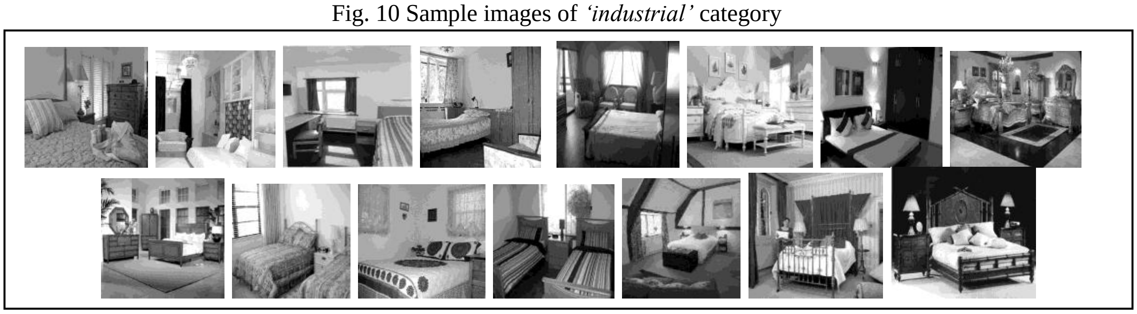
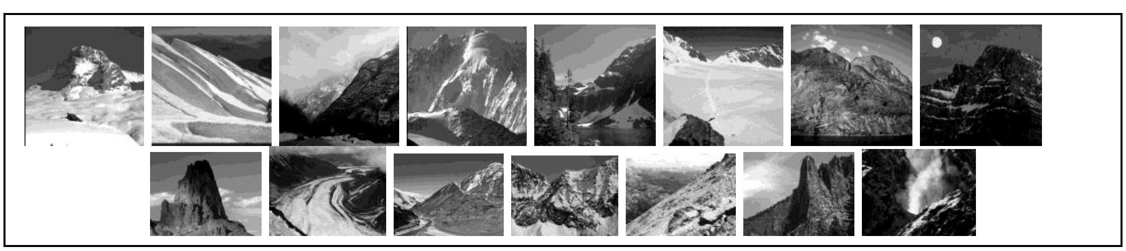


Fig. 9 Sample images of 'industry' category

Fig. 10 Sample images of *'industrial'* category





ijournals International Journal of Software & Hardware Research in Engineering
ISSN No: 2347-4890 Volume 2 Issue 5, May 2014

Decimal Scaling: This type of scaling transforms the data into a range between [-1,1]. The transformation formula is $v'(i) = v(i)/10^k$ for the smallest k such that max (|v'(i)|) ≤ 1.

Min-Max Normalization: This type of normalization transforms the data into a desired range, usually [0,1]. The transformation formula is $v'(i) = (v(i) - minA) / (maxA-minA) * (new\_maxA- new\_minA) + new\_minA$ where, [minA, maxA] is the initial range and [new_minA, new_maxA] is the new range.

C. Neural Classification

Backpropagation algorithm is used to train and test the neural classifier with all the extracted features individually. The neural structure is normalized such that it gives the maximum performance. The detailed results are given in Table 2.

Table 2 Performance using Artificial Neural Networks

Feature Extraction Method	TP	FP	%	Exec. Time
Haar	70	60	65	124
Geometrical Moments	80	87	83.5	110.5
Statistical Moments	68	74	71	97.2
Texture Moments	69	89	79	265.3
GLCM	**93**	**92**	**92.5**	**770.5**
Zernike	82	96	89	963.3
Hybrid	**85**	**95**	**90**	**53.7**

GLCM produces 92.5% of classification rate which is highest performance among all the features used in 770.50 seconds. But, hybrid feature produces 90.0% of classification rate which is achieved in 53.50 seconds of execution time. The comparative graph is given in Fig. 12.

D. Classification using Polynomial Kernel Function

Polynomial Kernel Function (degree 2): This gives the highest performance of 90.0 % with 187.8 seconds with the Gray Level Cooccurrence Matrix features. Hybrid feature is giving the performance level of 84.5 % with 19.4 seconds of execution time. The detailed results with all the features are given in Table 3. The comparative graph is given in Fig. 13.

Polynomial Kernel Function (degree 5): This gives the highest performance of 93.0 % with 184.5 seconds with the Gray Level Cooccurrence Matrix features. Hybrid feature is giving the performance level of 88.0% with 18.7 seconds of execution time. The detailed results with all the features are given in Table 4. The comparative graph is given in Fig. 14.

Table 3 Performance using Polynomial function (Degree 2)

Feature Extraction Method	TP	FP	%	Exec. Time
Haar	73	71	72	29.5
Geometrical Moments	69	81	75	30.3
Statistical Moments	38	88	63	18.5
Texture Moments	84	76	80	45
GLCM	**95**	**85**	**90**	**187.8**
Zernike	86	73	79.5	697.7
Hybrid	**73**	**96**	**84.5**	**19.4**

Table 4 Performance using Polynomial function (Degree 5)

Feature Extraction Method	TP	FP	%	Exec. Time
Haar	55	74	65	21.1
Geometrical Moments	90	85	87.5	26.3
Statistical Moments	81	95	84	18.6
Texture Moments	89	82	85.5	17.9
GLCM	**98**	**88**	**93**	**184.5**
Zernike	89	74	81.5	1342.0
Hybrid	**79**	**91**	**88**	**18.7**

E. Classification using Radial Basis Kernel Function

Radial Basis Kernel Function (degree 5): This gives the highest performance of 96.0 % with 197.03 seconds with the Gray Level Cooccurrence Matrix features. And the same is giving the performance level of 92.0 % with hybrid features with the minimum of 19.36 seconds of execution time. The detailed results with all the features are given in Table 5. The comparative graph is given in Fig. 15.

Table 5 Performance using Radial Basis function (Degree 5)

FE	TP	FP	%	Exec. Time
Haar	80	20	69	31.4
Geometrical Moments	81	85	83	38.3
Statistical Moments	61	96	78.5	18.8
Texture Moments	81	82	81.5	18
GLCM	**97**	**95**	**96**	**197**
Zernike	85	100	92.5	674.2
Hybrid	**88**	**96**	**92**	**19.4**

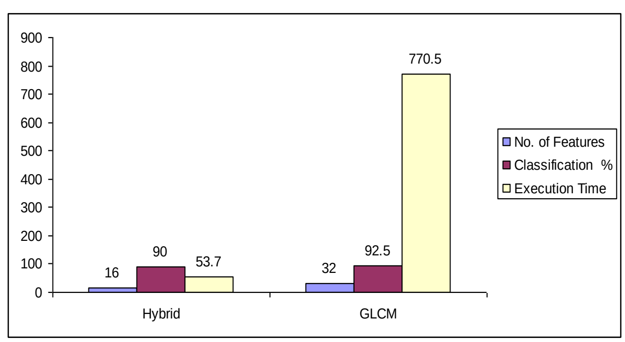


Fig. 12 Performance and time analysis graph using ANN

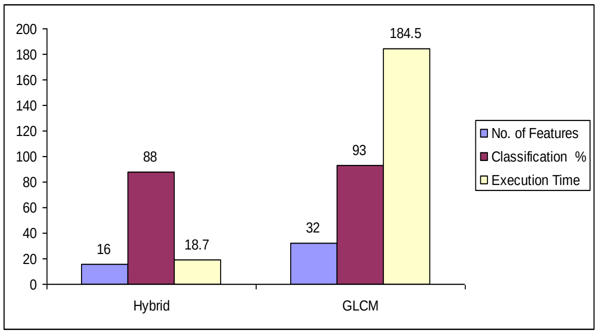


Fig. 14 Performance and time analysis graph using Polynomial Kernel (n = 5).

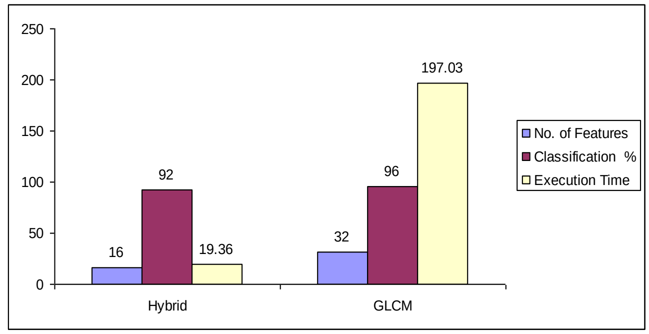
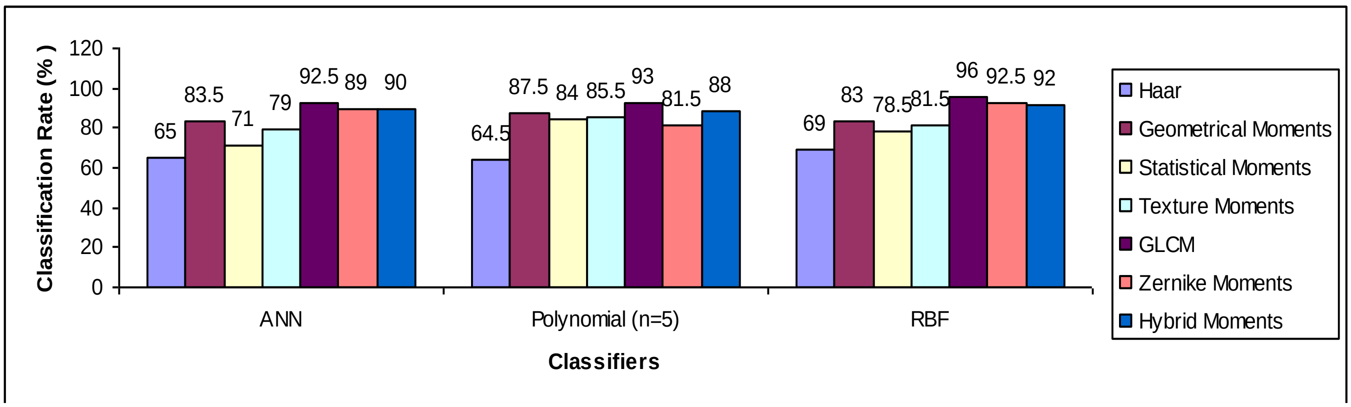


Fig. 15 Performance and time analysis graph using RBF (n= 5)



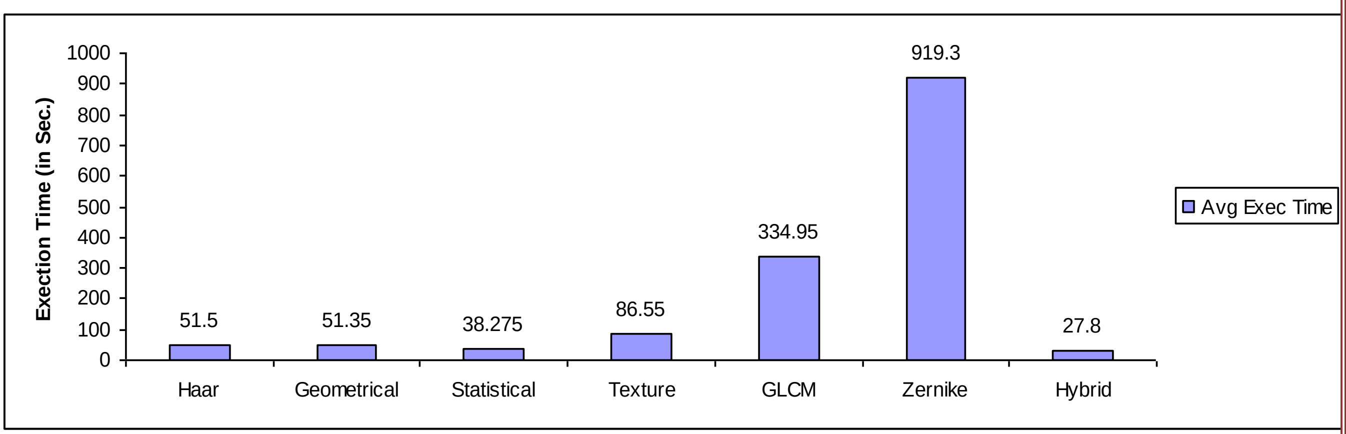


Fig 17 The Complete showcase of average execution times of feature extraction methods

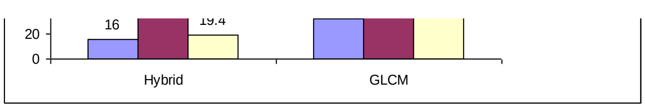


Fig. 13 Performance and time analysis graph using Polynomial Kernel (n = 2).

moments, GLCM, Zernike Moments. Artificial neural Networks, Polynomial and Radial Basis Kernel function are used to classify the scenes. Backpropagation algorithm is used for the paradigm of artificial neural networks. Polynomial kernel function (degree 2 and 5), Radial basis kernel function (degree 5) are used for support vector machines. The performances of the classifiers over the said discussed features are given in Table 2, 3, 4 and 5. This shows that co-occurrence matrix is giving the highest classification rate over all the classifiers used. But, it is consuming more execution time and suitable for real time environments. The pictorial representation of the experimental result is given in Fig. 16. Hence, co-occurrence features performances in terms of classification rate and execution time is compared with the proposed hybrid features and the same is given in Fig. 12, 13, 14 and 15. The results are proving the efficiency of the proposed hybrid feature extraction method which consumes minimum amount of execution time. Average execution times of all the feature extraction methods are given in Fig. 17. This shows that the average execution time of the proposed features is 27.8 seconds over all the classifiers used. But, average execution time of co-occurrence features, which is giving maximum classification rate over all the features, is 334.95 seconds. The proposed hybrid feature extraction method gives maximum classification rate of 92.0% when the radial basis kernel function is used for the classification in support vector machine. From our experiment, the proposed hybrid features and radial basis kernel function is derived as the better combination for solving natural scene categorization problems in real time environments.

7. Conclusion

This paper concentrates on selecting better combination of feature extraction method and the classifier which suits well for the natural scene categorization problems. The selection of feature extraction method is done as it has less feature data, gives maximum performance with less execution time. Results are proving that GLCM gives highest performance with maximum execution time over all the classifiers used. The proposed Hybrid Feature gives the comparatively better performance with less execution time over all the classifiers used. The proposed hybrid feature is compared with the all the feature extraction methods which are commonly used by our research community. The results are proving the efficiency of proposed hybrid features for natural scene categorization problems in real time environments. This complete work is implemented using SVM Toolbox in Matlab 6.5.

Reference

[1] A. Guerin-Dugue and A. Oliva, “Classification of scene photographs from local orientations features”, Pattern Recognition Letters 21 (2000) 1135-1140.
[2] A. Chella, M. Frixione and S. Gaglio, “Understanding dynamic scenes”, Artificial Intelligence 123 (2000) 89-132.
[3] Ilkka Autio and Tapio Elomaa, ”Flexible view recognition for indoor navigation based on Gabor filter and support vector machines”, Pattern Recognition 36 (2003) 2769-2779.
[4] Navid Serrano, Andreas E. Savakis and Jiebo Luo, “Improved scene classification using efficient low-level features and semantic cues”, Pattern Recognition 37 (2004) 1773-1784.
[5] Florica Mindru and et al., “Moment invariants for recognition under changing viewpoint and illumination”, Computer Vision and Image understanding 94 (2004) 3-27.
[6] Matti Pietikainen, Tomi Nurmela, Topi Maenpaa, Markus Turtinen, ”View-based recognition of real-world textures”, Pattern Recognition 37 (2004) 313-323.
[7] Matthew Boutell and Jiebo Luo, “Beyond pixels: Exploiting camera metadata for photo classification”, Pattern Recognition 38 (2005) 935-946.
[8] Andrew Payne and Sameer Singh, “Indoor vs. outdoor scene classification in digital photographs”, Pattern Recognition 38 (2005) 1533-1545.
[9] Ian Stefan Martin, “Robust Learning and Segmentation for scene Understanding”, PhD Thesis, Dept. of Electrical Engineering and Computer science, Massachusetts Institute of Technology, May 2005.
[10] Manuele Bicego, Marco Cristani and Vittorio Murino, “Unsupervised scene analysis: A hidden markov model approach”, Computer vision and image understanding 102 (2006) 22-41.
[11] A. Bosch, X. Munoz and J. Freixenet, “Segmentation and description of natural outdoor scenes”, Image and Vision computing 25 (2007) 727-740.
[12] Anna Bosch, Xavier Munoz and Robert Marti, “Which is the best way to organize/classify images by content?”, Image and vision computing 25 (2007) 778-791.
[13] Lance M. Kaplan, “Extended Fractal Analysis for Texture Classification and Segmentation”, IEEE Transactions on Image Processing, Vol. 8, No. 11, November 1999.
[14] Arivazhagan S *et al.*, “Texture Classification using Ridgelet Transform”, Proc. of Sixth Intl. Conf. on Computational Intelligence and Multimedia Applications, 2005.
[15] Arivazhagan S *et al.*, “Texture Classification using Curvelet Statistical and Co-occurrence Features”, Proc. of 18th Intl. Conf. on Pattern Recognition, 2006.
[16] G. Y. Chen and P. Bhattacharya, “Invariant Texture Classification using Ridgelet Packets”, Proc. of 18th Intl. Conf. on Pattern Recognition, 2006.
[17] Devendran V and et al., “ANN and SVM based Image classification using Wavelet Decomposition”, Asian Journal of Information Technology 6(11), 1174 – 1180, 2007.
[18] Devendran V and et al., “Scene Categorization using Invariant Moments and Neural Networks”, IEEE Computer Society Press, Vol. 1, pp. 164-168, 2007.
[19] Devendran V and et al., “Invariant Moments to Scene Categorization using Support Vector

Machines", International Journal of Soft Computing 3(2): 128-133, 2008.

[20] Devendran V *et al.*, "Feature Selection for Scene Categorization using Support Vector Machines", International Congress on Image and Signal Processing , China, 27-31 May 2008. (Accepted)

[21] B.Yegnanarayana, "Artificial Neural Networks", Prentice-Hall of India, New Delhi, 1999

[22] Vapnik, V.N., "The support vector method of function estimation. In: Generalization in Neural Network and Machine Learning. Springer-Verlag, New York, NY, pp. 239-268, 1998.

[23] Burges, C.J.C., "A tutorial on support vector machines for pattern recognition", data mining and knowledge discovery 2(2), 121-167, 1998. Available from http://svm.first.gmd.de

[24] Viola P, M Jones, "Rapid object detection using a boosted cascade of simple features", in the Proc. of Intl. Conf. on Computer Vision and Pattern Recogntion(CVPR), Volume-I, 511-518, 2001.

[25] Hu, M.K., "Visual pattern recognition by moments invariants", IRE Trans. Information Theory, 8: 179-187, 1962.

[26] Khotanzad A and Y. H. Hong, "Invariant Image Recognition by Zernike Moments", IEEE Transactions on Pattern Recognition and Machine Intelligence, Vol. 12, Issue 5, 489-497, May 1990.

[27] www-cvr.ai.uiuc.edu/ponce_grp/data

