A Supervised Learning Framework for Generic Object Detection in Images

Saad Ali
Computer Vision Lab
University of Central Florida
Orlando, FL, U.S.A
sali@cs.ucf.edu

Mubarak Shah
Computer Vision Lab
University of Central Florida
Orlando, FL, U.S.A
shah@cs.ucf.edu

Abstract

In recent years Kernel Principal Component Analysis (Kernel PCA) has gained much attention because of its ability to capture nonlinear image features, which are particularly important for encoding image structure. Boosting has been established as a powerful learning algorithm that can be used for feature selection. In this paper we present a novel framework for object class detection that combines the feature reduction and feature selection abilities of Kernel PCA and AdaBoost respectively. The classifier obtained in this way is able to handle change in object appearance, illumination conditions, and surrounding clutter. A nonlinear subspace is learned for positive and negative object classes using Kernel PCA. Features are derived by projecting example images onto the learned subspaces. Base learners are modeled using Bayes classifier. AdaBoost is then employed to discover the features that are most relevant for the object detection task at hand. The proposed method has been successfully tested on wide range of object classes (cars, airplanes, pedestrians, motorcycles, etc) using standard data sets and has shown remarkable performance. Using a small training set, a classifier learned in this way was able to generalize the intra-class variation while still maintaining high detection rate. In most object categories we achieved detection rates of above 95% with minimal false alarm rates. We demonstrate the effectiveness of our approach in terms of absolute performance parameters and comparative performance against current state of the art approaches.

1. Introduction

Detection and classification of the object of interest in an unconstrained environment is a challenging problem. Objects can occur under different visual appearances, poses, lighting conditions, backgrounds and clutter (Fig. 1). In addition to dealing with these intra-class variations, a successful object detector needs to tackle diverse imagery that exists in different applications. Automated object detection has a wide range of applications such as surveillance, military target recognition, content based image retrieval, robotics, image mining, etc. Hence, there is a pressing need for a methodology which can carry out automatic object detection and indexing across wide range of imagery.

Traditional methods for visual classification involve two steps. First, features are extracted from the image and the object of interest is represented using those features. In the second step a classifier is learned using the chosen feature representation. Popular classifiers employed for this task include Support Vector Machines, Perceptron, Winnow, Bayes Classifier, Fisher Linear Discriminant, etc. These are termed as hyperplane classifiers, which work under the assumption that all features of the data are useful for classification and that the data is linearly separable (or by linear combination of hyper-planes). Unfortunately, the images of objects such as cars, persons, airplanes, faces, etc, taken under different photometric and geometric conditions results in a highly nonlinear and non-convex feature space. Imaging process and low level image features such as gray levels, color or texture, that are derived from images acquired through this process, are nonlinear functions of various factors. Therefore a simple linear separation of class and non-class images in feature space is not optimal [17, 12]. However, most of the current approaches use color, texture, orientation or blob features and try to learn a linear classifier using them. Others try to compute similarity measures (L_1 or L_2 norm) between these high dimensional features to return the relevant object. But in high dimensions, data becomes very sparse and distance measures

Figure 1: Examples of variation among object categories (Airplane and Cars) in terms of appearance, illumination condition, and background.

become increasingly meaningless. Therefore it is common to observe a degradation in the quality of results returned by such systems.

Principal Component Analysis (PCA) is an orthogonal basis transformation that can be effectively performed on a set of observations that vary linearly. However, it fails to detect structure in given data if the variations among the observations are nonlinear, which is the case when one is trying to extract features from object categories that vary in their appearance, pose and illumination conditions. Therefore any subsequent learning algorithm will have poor classification performance under these conditions.

To overcome the above mentioned shortcomings we propose an integrated framework of Kernel PCA [1] and AdaBoost. We demonstrate the feasibility of our approach on the task of object detection on a wide range of object categories. The essential idea is to employ Kernel PCA as nonlinear feature extractor by mapping input space to a higher dimensional feature space, through a non-linear map, where the data is linearly separable. Cover's theorem [3] provides the justification of converting data to higher dimensional space. This theorem formalizes the intuition that the number of separations increases with the dimensionality as we can have more views of the class and non-class data. Note that in practice we do not have to compute the expensive higher dimensional mapping as we can achieve the same effect by using the kernel trick [2]. This mapping will solve the problem of nonlinear distribution of low level image features. It will also act as a dimensionality reduction step. Once in the feature space, which is of high dimension, we uncover the patterns by selecting only the relevant (discriminative) dimensions using AdaBoost. As a result the final classifier will only have to compute highly discriminative features which speeds up the classification process. In addition a classifier in our framework can be learned using a small set of examples which is an added advantage.

2. Related Work

During last decade object detection and recognition has been an active area of research. Recently, part based object recognition [4, 13, 5, 6] and affine invariant features [7, 8, 9] have shown promising results. Part based approaches encode the object structure by using a set of patches covering important parts of the object. Patches themselves are detected using interest point operators e.g., Harris, SIFT etc. Recently, [14] used four different features to encode the extracted patches. These features include intensity, intensity moments, moment invariants and SIFT. A boosting framework is then used to select the best features which are used for classification. In contrast, our learning method uses a global approach for capturing the object structure. In affine invariant approaches for object recognition small patches are extracted from the image which are characterized by view point invariant descriptors. These descriptors are used to match the object.

PCA is a powerful technique for extracting global structure from a high dimensional data set. It has been used to extract features for face recognition such as in the well known Eigenfaces method [15], where the eigenfaces correspond to the eigenvectors associated with the largest eigenvalues of the face covariance matrix. Kernel PCA was proposed as a nonlinear extension of PCA in the pioneering work of [1], which computes the principal components in a high dimensional feature space which is nonlinearly related to the input space. Therefore it is able to extract nonlinear principal components. Yand [10] and Moghaddam [11] compared the face recognition performance using Kernel PCA and Eigenfaces method by using Kernel PCA with the cubic polynomial kernel and Gaussian kernel respectively. Their results showed that Kernel PCA achieved much lower error rates.

In addition, Kernel PCA is also used to model the variability in classes of 3D-shapes [18, 16]. In [19] Kernel PCA is used in conjunction with SVM to learn the view subspaces for multi-view face detection and recognition. Each view is treated separately by first using Kernel PCA to extract the nonlinear features and then training a SVM for that view. During testing an image is provided to each SVM for labeling. Recently [20] employed it for recognition of facial expression using Gabor filters. Features derived by Gabor filters were nonlinearly projected onto higher dimensional feature space by employing fractional power polynomial as a kernel function.

Our framework enables us to exploit strengths of both Kernel PCA and boosting, first by modelling the nonlinear subspaces of object categories using Kernel PCA, and second by selecting highly discriminative features using boosting. In addition, our framework is able to handle multiple categories as opposed to above mentioned approaches that are restricted to just one category. We illustrate the robust performance of this approach on standard data sets.

3. Kernel PCA for Feature Extraction

3.1 Kernel PCA

Given a set of examples $x_i \in \Re^N$, *i*=1,...m, which are centered, $\sum_{i=1}^{m} x_i$ = 0, PCA finds the principal axis by diagonalizing the covariance matrix:

$$C = \frac{1}{m} \sum_{j=1}^{m} x_j {x_j}^\top. \tag{1}$$

Eigenvalue equation, $\lambda\nu = C\nu$ is solved where ν is eigenvector matrix. First few eigenvectors are used as the

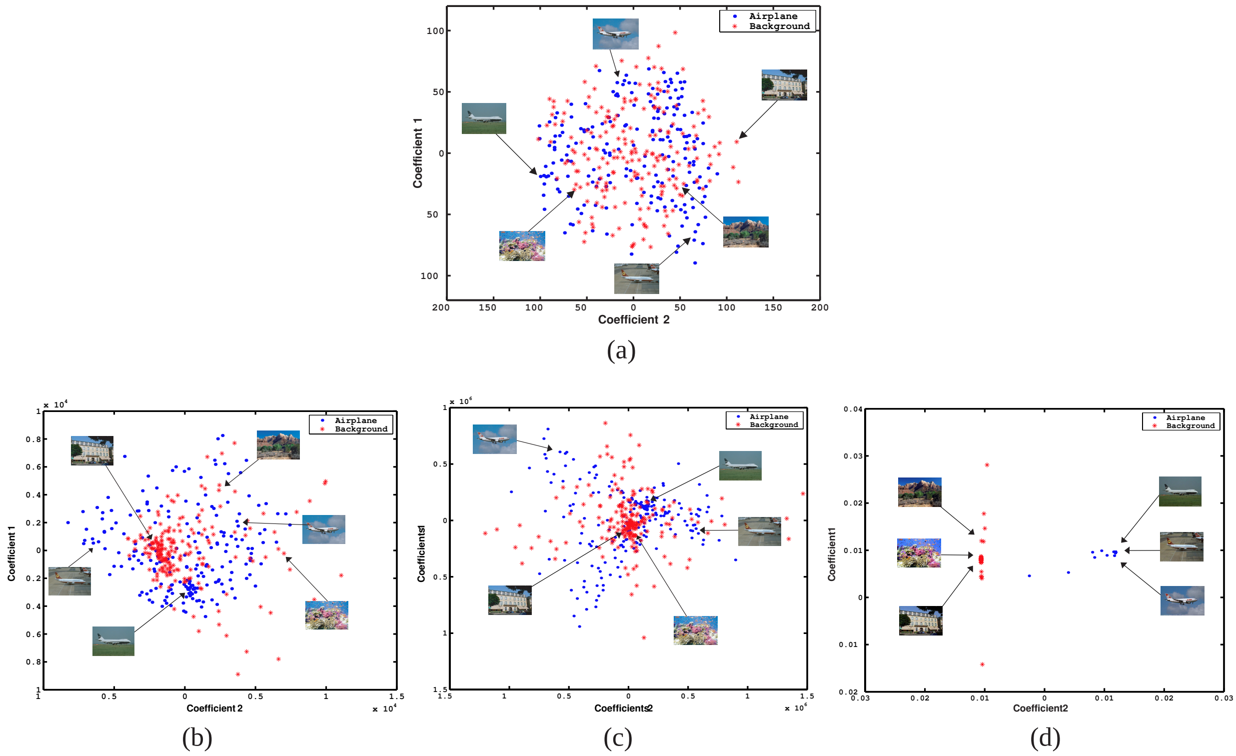

Figure 2: The above sequence of figures shows the ability of Kernel PCA to capture the nonlinear structure of the data under consideration. The first two PCA and Kernel PCA coefficients obtained from the gradient images of Caltech Airplane and Caltech Background data sets are plotted against each other. Blue dots and red asterisks represent the coefficients for Airplane and Background respectively. (a) The figure shows the first two PCA coefficients calculated from the Airplane and Background data set. It validates the argument that due to the overlap and non-convexity of these classes, a linear classifier will not ensure a robust separability. (b) In this figure we plotted the first two Kernel PCA coefficients obtained using a polynomial kernel of degree two. Clusters are much more coherent and thus signifying the ability of the Kernel PCA to deal with nonlinear features present in the images. (c) It shows a plot of first two Kernel PCA coefficints obtained using polynomial kernel of degree three. Clusters are much more prominent. (d) Plot of first two Kernel PCA coefficients obtained by using the Gaussian kernel clusters are well separated.

basis vectors of the lower dimensional subspace. Eigen features are then derived by projecting the examples onto these basis vectors.

Kernel PCA is performed by first mapping the data from input space to a higher dimensional feature space i.e. using a map $\phi : \Re^N \rightarrow$ F, and then performing a linear PCA in F. The covariance matrix in this new space F is,

$$\overline{C} = \frac{1}{m}\sum_{j=1}^{m}\phi(x_j)\phi(x_j)^\top. \tag{2}$$

The eigenvalue problem now becomes λV = $\overline{C}$V. As mentioned previously we do not have to explicitly compute the nonlinear map ϕ. We can achieve the same goal by using the kernel function *k*(x_i,x_j) $=$ $(\phi(x_i).\phi(x_j))$, which implicitly computes the dot product of vectors x_i and x_j in the higher dimensional space [2]. Kernel functions can also be thought of as functions measuring similarity between instances. The kernel value will be greater if two samples are similar, otherwise it falls off to zero if samples are distant. The most often used kernel types are polynomial and Gaussian kernels (Table 1). Pairwise similarity between input

Gaussian Kernel	$k(x_i, x_j) = exp(\frac{-\|x_i-x_j\|^2}{c}$
Polynomial Kernel	$k(x_i, x_j) = (x_i.x_j + a)^d$, d=1,2..
Sigmoid Kernel	$tanh(k(x_i.x_j) + a)$

Table 1: Kernel Functions

examples are captured in a matrix K which is also called Gram matrix. Each entry $K_{i,j}$ of this matrix is calculated using the kernel function $k(x_i, x_j)$. Eigenvalue equation in terms of Gram matrix is written as (see [2]),

$$mA\Lambda = KA, \tag{3}$$

with A= $(\alpha_1, ..., \alpha_M)$ and Λ = *diag*$(\lambda_1, ..., \lambda_M)$. A is a *m* x *m* orthogonal eigenvector matrix and Λ is a diagonal eigenvalue matrix with diagonal elements in decreasing order. Since the eigenvalue equation is solved for α's instead of eigenvectors v_i of Kernel PCA, we will have to normalize A to ensure that eigenvalues of Kernel PCA have unit norm in the feature space, therefore $\alpha_j = \alpha_j / \sqrt{\lambda_j}$. After normalization the eigenvector matrix V of Kernel PCA is computed as, V = DA where D =[$\phi(x_1)\ \phi(x_2)\ ...\ \phi(x_m)$] is the data matrix in feature space. Now let **x** be a test example whose map in the higher dimensional feature space is $\phi(\mathbf{x})$. The Kernel PCA features for this example are derived as follows:

$$F = V^\top \phi(\mathbf{x}) = A^\top B, \tag{4}$$

where $B = [\phi(x_1)\phi(\mathbf{x})\ \phi(x_2)\phi(\mathbf{x})\ ...\ \phi(x_m)\phi(\mathbf{x})]$.

3.2 Feature Extraction

Let $(p_1, p_2, ...p_m)$ and $(n_1, n_2, ...n_m)$ be the positive and negative images of the training set provided for learning. Gradient magnitudes are extracted from the images by convolving them with sobel gradient operator. Gradient provides better shape cues than gray level intensity or color texture patterns, which are more biased towards the visual appearance of the object and background clutter. Gradients are more stable to illumination changes as well. Gradient images are resized to 128 by 128 pixels, converted into column vector form and made zero mean and unit variance. We computed Gram matrices K_p and K_n for positive and negative examples respectively. Eigenvector matrices A_p and A_n are calculated using eq. 3. Features for our base learners were obtained by projecting each positive and negative training example onto the positive and negative higher dimensional subspaces by plugging A_p and A_n in eq. 4, respectively. The feature vector for any particular example will be of the form, $\mathbf{f} = [d_1, d_2, ..., d_{w_1}, d_{d_1+1}, ..., d_{w_1+w_2}]$ where w_1 and w_2 are the number of principal components that we retained for each class. Accordingly the total number of base learners will be $w_1 + w_2$.

4. Learning Classifier with Boosting

The notion of focusing on the most relevant information in potentially high dimensional data is very important. Efficiency of the final system depends on whether we are able to discover the irrelevant features that hide the useful information in a sea of noise or not. Features generated by Kernel PCA lie in a high dimensional nonlinear subspace and we want to find out if all of those dimensions are useful for the classification task at hand, or can we achieve the same goal by using a subset of those dimensions. Therefore we employ AdaBoost for this purpose. Adaboost provides a powerful stage wise learning approach for classification and feature selection. AdaBoost is an ensemble classifier learning algorithm that works by creating a sequence of base learners in iterative fashion, where each base learner is selected based on its performance on the training set. In each iteration the weight distribution over the training set is updated in a way that forces the base learners to focus on the example that are hard to classify. This results in a classifier with low training error and good generalization performance.

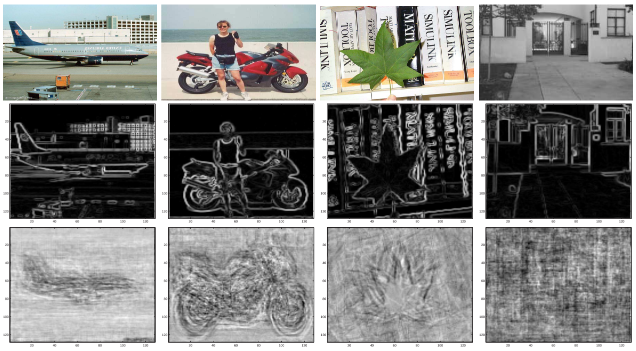

Figure 3: In top to bottom form, each image is followed by its gradient and reconstructed image structure from top 150 eigenvectors. Categories are Airplane, Motorcycle, Leafs and Background. Note how well the top eigenvectors cover the main structure while remaining components pick up the noise. No consistent structure present in the background is reflected in its reconstruction.

Note that one may be tempted to use nearest neighbor (NN) classifier or any similar classifier to categorize the features derived from Kernel PCA without carrying out any feature selection. This will have adverse effect on the classification performance as NN uses all features for its distance computation which will include some features generated from the noisy data. In addition, the number of training examples required to reach a given accuracy grows exponentially with the number of irrelevant features in case of NN. On the other hand our framework guarantees to provide a classifier based on the subset of most discriminative features using small set of training examples as it uses Adaboost to select the best features.

We preferred to incorporate Adaboost in our framework, instead of SVM, as boosting has added advantages of low error rate and computational efficiency. The solution of SVM is expressed as a linear combination of training examples using coefficients. By maximizing the smallest margin, SVM gives a sparse solution in the example space where most of the coefficients become zero. Examples having non-zero coefficients are called support vectors, which form the final solution. Boosting, in contrast, performs computation explicitly in the feature space. As we know, the underlying concept behind boosting is that only few hypotheses/base learners are needed to express the final solution. Boosting thus finds a sparse solution in the feature space by selecting only the relevant features. Although accura-

Input: Positive and negative training sample $P_1, P_2, ..., P_u$ and $N_1, N_2, ..., N_v$, where each example is a *d* dimensional vector obtained by vectorizing positive and negative class images.

Output: Classifier *H(x)*

1. Compute the kernel matrices K_p and K_n from positive and negative training samples. Their dimensions will be *u*×*u* and *v*×*v* respectively. Each entry of the matrix is obtained by evaluating one of the kernel functions mentioned in Table 1.
2. Solve eigenvalue equations:
$$u\Lambda A_p = K_p A_p, \quad (5)$$
$$v\Lambda A_n = K_n A_n, \quad (6)$$
where Λ and A_n are eigenvalue and eigenvector matrices, respectively.
3. Obtain Kernel PCA based feature vectors by computing principal component projections of each training sample onto the nonlinear subspaces of positive and negative samples using A_p and A_n respectively. If f_p and f_n are the feature vectors obtained by this projection, then the augmented feature vector will be $f = [f_p \;\; f_n]$.
4. Construct a one dimensional histogram h_i for *ith* dimension of feature vector **f**. These histograms will be the weak classifiers.
5. Train AdaBoost using algorithm proposed in [21]. For this step training samples will be the labeled feature vectors obtained in Step 3. Training will return a classifier which will be the weighted combination of weak classifiers h_i.
6. Output the strong classifier *H(x)*

Figure 6: The steps involved in our algorithm.

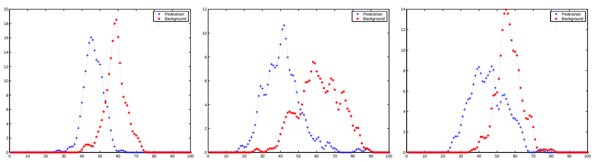

Figure 4: Histograms of three different feature dimensions used in training of Pedestrian-Background Classification. Blue and Red represent pedestrian and background respectively.

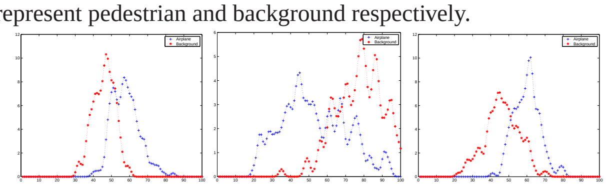

Figure 5: Histograms of three different feature dimensions used in training of Airplane-Background Classification. Blue and Red represent airplane and background respectively.

cies of these two methods are guided by the training data, SVM has one major drawback. At run time it needs to compute all the features. Computing all features may be feasible during the training stages but doing it at run time will be too costly, especially for the object detection task, where we need to search over thousands of possible image locations and scales. On other hand, boosting has a number of additional practical advantages. First, sparse feature selection allows for the construction of an efficient classification algorithm. It performs faster since the complexity depends only on a small number of base learners, which are bounded by the number of iterations in the training phase. Second, sparse features are useful in practice as they provide a clear understanding of which features are useful.

We used the Bayes classifier as the base learner for AdaBoost. Let c_p and c_n be the positive and negative class respectively. The classification decision of *ith* classifier is taken as c_p if $P(c_p|d_i) > P(c_n|d_i)$. The class conditional probability densities $p(d_i|c_p)$ and $p(d_i|c_n)$ are approximated through smoothed one dimensional histogram of the i^{th} dimension of the feature vector **f**. In order to have good discrimination, ranges and bin widths of these histograms needed to be selected carefully. Example histograms of three different features for pedestrian/background and airplane/background feature vectors are given in Fig.4 and Fig. 5 respectively.

Boosting algorithm proposed by [21] was used for feature selection. In order to test a new image, we preprocess it according to the specifications described in 3.2. The feature vector is obtained by projecting it onto the nonlinear manifolds using A_p and A_n (eq. 4). Note that in eq. 4, $(x_1, x_1 ... x_m)$ are the same training examples that were used to construct the nonlinear manifolds. We need to save them as they will be used for testing any new example. The steps of our algorithm are summarized in Fig. 6.

5. Results and Discussion

This section assesses the performance of our object detection approach using the standard data sets available in the public domain. We performed the classification in the setting of one category versus the background.

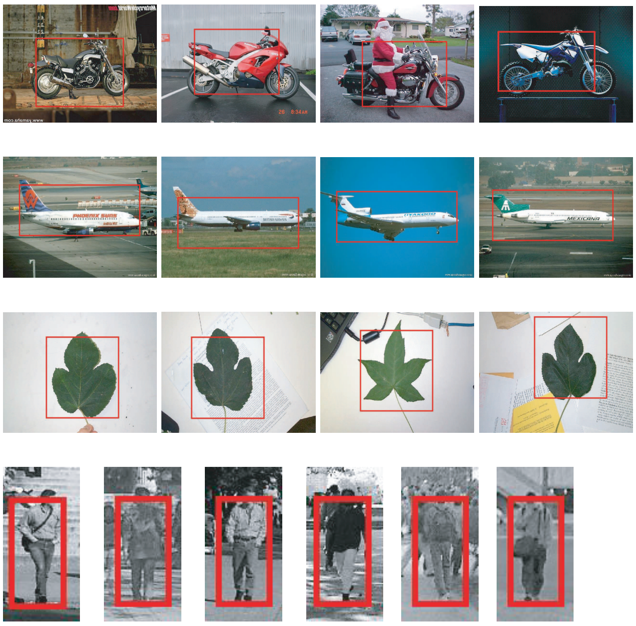

Figure 7: Some example results obtained using sliding window approach for handling scale variation. The results are for the Motorcycle, Airplane, Leafs, and Pedestrian data set.

5.1 Data Sets

We evaluated our object detection method on seven different data sets. Five of them were presented in [13] for object detection task. The data sets from Caltech include: Airplane, Car, Leaf, Face and Motorbike (http://www.vision.caltech.edu/). These data sets do not cover all arbitrary scales and poses. The other data sets include cars from UIUC and ETH Zurich, and pedestrians from MIT Center for Biological and Computational Learning (CBCL: http://cbcl.mit.edu/cbcl/). The negative examples were obtained randomly using all above mentioned data sets and Caltech background data set. Fig. 7 shows some example images from these data sets.

Data Set	Training Images	Testing Images
UIUC Cars	150	200
Caltech Car Rear	170	480
Caltech Airplane	200	874
Caltech Motorcycles	200	626
Caltech Faces	100	350
Caltech Leaves	50	137
MIT CBCL Pedestrians	200	724
ETH Zurich Cars	50	50

Table 2: Data sets used in experiments.

5.2 Experiments

The experiments were based on determining presence or absence of an object of interest. Each experiment was carried out by randomly splitting the data sets into two parts. One part was used for constructing the nonlinear subspaces, base learners and strong classifier while the other part was employed for testing. A description of the data set, as broken up into training and testing images, is given in the Table 2. Each experiment is performed using image gradients and image intensity as basic image representation. This allowed us to determine which basic representation is more useful for our detection framework. Furthermore, experiments were conducted using polynomial kernels (of degree two and three) and Gaussian kernel.

Table 3 displays the results of experiments on standard data sets. It lists the detection and false positive rates obtained for different image representations, including the number of principal components used for the experiments. Gaussian kernel was used for the experimentation in this instance. The table illustrates that both kinds of image representations are effective for classification, although gradient representation has slight advantage, visible from its corresponding high detection and low false positive rates. This can be attributed to the high quality of the images in the data sets.

Figure 8 relates the detection rate with the degree of nonlinear subspace (number of principal components). We obtained this graph by training our framework for different numbers of dimensions, starting from minimum value of 5 and going up to 150. The number of training and testing samples were kept the same throughout. From the graphs, we can observe that increasing subspace dimensions beyond a certain range does not seem to have strong influence on the performance of the boosted classifier. Detection rate does not change significantly after reaching its peak performance at around 35 to 40 principal components. This shows that after feature reduction through Kernel PCA the first 35 to 40 principal components provide us with a pool of discriminant features which are good enough for selection in the boosting stage. These features are now being continuously selected by the Adaboost, and therefore increasing subspace dimension is not contributing any more useful features. Fewer subspace dimensions in fact, allows our classifier to learn faster.

We show a comparison of our equal error rates with Fergus [13], Amores [22] and [14] in Table 4. The table illustrates that our approach performs well on these data sets. It improves on the performance rates for the motorbike and the face data sets. This shows that the combination of Kernel PCA and boosting is helping in extraction and selection of useful information from the underlying images and that the classifier is able to generalize to unseen examples.

It would be pertinent to discuss the relative disadvan-

Data Set	Gradient		Intensity		Principal Components
	Detection Rate	False Positive Rate	Detection Rate	False Positive Rate	
UIUC Cars	99.5%	0.8%	95%	6%	50
Caltech Car Rear	98%	1%	96%	2%	50
Caltech Airplane	98.5%	0.1%	89%	10%	50
Caltech Motorcycles	98.6%	0.3%	92%	12%	50
Caltech Faces	100%	0.2%	93%	6%	50
Caltech Leaves	100%	0.5%	96%	7%	30
MIT CBCL Pedestrians	100%	0.2%	97%	4%	50
ETH Zurich Cars	88%	9%	86%	14%	30

Table 3: This table summarizes the performance of our object detection framework on different data sets. These results reflect the best runs of our experiments conducted using the Gaussian kernel. Detections were performed at a single scale. The two sets of results used gradients and image gray levels as their initial image representations.

tages of using computationally efficient Haar features for a generic object detection framework. Haar features have recently become popular because of their tremendous success in face detection. These features make use of differences between regions where each region is uniform and different from the other. This holds true in cases of frontal faces and pedestrians, but for more complex objects like airplanes, we can observe that the uniformity criteria doesn't hold any more. Parts may appear or disappear for different instances of the object, for example, as is the case with airplane side windows and logos. One way to confront this problem would be to use thousands of images that cover all possible appearances, or train a separate detector for each possible point of view and shape. This entails gathering an even larger image database to cover all possible point of views and shapes. Approaches that adopted this line of attack have been pursued recently. However, they are not able to emulate success of frontal face detection because of the simplicity of Haar features and non-availability of large training databases. A drop in detection rates (81.5%) for profile face is reported by Viola [23] for multi-view face detection using Haar features. Similar trends are reflected in the results of Levi [24] for chair detection. Hence, there is still a need to develop algorithms that can learn from few examples and can generalize to a wide range of objects.

It should also be mentioned that our approach is easily extendable to a multi-class detection setting by using multi-class boosting algorithms. This will further aid in reducing the computational cost at testing time as only one classifier will be making all the decisions, unlike [19].

Figure 9 shows misclassified images from the airplane, motorbike, and face categories. These misclassifications were obtained while performing detections at only single scale. In the case of the airplane data set most misclassifications are because of the presence of more than one airplane which violates the learned structure, as all the training images contained a single airplane. In addition, Kernel PCA encodes the global structure of the object and therefore it will not be able to detect occluded objects. The part based approaches are a much better choice for such instances. It should be noted that our approach is extendible to multi-scale object detection by incorporating the sliding window approach.

Training time in our Matlab implementation varies from 15 to 30 minutes. We used Pentium4 desktop equipped with a 2.4 GHz processor and 512MB RAM. Testing code runs at 2 images/sec. In Fergus et. al [13] testing time is reported to be 2-3 second per image.

Data Set	Our Method	[13]	[22]	[14]
Caltech Car Rear	96%	90.3%	97%	97%
Caltech Airplane	90%	90.2%	92.7%	88.9%
Caltech Motorcycles	93.4%	92.5%	73.9%	92.2%
Caltech Faces	98%	96.4%	-	93.5%
Caltech Leaves	94.2%	-	97.8%	-

Table 4: Comparison of our results with [13, 22, 14].

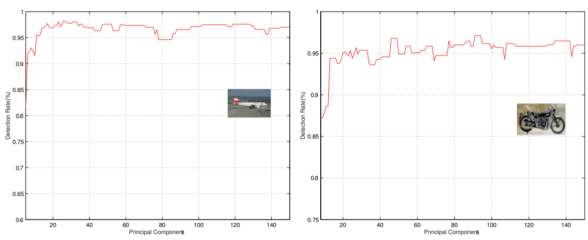

Figure 8: The plots show effect of subspace dimension on detection rate of the trained classifier for the Airplane and Motorcycle data set.

Figure 9: Some of the misclassified images for the Airplane, Motorcycle and Face data set.

6. Conclusion

We have proposed a new approach for object detection in a generic setting by integrating Kernel PCA with boosting. The key idea is to carry out nonlinear mapping of image features into a higher dimension where the classification task becomes simpler. Our approach is simple and easy to implement, yet it provides a highly accurate classifier for detection of the object of interest. In our experiments we have demonstrated its applicability to a number of challenging real world object categories. It showed much better results than any current state of the art method. In addition to high detection rates, it has several other advantages. It is scalable and can be extended to any object category. It requires a small set of examples for training. It is able to handle the nonlinear nature of features derived from the images. In the future we will try to extend the method where a single classifier can handle more than two categories. We are also planning to work on further increasing the computational efficiency of our framework.

References

[1] B. Schölkopf et. al., " Nonlinear component analysis as a kernel eigenvalue problem, " *Neural Computation*, 1998.

[2] B. Schölkopf and A. Smola, " Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond, "*MIT Press*, 2002.

[3] T. M. Cover, " Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition,"*IEEE Transaction on Electronic Computers*, 1965.

[4] S. Agarwal et. al., "Learning to detect objects in images via a sparse, part-based representation," *IEEE PAMI*, 26(11), pp. 1475-1490, 2004.

[5] H. Schneiderman and T. Kanade, "Object Detection Using the Statistics of Parts," *ICCV*, 2002.

[6] E. Bart et. al., "View-Invariant Recognition Using Corresponding Object Fragments," *In Proc. ECCV*, 2004.

[7] David G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, Vol. 60, no. 2,2004.

[8] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector, "*In Proc. ECCV*, 2002.

[9] T. Tuytelaars and Luc J. Van Gool, "Content-Based Image Retrieval Based on Local Affinely Invariant Regions, " *ICVIS*, 1999.

[10] M.H. Yang et. al., "Face Recognition using Kernel Eigenfaces," *International Conference on Image Processing*, 2000.

[11] B. Moghaddam, "Principal Manifolds and Probabilistic SubSpaces for Visual Recognition," *IEEE PAMI*, June 2002.

[12] M. Vidal-Naquet and S. Ullman, "Object Recognition with Informative Features and Linear Classification, "*International Conference of Computer Vision*, 2003.

[13] R. Fergus et. al., " Object Class Recognition by Unsupervised Scale-Invariant Learning,"*CVPR*, 2003.

[14] A. Opelt et. al., " Weak Hypotheses and Boosting for Generic Object Detection and Recognition,"*ECCV*, 2004.

[15] M. Turk and A. Pentland, "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience*, 1991.

[16] S. Romdhani et. al., " A Multi-view Nonlinear Active Shape Model using Kernel PCA,"*In Proc. BMVC* , 1999.

[17] D. Cremers et. al., " Nonlinear Shape Statistics via Kernel Spaces, " *DAGM-Symposium*, 2001.

[18] C. J. Twining C. J. Taylor, "Kernel Principal Component Analysis and the Construction of Non-Linear Active Shape Models, "*In Proc. BMVC*, 2001.

[19] S. Z. Li et. al., "Kernel Machine Based Learning for Multi-View Face Detection and Pose Estimation, "*ICCV*, 2001.

[20] C. Liu, "Gabor-Based Kernel PCA with Fractional Power Polynomial Models for Face Recognition, "*PAMI*, 26(5), 2004.

[21] Y. Freund and Robert E. Schapire, " Experiments with a New Boosting Algorithm,"*ICML*, 1996.

[22] J. Amores et. al., " Boosting contextual information in content-based image retrieval,"*ACM SIGMM*, 2004.

[23] P. Viola and M. Jones, " Fast Multi-View Face Detection,"*Technical Report TR2003-96, Mitsubishi Electric Research Laboratories*, 2003.

[24] K. Levi and Y. Weiss, " Learning Object Detection from a Small Number of Examples: The Importance of Good Features,"*CVPR*, 2004.