An Investigation of Significant Object Recognition Techniques

V.N. Pawar
Assistant Professor
A.C. Patil College of Engineering, Navi Mumbai, India

Sanjay N. Talbar
Professor
SGGS Institute of Technology, Nanded, India

Summary
Our day to day activities rely largely on the precise and rapid identification of objects in our visual environment. In the current scenario, Object recognition is one of the most actively researched areas of computer vision and pattern recognition. The domain of Object recognition hopes to achieve near human levels of recognition for tens of thousands of object categories under a broad variety of conditions. The significant challenge of object recognition is the ability of the system to recognize any member of a category of objects regardless of wide variations in visual appearance due to disparities in the form and color of the object, occlusions, geometrical transformations, changes in illumination, and potentially non-rigid deformations of the object itself. In this article, we investigate the effectiveness of four vital research techniques for object recognition in digital images. The techniques investigated in the article include: Principal Component Analysis, Support Vector Machines, Hidden Markov Model and k- Nearest Neighbors classifier. The comparison is examined in terms of recognition accuracy and false positives. The Columbia Object Image Library (COIL-20) is utilized in the investigation of the techniques.

Key words:
Object Recognition, Object Model, Features, Principal Component Analysis (PCA), Support Vector Machines (SVM), Hidden Markov Model (HMM), k-Nearest Neighbors (k-NN), Columbia Object Image Library (COIL-20), Accuracy, False Positives.

1. Introduction

Obviously, nature presents immense quantity of visual information. Images and videos possess different kind of semantic content, which can serve as a significant resource when certain information is extracted for numerous application areas. Object recognition systems are a promising approach to discover such semantic content. Object recognition plays a crucial role in Computer Vision applications, specifically in the semantic description of visual content whereas it is a simple task for a human observer [1], [2]. Object recognition is the task of identifying and labeling the parts of a two-dimensional (2D) image of a scene that correspond to objects in the scene [3]. It is challenging to recognize an object from visual information. The recognition is considered chiefly invariant to the dramatic changes caused in object's appearance namely location, size, viewpoint, illumination, occlusion and considerably by the variability in viewing conditions. Some common examples of object recognition at work are: detecting a pedestrian in the view while driving, classifying an animal as a cat or a dog, recognizing a familiar face in a crowd.

Object recognition can be observed as a learning problem. To start with the system is trained on sample images of the target object class and other objects, learning to differentiate them. Subsequently, when new images are fed the system can sense the presence of the target object class [8]. Robotics and factory automation are some application domains with recognition of objects as their prime activity [4]. Object recognition has been one of the prime areas of research that has achieved tremendous progress in the past few years irrespective of its engineering applications. So many studies have been carried out on object recognition but still it remains as a hard and computational expensive problem [5]. Searching for a known object in a specified scene and locating a given object are inherently different problems [6].The recognition process necessitates prior acquisition and storage of appropriate object descriptions, or models, in a model-base. The recognition problem then becomes one of hypothesizing an object-to-model correspondence and then verifying the correctness of the hypothesis. A hypothesis is accepted if the error between the projected model features and the corresponding image features is below a specified threshold. A model includes shape, texture, and context knowledge about the occurrence of objects in a scene [3].

With the objective of simplifying object recognition and reducing computational cost, most systems (e.g. [29]) limit the recognition to specific classes of objects. In these cases, prior knowledge of classes permits one to select the most descriptive features for the objects at hand and to circumscribe the search space. Nevertheless, even under this limitation, high classification performance is seldom reached. In addition, many object recognition systems rely on user interaction to judge the correctness of returned items or to improve system response [55]. Object recognition is a computationally expensive process. Fast algorithms are essential at all stages of the recognition process including feature extraction, invariant computation, and matching objects in target images with those in the

Manuscript received May 5, 2009
Manuscript revised May 20, 2009

model-base. Object recognition is tricky because a combination of factors must be considered to identify objects. These factors may include limitations on allowable shapes, the semantics of the scene context, and the information present in the image itself [3].Objects are likely appear at different locations in the image and they can be deformed, rotated, rescaled, differently illuminated or also occluded with respect to a reference view [5].

For effective visual object recognition, a large number of views of each object are required due to viewpoint changes and it is also necessary to recognize a large number of objects, even for relatively simple tasks [7]. Constructing appropriate object models is vital to object recognition, which is a fundamental difficulty in computer vision. Desirable characteristics of a model include good representation of objects, fast and efficient learning algorithms with minimum supervised information [9]. Evidently, literature possesses so many extensive studies on object recognition (e.g. 11-20). In general, the existing approaches use image databases which illustrate the object of interest at prominent scales and with minor variations on it. The most common object recognition approaches can be classified into appearance-based [6, 11-15], model-based [14, 16-20] and approaches based on local features [21-28]. Many practical object recognition systems are appearance-based or model-based. To be successful they address two major interrelated problems: object representation and object matching. The representation should be good enough to allow for reliable and efficient matching [10].

In this paper, we have investigated four vital research techniques available in the literature for the recognition of objects in digital images. The object recognition approaches based on the pattern recognition techniques: Principal Component Analysis (PCA), Hidden Markov Model (HMM), Support Vector Machines (SVM) and *k*-Nearest Neighbors (*k*-NN) are chosen for investigation. The techniques elected for investigation are programmed in Matlab and the investigation is performed with the aid of the Columbia Object Image Library (COIL-20) dataset, which contains gray scale images of 20 objects; for each object 72 views are gathered, with a separation of 5^o. Initially, four distinct datasets are formed from the original dataset for investigation. The formed datasets are of size 6, 12, 24 and 36 respectively, each with different views of objects for training. Afterwards, the programmed techniques are trained with the formed datasets. Following the above, the remaining images respective to the training datasets are given as input to get the recognition accuracy and false Positives of the corresponding dataset. The results of the investigation are presented in the experimental results section.

The rest of the paper is organized as follows: A concise introduction about object recognition system is given in Section 2. A description of the investigated object recognition techniques is provided in Section 3. The results of the investigation are presented in Section 4. Finally, conclusions are summed up in Section 5.

2. Object Recognition System

The elemental cognitive task which is necessary for continued existence is object recognition, e.g., to detect predators or to discriminate food from non-food. In spite of the apparent effortlessness with which the visual system carries out object recognition, it is a very complicated computational task requiring a quantitative trade-off among invariance to few object transformations on one side and specificity for each objects on the other hand [52]. The problem of object recognition can be declared as follows: it is specified that a scene consisting one or more objects, and an image of the scene taken by a camera of unknown position and orientation, object recognition include solving the following two sub problems. Identification: What are the objects that are present in the scene? and Location: What is the position and orientation of each such object in relation to the camera?

A complex computational problem is resolved by the system that does object recognition. There is high inconsistency in appearance among the objects within the same class and inconsistency in viewing conditions for a particular object. The system must be capable of detecting the presence of an object for example, a face under different illuminations, scale, and views, while distinguishing it from background clutter and other classes [8]. There are two main variables in an approach that differentiate one system from another in object recognition systems. The primary variable is what features the system which uses to represent object classes. These features can be nonspecific, which can be used for any class, or class-specific. The second variable is the classifier, the module that finds out whether an object is from the target class or not, after being trained on tagged examples [8]. Commonly there will be five stages in an object recognition system: Pre-processing, Grouping, Invariant Extraction, Hypothesis Generation and Hypothesis Verification

The difficulties associated with the object recognition system are as follows: To the retina any sole object is capable of projecting infinity of image configurations. To the viewer the orientation of the object can differ endlessly, each one resulting in a dissimilar two-dimensional projection. It is possible to occlude the object with other objects or texture fields, as when we had a look behind foliage. It is not essential for the object to be offered as a

full-colored textured image; as an alternative the representation can be a basic line drawing [1]. Object recognition is also related to content-based image retrieval and multimedia indexing as a number of generic objects can be recognized. Object recognition has also been studied extensively in psychology, computational neuroscience and cognitive science [53, 54].

3. Description of Investigated Object Recognition Techniques

The object recognition approaches based on pattern recognition techniques: Principal Component Analysis (PCA), Support Vector Machines (SVMs), Hidden Markov Model (HMM) and *k*- nearest neighbor (*k*-NN) are selected for examination. SVMs have been used for a variety of learning and pattern recognition tasks such as face recognition [57], pedestrian detection [58], world-wide web searching [59] and more along with object recognition. Likewise HMM [43, 60-62], PCA [42, 63-66] and k-NN [67, 68] have been largely applied for the recognition of objects in digital images. A description of the four significant object recognition techniques, selected for investigation, is presented in this section. The details of the selected techniques are as follows:

- "Nearest Neighbor search algorithms for generic object recognition" by Ferid Bajramovic et al. [30].
- "Vision based object recognition and localization using Principal Component Analysis" by Yogesh Girdhar and Daniel Pomerantz [42].
- "Hidden Markov Model approach for appearance-based 3D object recognition" by Manuele Bicego et al. [43]
- "Object recognition using hierarchical Support Vector Machines" proposed by Katarina Mele and Jasna Maver [46]

3.1 Nearest Neighbor Search Algorithms for Generic Object Recognition

Ferid Bajramovic et al. [30] proposed a set of thinning methods and query structures for *k*-NN which are appropriate for reducing the memory requirements and/or time incurred for classification of generic object recognition.

3.1.1. Nearest Neighbor Classifier

The k nearest neighbors ($k-NN$) classifier needs a labeled training data set $\{X,Y\}=\{(x_1,y_1),....,(x_n,y_n)\}$ which consists of d dimensional feature vectors x_i and their class labels y_i. In order to classify a new feature vector x for $k=1$, it locates the closest element x_i in X and assigns the label y_i to x. The misclassification error of the 1-NN classifier converges $(for\, n\to\infty)$ to at most twice the Bayes-optimal error [31].

For $k>1$, find the k nearest neighbors $(xi_1,...,xi_k)$ of x in X. Then carry out a voting amongst the class labels $(yi_1,...,yi_k)$ of those found neighbors. The classic rule is to choose the class with the most number of votes within the set of neighbors, ties can be broken arbitrarily. In the above case i.e., $k>1$, the asymptotic $(n\to\infty)$ misclassification error of the $k-NN$ classifier is as low as the Bayes-optimal error [31]. A rejection rule can be included to enhance the voting mechanism. There are a number of possibilities: reject ties, reject if majority is too small, reject if not all neighbors are in the same class (unanimous voting). Universally when a strict voting mechanism is employed the higher the chances of rejection and lower the misclassification rate.

3.1.2. Efficient Query Structure

There are a number of approaches to advance the running time of brute force nearest neighbor search [32, 33, 34]. But there is no exact algorithm which can improve both time and space requirements in the worst case.

kd-Tree

The most relevant practically known approach for higher dimensions is the $kd-Tree$ established by Friedman, Bentley and Finkel [35]. $kd-Tree$ is a multidimensional search tree for points in k dimensional space? Levels of the tree are split along successive dimensions at the points. The basic idea of the $kd-Tree$ is to partition the space using hyper-planes orthogonal to the coordinate axes. Each leaf node possesses a bucket with a number of vectors; the other nodes in the binary $kd-Tree$ compose of a splitting dimension d and a splitting value v.

kd-Tree for Approximate Nearest Neighbor

It is not very essential to compute the nearest neighbor when we apply NN classification for generic object recognition. The only criteria set is that the classification is said to be correct if the data points found belong to the same class. So we go for the approximate nearest neighbor approach for generic object recognition developed by Arya

and Mount [36]. An $(1+\in)$ approximate nearest neighbor is defined as follows:

Definition: A vector q *is called* $(1+\in)$ *approximate nearest neighbor of* $x \in X$ *if for all* $y \in X$ *:* $d(x,y) \le (1+\in)d(y,q)$. *The value* $\in$ *is also called the error bound. If* $\in = 0$, *the query is equivalent to the exact nearest neighbor classification. Else the minimum distance to the real nearest neighbor is at least* $1/(1+\in)$ *of the calculated distance. To find a given query vector* q, *the leaf cell in the tree is located by descending the tree. Only those neighboring cells which are in the range of* $d(x,q)/(1+\in)$ *are searched for a closer training vector. Arya [36, 37] showed that the algorithm has poly-logarithmic query time and requires nearly linear space which can be made quite independent of the vector distribution.*

3.1.3 Thinning

Thinning is defined as the process of reducing the training data set $\{X,Y\}$ to a smaller subset $\{X',Y'\}$. After the thinning process, the classifier only uses $\{X',Y'\}$. The above process results in reduced memory requirements and query times. An important property of thinned data sets $\{X',Y'\}$ [31]:

Definition: A set $\{X',Y'\} \in \{K,Y\}$ *is called consistent subset of* $\{X,Y\}$ *if the 1-NN classifier for* $\{X',Y'\}$ *correctly classifies all members of the original set* $\{X,Y\}$. *This property is very desirable, as it provides assurance of perfect recognition of the 1-NN classifier for* $\{X',Y'\}$ *applied to the whole training set* $\{X,Y\}$. *The same definition can be extended with respect to the k-NN classifier:*

Definition: A vector $x \in X$ *is called k-consistent with respect to* $\{X,Y\}$ *if the unanimous k-NN classifier for* $\{X,Y\}$ *classifies it correctly. Otherwise it is called k in consistent with respect to* $\{X,Y\}$. *A set* $\{X,Y\}$ *is called k-consistent set if it has no elements which are k-inconsistent with respect to* $\{X,Y\}$. *A subset* $\{X',Y'\} \in \{X,Y\}$ *is called k-consistent subset of* $\{X,Y\}$ *if all members of* $\{X,Y\}$ *are k-consistent with respect to* $\{X',Y'\}$.

Obviously, the terms consistent subset and 1-consistent subset are equivalent. As for the 1-NN case, the property *k*-consistent subset guarantees perfect recognition of the *k*-NN classifier for $\{X',Y'\}$ applied to the whole training set $\{X,Y\}$.

Condensed Nearest Neighbor

Hart [31, 38] proposed a thinning algorithm called condensed nearest neighbor (CNN). To start with, one element of the training set is chosen arbitrarily. Subsequently, a complete scan over all remaining elements is performed. During the scan, all elements which are 1-inconsistent with respect to the new growing set are added to the new set. Additional scans are done until the new set remains unchanged for a complete scan. The thinned subset is guaranteed to be a 1-consistent subset of the training set [31]. Hart's algorithm successfully reduces the size of the data and thereby improves memory requirements and query execution times, but it also reduces the recognition rate typically [31].

Reduced Nearest Neighbor

Gates [31, 39] proposed a post-processing step for the CNN thinning algorithm. As the initial members of the thinned set are chosen arbitrarily and as additional members are added, it may be possible to remove some vectors and still retain a 1-NN consistent subset of the training set. The post-processing algorithm simply checks for each vector of the thinned set if the thinned set without that vector is still a 1-NN consistent subset of the training set. If it is, the vector is removed.

Baram's Method

Baram [31, 40] proposed a thinning algorithm that thins each class individually. For each class, a new set for the thinned class is initialized with an arbitrary member of that class. Subsequently, each vector of that class, which is 1-inconsistent with respect to a modified training set in which the current class is replaced by the growing thinned version of that class, is added. Naturally, this algorithm can also be extended to a *k*-NN version.

Proximity Graph Based Thinning

All the thinning algorithms dealt above exhibit the property that different thinned-sets will result from considering the data points in a different order. As this is undesirable, they also consider order-independent, graph-based thinning algorithms. The origin for all these order-independent algorithms is the Delaunay graph [41], which is constructed by connecting nodes in adjacent Voronoi cells. A Voronoi cell is the region of space around a point that is closer to that point than to any other point. If we remove a point from our set, all points falling in its Voronoi cell will now fall in a cell belonging to one of its neighbors in the Delauny graph. This advocates a thinning algorithm: by removing all points that are surrounded by Delauny neighbors of the same class, we are provided with a thinned set that has exactly the same classification

properties as the original set in a 1-NN classification scheme. Despite its desirable properties, Delaunay Graph thinning possesses two critical drawbacks: the algorithm is exponential in the dimensionality of the data, and empirically removes very few points for real datasets [41]. It seems that tolerating some shift in the decision boundary can (greatly) increase the number of points removed in thinning.

3.2 Vision Based Object Recognition and Localization Using Principal Component Analysis (PCA)

To find a match between the images in a way that is more robust to shift, occlusion, and rotation an algorithm is proposed by Yogesh Girdhar and Daniel Pomerantz [42].To be capable of finding a match amid unaligned images corresponding to dissimilar viewpoints of the same object or location is their major goal. An application of this matching localizing a robust based on what it sees and judge against the formerly seen images of the world. That can be done by improving upon standard PCA matching algorithms. When compared with simple PCA matching, their algorithms perform competitively in a controlled environment, but definitely well again as the occlusion as well as the shifts in the scene are greater than before.

3.2.1 Principal Component Analysis (PCA)

An image recognition system should be able to match a given image to a training image of the same object effectively is the main goal. One of the approaches to do this is to take into account of all the diverse intensity values at every single pixel as well as to run a nearest neighbor algorithm on the query image to find out which training image it almost closely matches. For a N pixel $*N$ pixel image, with a training set of size T , this requires a run time of $O(N^2T)$, which is computationally difficult.

In PCA, the $N*N$ images are projected into a dimension of size much smaller than the original image. Preferably this subspace would strike a balance between retaining as much of the original information as possible and minimizing the dimension. It means that, once if they project into this subspace, they would like to be capable of reproducing the original image as closely as possible. By calculating the eigenvectors of the covariance matrix1 of their training images this is carried out by the PCA accurately. The major components of the trained images are the eigenvectors corresponding to the largest Eigen values. Generally, they prefer at most $|T| << N2$ eigenvectors, where $|T|$ is the number of images in the training set. It is now viable that when presented with a query image, they can then run nearest neighbor approaches on the projected space. Even though this approach works reasonably well, it is very prone to small changes in the query image such as partial object occlusion, image rotation, facial expressions, translation, scaling, and a variety of angles of lighting.

3.2.2. Occlusion and Shift Invariance Using Local Matching

They would like to break both their training and test images into multiple sub-windows prior to run PCA on the entire image. Later, to each of the sub-windows PCA is applied as well as the nearest neighbor algorithms. They can locate the closest matching training sub-windows for each sub-window of the query image and join all of the sub-windows using a voting scheme.

Choosing Sub-windows
While selecting the sub windows, they could naively divide their original images along a grid of suitable size, but this approach is computationally difficult since they would end up raising the dimension too substantially. However this would be very liable to translations as they would necessarily take no notice of few parts of the image in this event. If both the query image and the training image are the same, except translated a few pixels, not anything would match. As an alternative approach they can run an interest operator on the whole image. The interest operator will spot points of interest and we will base our windows around these. Preferably the interest operator should be capable of handling changes such as rotations, translations, and scaling well as then it would be able to find out the same point in both the training phase as well the query phase.

Harris corner detector is utilized as an interest operator. Even though Harris is designed for identifying corners in an image but it suits for several other reasons: to begin with, it is translation and (planar) rotation invariant as the corner detection is insensitive to these. Secondly, the algorithm can be run fast and is comparatively simple to implement. They should choose the appropriate windows once if they run the Harris operator on their image. So as to avoid redundancy in the operator and to better space their windows, they select the top N pixels as given by the Harris operator, but with the constraint that they not intersect. Once if they select these points, they then construct a circular window around each of these points by cutting the image in the region of these points in a circular fashion with fixed radius.

Matching Images
Currently they have a set of sub-windows, each of which is "owned" by an original image. Calculate the PCA on all of these images and project to the Eigen space produced

by the training data. For every sub-window of a query image, they are the capable of calculating the distance in the Eigen space to every other sub-window. Calculate the distance to the closest training sub windows for each query sub-window. As s last step take the results of PCA for each of the sub-windows and merge this into a voting scheme.

Every original image gets a vote if it comprises a sub-window that matches one of the query sub-windows. Then votes are weighed according to the exponential. That is, if x is the query sub-window and y is the closest training sub-window, they give a vote of size w_i where w_i is

$$w_i = e^{-d(x,y)^2}$$

In addition, instead of considering simply the top match, experiment by considering the top N closest sub-windows and allowing a contributing vote. According to the distance the votes are previously weighted, this could let better accuracy as this allows for the case where the closest image has sub-windows that often appear as second or third closest matches but hardly ever as the closest.

3.2.3. Overall Algorithm

- For every training image:
  - Use an interest operator (Harris) to compute N_w sub-windows of a given size.
  - Convert the sub-windows to polar coordinates.
  - Compute their amplitude spectrum in the frequency domain by taking their Fourier transforms
- Use PCA to compute a low dimensional eigenspace using all the sub-windows of all the training images.
- Project each window onto this space to get a low dimensional vector representative of the sub-window. This forms our training database.

Then in query phase, perform the same pipeline process and match the query image to a training image using the voting scheme

3.3 Hidden Markov Model (HMM) Approach for Object Recognition

A novel method for appearance based 3D object recognition was proposed by Manuele Bicego [43], based on the Hidden Markov Model approach. The object's view is analyzed in a raster-scan fashion to attain a sequence of partially overlapped sub-images. For every sub-image, wavelet coefficients are calculated, the most significant are retained, and, finally, arranged to create a feature vector. Wavelet coefficients are extracted as local descriptors, aiming to improve robustness with respect to noise and lighting changes, while retaining the capability in grabbing essential information of the signal, by discarding insignificant parts. They are used along with HMMs, an outstanding method capable of capturing the sequential nature of data, which, in this scenario, succeeds to explain the shape of an object from an unrolled sequence of its wavelet coefficients. In this manner, a prominent framework for object classification can be constructed. The vectors sequences (one for each view) are subsequently modeled using HMMs, providing specific attention to the initialization and the model selection issues.

3.3.1. Hidden Markov Models (HMMs)

A discrete-time first-order HMM (Rabiner, 1989 [44]) is a probabilistic model that illustrates a stochastic sequence $O = O_1, O_2, ..., O_T$ as being an indirect observation of an underlying (hidden) random sequence $Q = Q_1, Q_2, ..., Q_T$, where this hidden process is Markovian, even though the observed process may not be so. More typically, a HMM is defined by the following entities [44]:

- $S = \{S_1, S_2, ..., S_N\}$ the finite set of the hidden states;
- the transition matrix $A = \{a_{ij}, 1 \le j \le N\}$ representing the probability to move from state Si to state $a_{ij} = P[Q_{t+1} = S_j | Q_t = S_i] \;\; 1 \le \mathrm{i, j} \le \mathrm{N}$ wit h $a_{ij} \ge 0 \; and \sum_{j=1}^{N} a_{ij} = 1$
- the emission matrix $B = \{b(O|S_j)\}$ indicating the probability of the emission of the symbol O when system state is S_j ; This paper employs continuous HMM: $b(O|S_j)$ is represented by a Gaussian distribution, i.e. $b(O|S_j) = N(O|\mu_j, \Sigma_j)$ where $N(O|\mu, \Sigma)$ denotes a Gaussian density of mean l and covariance R , evaluated at O ;
- $\pi = \{\pi_i\}$, the initial state probability distribution, representing probabilities of initial states, i.e. $\pi_i = P[Q_1 = S_i] \;\; 1 \le \mathrm{i} \le \mathrm{N}$ with $\pi_i \ge 0 \; \text{and} \sum_{i=1}^{N} \pi_{\mathrm{i}} = 1$

For convenience, they denote a HMM as a triplet $\lambda = (A, B, \pi)$.

3.4. Object Recognition Using Hierarchical Support Vector Machines (SVM)

To carry out object recognition, the hierarchical SVMs learning technique is offered as an alternative by Katarina Mele and Jasna Maver [46]. With the help of the hierarchical structures the crisis of cluttered background can be prevented. From one- and two-class SVMs hierarchical trees are constructed. By uniting both the recognition process is enhanced and the number of false positives is lessened simultaneously. By means of the hierarchical SVM allow we can identify t the sides of the object, e.g., front side, back side, left side, and right side. By adapting the number of levels in tree structure to the difficulty of the learning objects the method can be made better. A representation tree with more levels is necessary for the objects with complex 3D whereas symmetrical objects do not require hierarchical representation.

3.4.1. Support Vector Machines (SVM)

SVM is a binary classifier [48]. It is also possible to use it for multi class problems [47]. The SVMs are convenient for classification in high dimensional space and consequently suitable for image classification. A set of N images of size $p \times r$ can be represented as a set of points $X_i; i = 1,2,..., N$ in $R^n; n = p \times r$. Each x_i can be a member of only one of the two classes $y_i \in \{-1,1\}$, pairs $\{(X_1, y_1),...,(X_n, y_n)\}$ form a training set. The optimal separating hyper-plane (OSH) is defined with $w \in R^n$ and $b \in R$ as follows:

$$(w.x_i) + b = 0$$

The computation of w is achieved by minimizing $\|w\|$ under correct classification of the training set, i.e., $\forall i,\ y_i f(x_i) \geq 1$

This is equivalent to maximizing the margin between the training points and the separating hyper-plane, and ensures good generalization property on the real population. It can be proven [33] that w is of the form $\sum_i \alpha_i y_i x_i$ where α_i come from the following quadratic optimization problem: Minimize

$$L(\alpha) = \sum \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j x_i x_j$$

Under

$$\forall,\ \alpha_i \geq 0\ \ and \sum_i \alpha_i y_i = 0.$$

The value of b does not appear in the optimization. It has to be computed, given the α_i:

$$b = \frac{\max_{yi=-1} \sum_i a_j y_j x_i x_j + \min_{yi=1} \sum_j \alpha_j y_j x_i x_j}{2}$$

Finally, using the expansion of w, we can write the classification function as:

$$f(x) = \sum_i \alpha_i y_i x_i x + b$$

A new point x_i is classified according to the sign of $f(x)$, i.e., we test which side of the hyper-plane it belongs to.

3.4.2. Black-White Method

To the cluttered background the Black-White (BW) method attempts to be invariant. Additional training images can be produced to defeat the problem of cluttered images; the original background can be replaced with all probable backgrounds. The number of all such images is vast; for that reason Roobaert [49] recommends a data selection approach, so called pedagogical learning. They put forwards that only extreme values can be taken as the background values. In fact this means that the training set contains objects pasted onto a white and black background. With the purpose of distinctiveness between the known objects and other parts of the scene a new class with the name "non-object" is set up. As a matter of fact all images that do not correspond to the known object belong to the non-object class. Producing a representative non object image set is a complicated task and regrettably very vital for suitable classification. The outcome of an inappropriate training set of the non-object class is a huge number of fake positives. It can be known from this that the method identifies the known objects at the places in the image where there is a background.

3.4.3. One-Class SVM

It is sensible to suppose that images of an object cluster in a certain way and that images of the non-object do not for the reason that they can be more or less anything. Where there is no enough information about some classes [50, 51] one-class SVM has been effectively used in such situations. The main aim of one-class SVM is to position all the data of one-class into a hyper sphere. The formulation of the problem is the as proceeds:

Consider a set of points $x_i \in R^n; i = 1,2,\cdots,N$ belonging to the same class. Let Φ be a feature map $R^n \to F$ such that the dot product in the image of Φ can be computed by evaluating some simple kernal:

$$k(x,y) = (\Phi(x).\Phi(x))$$

We are looking for the smallest hyper-space with radius R and center c that include all

$x_i : \min_{R\in R, c\in F} R^2$ subject to

$$\|\Phi(x_i) - c\|^2 \le R^2 \quad for\ i = 1,2,\cdots,N.$$

We solve the dual problem:

$\min_{\alpha} \sum_{ij} \alpha_i \alpha_j k(x_i, x_j) - \sum_i \alpha_i k(x_i, x_i)$ subject to

$$0 \le \alpha_i, \ \sum_i \alpha_i = 1$$

The solution is

$$c = \sum_i \alpha_i . \Phi(x_i)$$

and the decision funtion:

$$f(x) = \text{sgn}(R^2 - \sum_{ij} \alpha_i \alpha_j k(x_i, x_j) + 2\sum_i \alpha_i k(x_i, x) - k(x,x))$$

R^2 is computed for the above equation such that for any x_i with $0 < \alpha_i$ the argument of sgn is 0. The function $f(x)$ is positive inside hyper-sphere and negative on the complement. To accept also the points in the near vicinity of the hyper-sphere we relax the decision function by threshold t as follows:

$$f(x) = \text{sgn}(R^2 - \sum_{ij} \alpha_i \alpha_j k(x_i, .x_j) + 2\sum_i \alpha_i k(x_i, x) - k(x,x) - t)$$

3.4.4. Hierarchical One-Class SVM

Since One class SVMs cannot exploit the idea of pedagogical learning as it is done by the BW method. Black and white backgrounds broaden the points in R^n, which direct to large hyper spheres and at the same time to poor seperability of diverse classes. In order to avoid the problem of cluttered background they systematize the image pixels in a hierarchical structure which let us to deal with the object pixels alone.

They line up the images of an object according to a rotation angle. The reason for such an arrangement is the fact that images with close viewing angles are more similar.

Let X_i denote an image from the training set and let us form a set of binary images $B_i; i = 1,\cdots,N$ such that for each pixel of B_i

$$B_i(c,l) = \begin{cases} 1 \ if \ \ X_i(c,l) \ is \ an \ obect \ pixel \\ 0 \ if \ \ X_i(c,l) \ is \ an \ background \ pixel \end{cases}$$

First, the AND operation is performed on pixels of B_i:

$$M_1(c,l) = \bigwedge_{i=1}^{N} B_i(c,l) \quad c = 1,\cdots,p, \quad l = 1,\cdots,r.$$

It is evaluated from the computed intersection that the mask of level 1 of the tree structure. The mask is a binary image with value 1 at the pixels marked as objects otherwise it is 0 in all images. After that, the image set is divided into half and the intersection mask for each half individually calculated. Consequently, level 2 is formed by two masks. In a similar manner, level 3 is formed by the intersection masks got by additional division of the image groups. From the results it is revealed that the hierarchical method has a lower false positive rate.

4. Investigation Results

The results of the investigation on the chosen research techniques are provided here. The techniques selected for investigation are programmed in Matlab (Matlab 7.4). The images from COIL-20 data set are employed in examining the effectiveness of the investigated techniques. The description of the dataset and the experimental results are given in the following subsections.

4.1. Columbia Object Image Library (COIL-20)

Columbia Object Image Library (COIL-20) is a database consisting of gray-scale images of 20 objects [56]. A black background setup was used, against which objects were placed on a motorized turntable. The turntable was rotated with respect to a fixed camera through 360 degrees to vary the object's view. Images of the objects were observed at pose intervals of 5 degrees. This corresponds to 72 images per object. The database has two sets of images. The first set possesses 720 unprocessed images of 10 objects. The second encloses 1,440 size normalized images of 20 objects. The objects in the COIL-20 dataset are presented in Figure 1. As an example, the 72 views of an object are displayed in Figure 2.

Fig. 1. The Objects in the COIL-20 Dataset

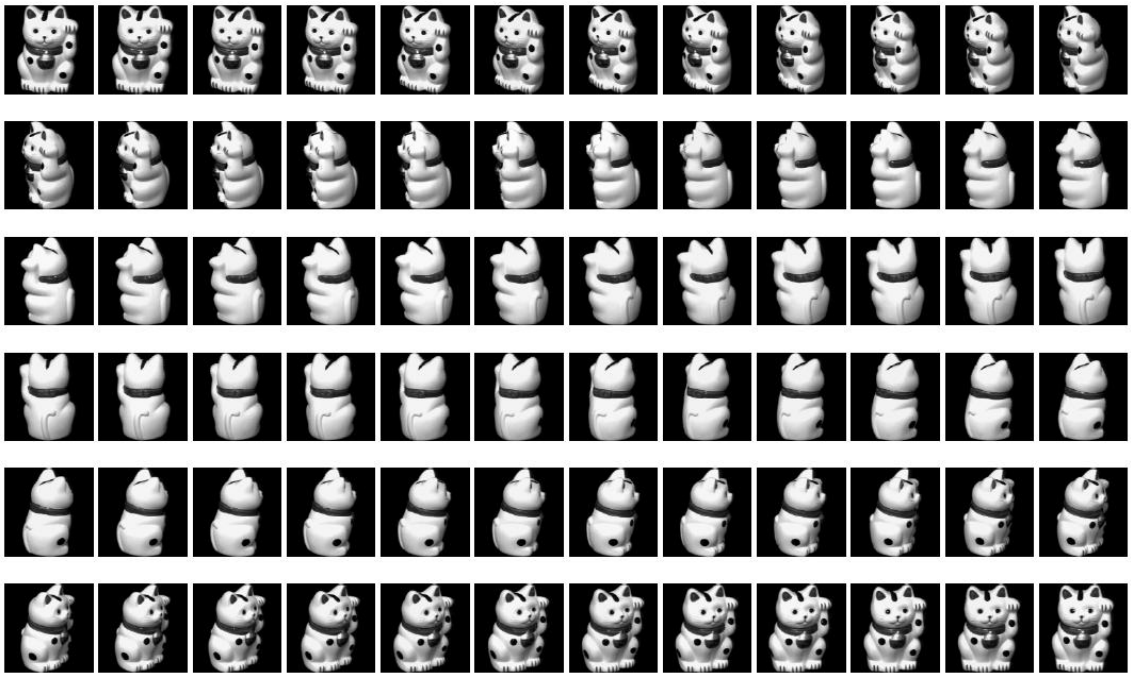

Fig. 2: The 72 views of an object in the COIL-20 Dataset

4.2 Experimental Results

The programmed techniques are examined in the following manner. The COIL-20 dataset is separated in to different disjoint sets, one employed for the training phase and the other for testing. Four different datasets are formed from the COIL-20 dataset. Each consists of 6, 12, 24 and 36 images with a separation of 60o, 30o, 15o and 10o respectively. The fact that the system is tested on a set different from the one used for training is a crucial factor for determining the capability of the system in generalizing to different views. Initially, all the programmed techniques are trained with the images in the training dataset. Afterwards, the images in the test dataset are given as input for examining the recognition accuracy and false positives of the Object recognition techniques. The results of the experiments are illustrated in Table 1.

Table 1: Results of experiments

Techniques	Training Set size	Degree (○)	Accuracy (%)	False Positives (%)
PCA	36	10	97.22	2.78
	24	15	77.08	22.92
	12	30	76.66	23.34
	6	60	42.42	57.58
k-NN	36	10	97.22	2.78
	24	15	80.12	19.88
	12	30	79.37	20.73
	6	60	48.62	51.38
HMM	36	10	100	0
	24	15	82.58	17.42
	12	30	81.47	18.53
	6	60	54.38	45.62
SVM	36	10	100	0
	24	15	83.59	16.41
	12	30	82.58	17.42
	6	60	60.72	39.28

5. Conclusion

The eventual scientific challenge of computer vision is object recognition, in a broader sense, scene understanding. To solve the problem of identifying objects the computer vision community has spent a great deal particularly in latest years. Intended for a lot of reasons object recognition is computationally hard, but the most basic is that any single object can produce an infinite set of diverse images on the retina, owing to deviation in object position, scale, pose and illumination, and due to the occurrence of visual clutter. In this paper, four vital pattern recognition approaches for object recognition in digital images are investigated. The investigated techniques are PCA, HMM, SVM and *k*-NN. The comparison has been performed in terms of recognition accuracy and false positives. The Columbia Object Image Library (COIL-20) is utilized in the investigation of the selected techniques.

References

[1] P. Duygulu, K. Barnard, N. de Freitas, and D. Forsyth, "Object Recognition as Machine Translation: Learning a lexicon for a fixed image vocabulary", In European Conference on Computer Vision (ECCV) Copenhagen, 2002.

[2] D. A. Forsyth and J. Ponce, "Computer Vision: a modern approach", Prentice Hall, 2002.

[3] P. Suetens, P. Fua, and A. J. Hanson, "Computational Strategies for Object Recognition", ACM Computing Surveys, Vol. 24, No. 1, pp. 5 – 62, March 1992.

[4] Santanu Chaudhury,S. Subramanian and Guturu Parthasarathy, "Abductive formalism for two-dimensional object recognition," Information Sciences, Vol. 68 , No. 1-2, pp. 33 - 63 ,1993.

[5] Michela Lecca, "A Self Configuring System for Object Recognition in Color Images," Proceedings of World Academy of Science, Engineering and Technology, Vol. 12, pp. 35 -40, March 2006.

[6] Philip Blackwell and David Austin, "Appearance Based Object Recognition with a Large Dataset using Decision Trees", Proceedings of the Australasian Conference on Robotics and Automation, 2004.

[7] Luke Cole, David Austin, Lance Cole, "Visual Object Recognition using Template Matching", Proceedings of the 2004 Australasian Conference on Robotics and Automation, Canberra, Australia, Nick Barnes & David Austin eds; December 6-8, 2004. ISBN: 0-9587583-6-0.

[8] J. Louie, "A biological model of object recognition with feature learning," Master's thesis, MIT, Cambridge, MA, 2003

[9] Nicolas Loeff, Himanshu Arora, Alexander Sorokin, and David Forsyth, "Efficient Unsupervised Learning for Localization and Detection in Object Categories," In NIPS 18, pages 811–818, 2006.

[10] A. Diplaros, T. Gevers and I. Patras, "Color-Shape Context for Object Recognition", IEEE Workshop on Color and Photometric Methods in Computer Vision (In conjuction with ICCV), October 2003.

[11] Rajesh Rao, "Dynamic appearance-based recognition", In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pages 540 - 546, 1997.

[12] J. Thureson and S. Carlsson, "Appearance based qualitative image description for object class recognition", In Proc.

European Conference of Computer Vision, pages 518–529, 2004

[13] Murphy-Chutorian, E. and Triesch, J., "Shared Features for Scalable Appearance-Based Object Recognition," Seventh IEEE Workshops on Application of Computer Vision, vol. 1, pp. 16 - 21, 5 - 7 Jan, 2005.

[14] Azad, P., Asfour, T. and Dillmann, R., "Combining Appearance-based and Model-based Methods for Real-Time Object Recognition and 6D Localization," In proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5339-5344,9-15 Oct, 2006.

[15] Martin Winter, "Spatial Relations of Features and Descriptors for Appearance Based Object Recognition", PhD thesis, Graz University of Technology, Faculty of Computer Science, 2007.

[16] Lamdan, Y.; Schwartz, J.T.; Wolfson, H.J., "Affine invariant model-based object recognition", IEEE Transactions on Robotics and Automation, Volume 6, Issue 5,Pp:578 - 589, October 1990

[17] Farshid Arman and J. K. Aggarwal, "Model-based object recognition in dense-range images—a review," ACM Computing Surveys, Vol. 25, No. 1, pp. 5 - 43, 1993.

[18] A.S. Mian, M. Bennamoun, and R.A. Owens, "A Novel Algorithm for Automatic 3D Model-Based Free-Form Object Recognition," Proc. IEEE Int'l Conf. Systems, Man, and Cybernetics, vol. 7, pp. 6348-6353, 2004.

[19] Ajmal S. Mian, Mohammed Bennamoun and Robyn Owens, "Three-Dimensional Model-Based Object Recognition and Segmentation in Cluttered Scenes," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 28, No. 10, pp. 1584 - 1601, 2006.

[20] P.A. Larsena,J.B. Rawlingsa and N.J. Ferrier, "Model-based object recognition to measure crystal size and shape distributions from in situ video images," Chemical Engineering Science, Vol. 62, No. 5, pp. 1430-1441,March 2007.

[21] William M. Wells, "Statistical Approaches to Feature-Based Object Recognition", International Journal of Computer Vision, v.21 n.1-2, p.63-98, January. 1997.

[22] G. Häusler and D. Ritter, "Feature-based object recognition and localization in 3D-space, using a single video image," Computer Vision and Image Understanding, Vol. 73 , No. 1, pp. 64 - 81, 1999.

[23] Lowe, D.G, "Object Recognition from Local Scale-Invariant Features", The Proceedings of the Seventh IEEE International Conference on Computer Vision, Vol. 2, pp. 1150-1157, 1999.

[24] Gotze, N.; Drue, S.; Hartmann, G., "Invariant object recognition with discriminant features based on local fast-Fourier Mellin transform", Proceedings. 15th International Conference on Pattern Recognition, Vol. 1, pp. 948 - 951, 2000.

[25] Zhengrong Ying and Castanon, D., "Feature based object recognition using statistical occlusion models with one-to-one correspondence," In Proceedings of Eighth IEEE International Conference on Computer Vision, Vol. 1, pp. 621 - 627, 2001.

[26] Andrew Stein and Martial Hebert, "Incorporating Background Invariance into Feature-Based Object Recognition," In Proceedings of the Seventh IEEE Workshops on Application of Computer Vision, Vol. 1, pp. 37-44, 2005.

[27] Thomas Serre , Lior Wolf , Tomaso Poggio, "Object Recognition with Features Inspired by Visual Cortex", Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2, p.994-1000, June 20-26, 2005 .

[28] Sungho Kim, Kuk-Jin Yoon and In So Kweon, "Object recognition using a generalized robust invariant feature and Gestalt's law of proximity and similarity, "Pattern Recognition, Vol. 41, No.2, pp. 726-741, 2008

[29] A. Hoogs, R. Collins, R. Kaucic, and J. Mundy, "A common set of perceptual observables for grouping, figure - ground discrimination, and texture classification", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 4, pp. 458–474, 2003.

[30] Bajramovic Ferid, Mattern Frank, Butko Nicholas, Denzler Joachim, "A Comparison of Nearest Neighbor Search Algorithms for Generic Object Recognition", Lecture notes in computer science, source: International Conference on Advanced Concepts for Intelligent Vision Systems No8, Antwerp , Belgique (2006), vol. 4179, pp. 1186-1197, 2006.

[31] Toussaint, G., "Geometric proximity graphs for improving nearest neighbor methods in instance-based learning and data mining', Int. J. of Comp. Geom. & Appl., Vol. 15, pp. 101–150, 2005.

[32] Clarkson, K., "A randomized algorithm for closest-point queries", SIAM Journal of Computing, Vol. 17, pp. 830–847, 1988.

[33] Dobkin, D., Lipton, R., "Multidimensional searching problems", SIAM Journal of Computing, Vol. 2, pp. 181–186, 1976

[34] Meisner, S., "Point location in arrangements of hyper planes", Information and Computation, Vol. 2, pp. 286–303, 1993.

[35] Friedman, J., Bentley, J., Finkel, R., "An algorithm for finding best matches in logarithmic expected time", ACM Transactions on Mathematical Software, Vol. 3, pp. 209–226, 1977.

[36] Arya, S., Mount, D., "Approximate nearest neighbor queries in fixed dimensions", In: Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 271–280, 1993.

[37] Arya, S., Mount, D., Netanyahu, N., Silverman, R., Wu, A., "An optimal algorithm for approximate nearest neighbor searching", Journal of the ACM, Vol. 45, pp. 891–923, 1998.

[38] Hart, P.E., "The condensed nearest neighbor rule", IEEE Transactions on Information Theory, Vol. 14, pp. 515–516, 1968.

[39] Gates, W., "The reduced nearest neighbor rule", IEEE Transactions on Information Theory, Vol. 18, pp. 431–433, 1972.

[40] Baram, Y., "A geometric approach to consistent classification", Pattern Recognition, Vol. 13, pp. 177–184, 2000.

[41] Toussaint, G., Bhattacharya, B., Poulsen, R., "The application of voronoi diagrams to nonparametric decision rules", In: 16th Symp. On Comp. Science and Statistics, pp. 97–108, 1984.

[42] Yogesh Girdhar and Daniel Pomerantz, "Vision Based Object Recognition and Localization", December 16, 2007

from
http://www.cim.mcgill.ca/~dpomeran/vision/project.pdf
[43] Manuele Bicego, Umberto Castellani, Vittorio Murino, "A Hidden Markov Model approach for appearance-based 3D object recognition", Pattern Recognition Letters, Vol. 26, pp. 2588–2599, 2005.
[44] Rabiner, L., "A tutorial on Hidden Markov Models and selected applications in speech recognition", Proc. IEEE, Vol. 77, No. 2, pp. 257–286, 1989.
[45] De Vore, R., Jawerth, B., Lucier, B., "Image compression through wavelet transform coding", IEEE Trans. Inform. Theory, Vol. 38, No. 2, pp. 719–746, 1992.
[46] Katarina Mele and Jasna Maver, "Object Recognition Using Hierarchical SVMs", In: Computer Vision Winter Workshop '03, Valtice, Czech Republic, 3-6 February 2003.
[47] N. Cristianini and J. Shawe-Taylor, "An Introduction to Support Vector Machines (and other kernel-Based Learning Methods)", CUP, 2000.
[48] M. Pontil and A. Verri, "Support vector machines for 3d object recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 6, pp. 637–646, 1998.
[49] D. Roobaert, "Pedagogical Support Vector Learning: a pure learning approach to object recognition", PhD thesis, Royal Institute of Technology (KTH), Dept. of Numerical Analysis and Computing Science (NADA), Stockholm, Sweden, 2001.
[50] Y. Chen, X. Zhou, and T. Huang, "One-class SVM for learning in image retrieval", In Proceedings of the 2001 IEEE International Conference On Image Processing (ICIP-01), pages 34–37, Thessaloniki, Greece, Oct. 7–10 2001.
[51] B. Sch ̈olkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution", Neural Computation, Vol. 13, No. 7, pp. 1443–1471, 2001.
[52] Riesenhuber, M. Object Recognition in Cortex: Neural Mechanisms and Possible Roles for Attention. In: Neurobiology of Attention, (Eds. L. Itti, G. Rees, and J. Tsotsos), Elsevier, 279-287, 2005.
[53] S. Edelman, "Representation and recognition in vision", MIT Press, 1999.
[54] D. Marr, "Vision", W. H. Freeman and Company, 1982.
[55] B. Ko and H. Byun, "Extracting Salient Regions And Learning Importance Scores In Region-Based Image Retrieval", International Journal of Patter Recognition and Artificial Intelligence, Vol. 17, No. 8, pp. 1349–1367, 2003.
[56] S. A. Nene, S. K. Nayar and H. Murase, "Columbia Object Image Library (COIL-20),"Technical Report CUCS-005-96, February 1996.
[57] B. Heisele, P. Ho, and T. Poggio, "Face recognition with support vector machines: Global versus component-based approach", In Proceedings of the Eighth International Conference On Computer Vision (ICCV-01), pages 688–694, Los Alamitos, CA, July 9–12 2001.
[58] C. Papageorgiou and T. Poggio, "Trainable pedestrian detection", In Proceedings of the 1999 International Conference on Image Processing (ICIP-99), pages 35–39, Los Alamitos, CA, Oct. 24–28 1999.
[59] I. Kkai and A. Lrincz, "Fast adapting value estimation-based hybrid architecture for searching the world-wide web", Applied Soft Computing, 2(1):11–23, 2002.
[60] C. de Trazegnies, C. Urdiales, A. Bandera and F. Sandoval, "A Hidden Markov Model object recognition technique for incomplete and distorted corner sequences," Image and Vision Computing, Vol. 21, No. 10, pp. 879-889 ,September 2003,
[61] Young K. Ham,Kil M. Lee and Rae-Hong Park, "Three-dimensional object recognition using hidden Markov models," In Proceedings of Visual Communications and Image Processing, Vol. 2501, pp.148-157, 1995.
[62] Young Kug Ham and Park R.-H.,"3D object recognition in range images using hidden markov models and neural networks," Pattern recognition, vol. 32, no. 5, pp. 729-742, 1999.
[63] Jeff Fortuna, Derek Schuurman, David Capson, "A Comparison of PCA and ICA for Object Recognition under Varying Illumination," icpr, vol. 3, pp.30011, 16th International Conference on Pattern Recognition (ICPR'02) - Volume 3, 2002
[64] M. Asunción Vicente1, Cesar Fernández, Oscar Reinoso1 and Luis Payá1, "3D Object Recognition from Appearance: PCA Versus ICA Approaches," Lecture Notes in Computer Science, Vol. 3211/2004, Pages 547-555, 2004.
[65] Deepti, P. and Das, S., "Generic Object Recognition Using 2-D PCA and Virtual Manifolds," IET International Conference on Visual Information Engineering, pp. 18 - 23,26-28 Sept. 2006.
[66] Jong-Min Kim and Hwan-Seok Yang, "A Study on Object Recognition Technology Using PCA in the Variable Illumination," Lecture Notes in Computer Science, Vol. 4093/2006, pp. 911-918, 2006.
[67] Jong-Min Kim, Jin-Kyoung Heo, Hwan-Seok Yang, Mang-Kyu Song, Seung-Kyu Park and Woong-Ki Lee, "Object Recognition Using K-Nearest Neighbor in Object Space," Lecture Notes in Computer Science, Vol. 4088/2006, pp. 781-786, 2006.
[68] MyungA Kang and JongMin Kim," Real Time Object Recognition Using K-Nearest Neighbor in Parametric Eigenspace," Lecture Notes in Computer Science, Vol. 4688/2007, pp. 403-411, 2007.

V.N. Pawar is a PhD student at SGGS Institute of Engineering & Technology, Vishnupuri, Nanded, India doing research in the field of object recognition systems. He received B.E. electronics and M.E. (Control Systems) from Shivaji University, Kolhapur, India in 1990, and 1997 respectively. He is presently working as Asst. Professor in Electronics Department, A.C. Patil College of Engineering, Navi Mumbai. He is life member of IETE and ISTE. His research interests are in Embedded Systems, Image Processing and Robotics.

Sanjay N. Talbar received his B.E and M.E degrees from SGGS Institute of Technology, Nanded, India in 1985 and 1990 respectively. He obtained his PhD (Highest degree) from SRTM University, India in 2000. He received the Young Scientist Award by URSI, Italy in 2003. He has Collaborative research programme at Cardiff University Wales, UK. Currently he is working as Professor in Electronics & Telecommunication Department of SGGS Institute of Technology Nanded, India. He is a member of many prestigious committees in academic field of India. His research interests are Signal & Image processing and Embedded Systems.