International Journal of Computational Vision and Biomechanics
Vol. 5 No. 2 (July-December, 2019)

Supervised Learning and Classification of Multi-class Objects Using Kernel Trick

M. Masudur Rahman & Anand Santhanam

Research School of Information Sciences and Engineering, The Australian National University
Canberra, ACT 2601, Australia, {Masud.Rahman;Anand}@rsise.anu.edu.au

Submitted: 15th March 2019 *Revised: 16th March 2019* *Accepted: 16th July 2019*

A novel multi-class object classification technique is proposed in this paper which uses kernel tricks for extracting nonlinear features and employs eigenvectors for separating object classes in the extracted features. The basis of the method is to employ kernel principal component analysis (KPCA) prior to using principal component analysis (PCA) for mapping input space to a higher dimensional feature space through a non-linear map. The conversion of non-linear data by gaussian kernel (via radial basis function) into linear form for its simplification and application of PCA is referred as 'kernel trick'. The employed data sets include our Smart Cars data which are images of the moving vehicles and some other generic databases. The obtained results emphasize the representation of the multi-class objects in their feature spaces, how the features are separated among the object classes and classification results. We also introduce eigenvectors as a classifier in this paper and, therefore, the proposed method does not require any conventional classifiers for the classification. Eigendimension matching conforms whether an image feature is 'in-space' or 'out-space' by comparing the dimensional ranges. The experimental results show the robustness of the feature separation using kernel tricks with our car database that leads to the cars' classification from its viewpoint. Further experiments with other generic databases and various traffic scenarios show the remarkable performance of separating and classifying objects of the the proposed method.

1. INTRODUCTION AND MOTIVATION

In computer vision problems, data structures include linear and non-linear classes. PCA (principal component analysis) is a classical and one of the most popular methods which successfully handles linear data sets in order to extract the features from the data. In fact, PCA finds an orthonormal transformation to maximize the scatter of the data samples by generating a set of orthonormal basis vectors. However, it fails to extract the right features in given data if the variation among the data samples is non-linear and the data samples vary in their appearance, pose and illumination conditions. Moreover, PCA is inadequate to analyze real data such as detection and classification of real objects with non-linear characteristics. Recently, KPCA has been re-invented and introduced for solving such problems in computer vision applications. The idea of the kernel trick is to project the input data into a high dimensional feature space F with a non-linear mapping at first, and then the data are analyzed in F so that non-linear relations of input data can be featured. An example result of KPCA's effort for clustering non-linear data is shown in Fig. 1 and the original data is shown in Fig. 1. The kernel trick has firstly been successfully implemented in SVM classifier for structural risk minimization [17]. Scholkopf [26] introduced the Kernel trick in machine vision problems. Since the kernel trick is used firstly to map the input data into the implicit feature space F, and then PCA is performed in F to extract non-linear principal components of the input data, KPCA not only inherits the good properties of PCA, but also possesses the capability of non-linear representation and classification. Therefore, it has been demonstrated to be more efficient than PCA in object recognition and classification [13, 32] to describe the real non-linear images.

Once feature extraction and representation of the objects are done, objects' classification is the next challenging step. In fact, feature extraction and representation are the preliminary steps for visual classification. However, we need to employ a classifier for learning the selected features. Many popular classifiers employed for this task that includes SVM (support vector machines), Bayes classifier, Perceptron, Fisher linear discriminant, etc. However, most of the classifiers work under the assumption that the features of the data sets should linearly be separable. In this particular study, our images of the objects include various views of the moving cars, airplanes, pedestrians, road side obstacles, etc. and they are taken under natural environments that includes occlusion, various lighting conditions and different geometrical condition. Therefore, a simple linear separation method for classifying the class and non-class features are not optimal [30] for this particular problem.

In this paper, we propose a kernel trick for overcoming the above mentioned shortcomings in dealing with our non-linear objects feature selection. This paper also introduces an eigenvector (or eigendimension) based classification technique which does not rely either on object class or category. We take the essential idea of the kernel trick as it converts the non-linear feature by mapping input space to a higher dimensional feature space, through a non-linear map, where the data is linearly separable by the traditional PCA. It is worthwhile to mention 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. This mapping will solve the problem of nonlinear distribution of low level image features [2]. In classification, the eigendimension matching algorithm requires only few eigenvectors for performing the classification of the selected features. As a result, we discard most of the dimensions (or eigenvectors) in the final stage of classification which certainly speeds up the classification process.

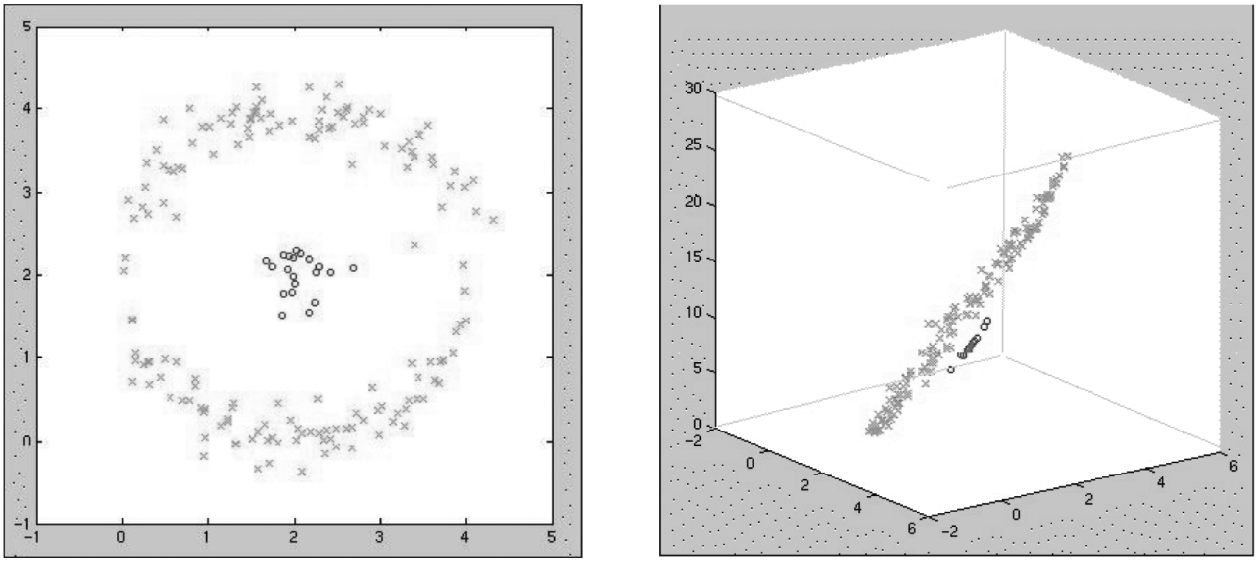
This paper is organized as follows: Section 2 outlines previous work related to this topic. Mathematical background of developing PCA and KPCA are given in Section 3. The proposed algorithm, the eigendimension matching, is given in Section 4. Section 5 describes details of the data sets and experimental procedures along with experimental results. Section 6 discusses various issues related to the proposed algorithm. Section 7 concludes this paper with hinting for its further extension.

2. PREVIOUS WORK

Detection and classification in an unconstrained environment is always a challenging problem. In the past, many fruitful methods have been developed for the object detection [8] and classification [14-16]. In general, object detection can mainly be done in two ways: part based [1, 2, 25, 26] and shape based object recognition [21-23]. In the part based approach, an object structure is encoded by using a set of patches covering important parts of the object. These patches themselves are detected using interest point operators, such as SIFT. In addition, affine invariant approach is also well known for object detection. In this approach, small patches are extracted from the image which are characterized by view point invariant descriptors [2]. These descriptors are used to match the object. Shape or appearance based methods use a global approach for capturing the object structure. PCA [7] is one of the powerful techniques for extracting global structure from a high dimensional data set. It has become well-known to the vision communities after its successful application for extracting facial features in the Eigenfaces method [27-28]. However, PCA is only capable of linear feature extraction and, therefore, it is not suitable for non-linear feature selection. KPCA, on the other hand, was introduced as a non-linear extension of PCA in spectrum analysis [17] and object analysis [26], which computes the principal components in a high dimensional feature space related to the input space. Our interest is on non-linear data and, therefore, we focus on using the kernel trick for our non-linear datasets. Yang [32] and Moghaddam [20] compared the face recognition performance and Eigenfaces method by using Kernel PCA with the cubic polynomial kernel and Gaussian kernel, respectively. Many other researchers [9, 11, 18] have attempted to employ this technique particularly for face detection. Baudat [3] applied kernel trick for generalizing the discriminant features. KPCA is also used to model the variability in classes of 3D-shapes [24, 29]. Liu [17] has recently employed it for recognition of facial expression using Gabor filters. Features derived by Gabor filters were non-linearly projected onto higher dimensional feature space by employing fractional power polynomial as a kernel function. Our main focus is to classify moving vehicles' image features which include multi-class and objects with non-linear characteristics.

For vehicle tracking and classification, we have seen a number of systems proposed [8, 10, 12, 14-16, 33]. Model based approaches have mostly been employed to track and detect vehicles. In [15, 16], a model based moving object classification approach that uses parameterized 3D models is proposed. 3D wireframe models [14-16, 33] have also been successfully employed for car tracking. Background subtraction models, when vehicles are well separated, have been explored in [8, 19]. To our knowledge, the only application of PCA for car classification is that of Bogomolov *et al.* [4]. They have employed this technique by combining motion and appearance features. It should be noted that little attention has been given in the mentioned works for analyzing non-linearity in the image features. For the classification, on the other hand, traditional classifiers employed for classifying the object features include Bayes classifier, SVM, Discriminant functions, etc. In most of the shape based models, similarity measures using L1 or L2 norm are employed for classifying the features and/or objects. In the PCA approach, many classifiers have been proposed previously, for example, Euclidean distance-based classifier [5, 21, 22], Mahalanobis distance-based classifier [5, 23], minimum subspace angle-based classifier [6] and support vector machine-based classifier [24]. These traditional classifiers only work with linearly separable datasets. However, we introduce a new classifier called eigendimension matching-based classifier that work for both linear and non-linear datasets. We have found a similar proposal in Weiss's works [31] on image segmentation using eigenvectors where he segmented the images by grouping method. The way he created and employed eigenvectors are completely different with our eigendimension matching algorithm.

This study will concentrate on separating and clustering the feature spaces by appropriately designing a radial basis function (rbf) for the gaussian kernel. Once the features are separated with respect to the datasets, we then define the maximum and minimum ranges of chosen eigendimensions to classify the feature spaces between car and non-car images and also car viewpoints. It is worthwhile to mention that the present study employs car's viewpoint images for training the system. We develop the kernelized feature space using these viewpoint images (negative and positive samples) and then classify the respective feature spaces by matching the eigendimensions. In the testing session, any viewpoint image is sufficient to detect the particular object in the database. The classifications do not depend only on two-class problems as proposed in [2] but it can successfully classify the multi-



(a) Non-linear data representation

(b) Clustered by KPCA]

Figure 1: Example of KPCA's robustness for clustering non-linear data.

class problems. As mentioned earlier, we employ only the gaussian kernel in this particular study defining a right rbf value for separating the multi-class feature space.

3. NON-LINEAR FEATURE EXTRACTION

Given a set of images $\mathbf{x}_k$ which is linearly separable, where $k = 1, ..., M$ and $\mathbf{x}_k \varepsilon \mathbf{R}^N$. The image set is centered by

$$\sum_{k=1}^{M} \mathbf{x}_k = 0 \tag{1}$$

PCA diagonalizes the covariance matrix,

$$C = \frac{1}{M}\sum_{j=1}^{M} \mathbf{x}_j \mathbf{x}_j^{\ T}. \tag{2}$$

Eigenvalue equation, $\lambda \upsilon = C\upsilon$ is solved where υ is eigenvector matrix for eigenvalues $\lambda \geq 0$ and $v\varepsilon\mathbf{R}^N$. First few eigenvectors are used as the basis vectors of the lower dimensional subspace. Eigen features are then derived by projecting the samples onto these basis vectors. Hence the equation 2 is equivalent to

$$\lambda(\mathbf{x}_k.v) = (\mathbf{x}_k.Cv) \tag{3}$$

for all $k = 1...M$.

The dot product F of the linear feature space can be computed as

$$F = v^T.\mathbf{x}_k \tag{4}$$

for all $k = 1...M$.

For computing non-linear feature space, Kernel PCA is performed by first mapping the data from input space to a higher dimensional feature space, i.e., using a map $\phi : \mathbb{R}^N \rightarrow F$, and then performing a linear PCA in F. Now, the covariance matrix in this new space F becomes

$$\overline{C} = \frac{1}{M}\sum_{j=1}^{M} \phi(x_j)\phi(x_j)^T \tag{5}$$

The eigenvalue problem now becomes $\lambda V = \overline{C}V$ for non-linear space. We do not have to explicitly compute the non-linear map ϕ. We can achieve the same goal by using a kernel functions. In certain cases, it is possible to compute dot products in these high dimensional feature spaces without actually having to explicitly carry out the mapping into these spaces. If the subsequent processing can be carried out using dot products exclusively, then we can work in the high dimensional space without explicitly mapping into the spaces. We employ dot products of the form

$$k(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle \tag{6}$$

which allow us to compute the value of the dot product in F without having to explicitly compute the map Φ, as shown in Eq. 6.

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 used kernels are shown in Table 1.

Table 1
Kernel Functions

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

After performing the kernel trick, it is important to note that all solutions $\mathbf{V}$ lie in the span of $\phi(x_1), \ldots, \phi(x_{\mathbf{M}})$. This has two useful consequences. First,

$$\lambda(\phi(\mathbf{x}_k) \cdot \mathbf{V}) = (\phi(\mathbf{x}_k) \cdot \overline{\mathbf{C}}\mathbf{V}) \tag{7}$$

for all $k = 1,..., M$

Second, there exist coefficients $\alpha_i (i = 1, \ldots, \mathbf{M})$ such that,

$$\mathbf{V} = \sum_{i=1}^{\mathbf{M}} \alpha_i \phi(\mathbf{x}_i) \tag{8}$$

Combining Eq. 7 and Eq. 8 we get the following Eq. 9

$$\lambda \sum_{i=1}^{\mathbf{M}} \alpha_i (\phi(\mathbf{x}_k) \cdot \phi(\mathbf{x}_i)) = \frac{1}{\mathbf{M}} \sum_{i=1}^{\mathbf{M}} \alpha_i (\phi(\mathbf{x}_k) \cdot \sum_{j=1}^{\mathbf{M}} \phi(\mathbf{x}_j))(\phi(\mathbf{x}_j) \cdot \phi(\mathbf{x}_i)) \tag{9}$$

Defining an M × M gram matrix K by the Eq. 10

$$\mathbf{K} := (\phi(x_i) \cdot \phi(x_j)) \tag{10}$$

we arrive at the eigenvalue problem for solving non-zero eigenvalues, as illustrated below in Eq. 11.

$$\mathbf{M}\lambda\mathbf{K}\alpha = \mathbf{K}^2\alpha \tag{11}$$

The next step is to normalize the eigenvectors $\alpha_1, \ldots, \alpha_{\mathbf{M}}$.

The last step to Kernel PCA involves principal component extraction. This is performed by computing the projection of a test sample $\phi(\mathbf{x})$ onto the eigenvectors $\mathbf{V}^k$ in $\mathbf{F}$:

$$(\mathbf{V}^k . \phi(\mathbf{x})) = \sum_{i=1}^{l} \alpha_i^k (\phi(\mathbf{x}_i) . \phi(\mathbf{x})). \tag{12}$$

The Eq. 12 can simply be re-written as

$$f = \mathbf{V}^T \phi(\mathbf{x}) = A^T B \tag{13}$$

where $A = \alpha_1, \ldots, \alpha_l$ and $B = [\phi(x_1)\phi(\mathbf{x}), \ldots, \phi(x_l)(\mathbf{x})]$.

- j is the image feature point
- ky is the total number of images in the dataset
- L is the number of selected eigendimensions

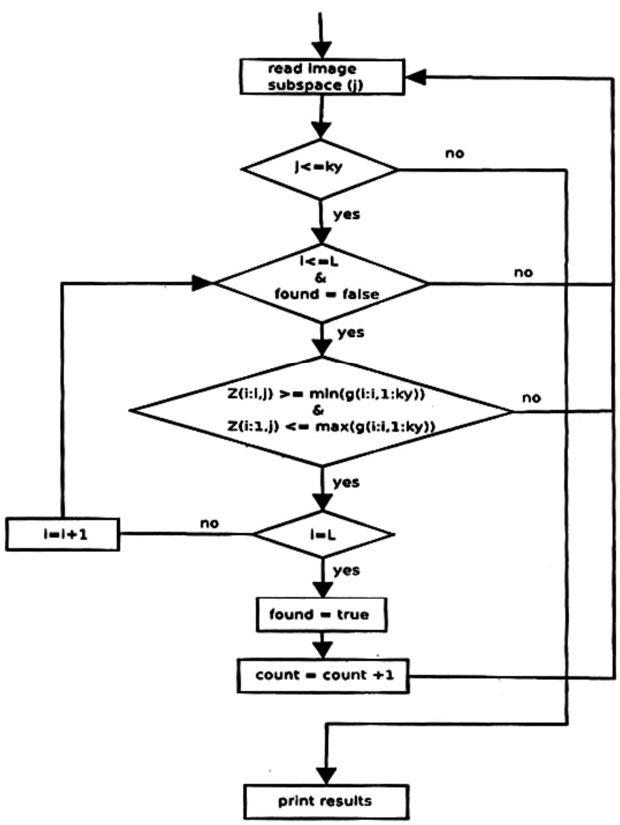


Figure 2: Program Structure for matching eigendimensions.

4. EIGENDIMENSION MATCHING ALGORITHM

The eigen decomposition of a raw data results in a feature space that can be defined by eigenvectors. Each eigenvector is called an eigendimension. Similar data will be heavily clustered around an area and dissimilar data will be separated. Separation of a region based on eigendimension is proposed in this paper. This is an attractive approach in that, it is based on matching of few eigendimensions for the classification. In this method, we need to calculate the minimum and maximum range of each eigendimension of the training datasets. The classification decisions are:

- Every selected eigendimension of the testing dataset should be greater than or equal to the minimum range of the corresponding eigendimension of the training dataset.
- Every selected eigendimension of the testing dataset should be lesser than or equal to the maximum range of the corresponding eigendimension of the training dataset.

It can be formulated as:

$$min(g_i) < Z_j < max(g_i) i = 1, \ldots, L \tag{14}$$

A flow chart of the proposed algorithm which uses the commonly known Matlab notations, is given in Fig. 2.

5. EXPERIMENT AND RESULTS

This section is dedicated towards describing the data sets used in the experiments, how our experiments were conducted and highlighting the obtained results. We have conducted experiments employing our databases along with some other standard databases. Our databases include Smart Cars and RTA database collected by our group but they are not available in public domain. The standard databases include Caltech car-back (www.vision.caltech.edu) and UIUC non-car. The main focus is our Smart Cars database where we have put maximum effort for developing and separating the feature space, and classifying the objects from their viewpoints. This experiment emphasizes on the classification of moving vehicle images, since we need on identifying the vehicles around our smart car for assisting the driver.

5.1 Data Set

We evaluated our object classification (mainly car images) using eigendimension matching on four different data sets. The Smart Cars data sets include various cars orientations (mainly car front, car back and car side) and these viewpoint images are used in the training stage. The unfamiliar car image is classified by comparing the training images stored in the kernelled feature space or called eigenspace. The images of RTA data sets are already segmented front view images of the moving cars taken at night. The Caltech car images are only the rear views of cars and, therefore, it represents and classifies the rear viewpoints only. A wide range of non-car images obtained from the UIUC database is used for evaluating the car and non-car classification in

Figure 3: Cars marked in a real-world scene for obtaining the orientations.

the experiment. The information about the images and data sets used in the experiment for training and testing stages is described in Table 2. In case of obtaining the Smart Cars database, we have manually marked and tracked the car images obtained from the camera. These are classified as three different views: *Car-rear*, *Car-front* and *Car-side*. The other images are considered as non-cars. Fig. 3 shows some of the cars marked in a real-world scene. Fig. 4 shows some of the sample images from different databases used in the experiments.

Table 2
Datasets and images used in the experiment

Data Set	*Training Images*	*Testing Images*
Smart Cars: Car-back	900	1800
Smart Cars: Car-front	45	90
Smart Cars: Car-side	34	34
Smart Cars: Non-car	1000	3000
Caltech: Car-back	170	480
RTA :Car-front	160	360
UIUC :Non-car	160	360

Figure 4: Some of the images used in the experiments.

Each of the images that has been used in our experiments are initially resized into a 32×32 image from its original resolution. The color images are also converted to grey-scale before the kernelled feature space is developed. However, the grey images obtained from some databases have been omitted in this pre-processing conversion step.

5.2 Experimental Details

To observe the effectiveness of PCA and KPCA's ability, we have designed two different experiments, (1) Car viewpoint classification and (2) Car vs Non-car classification. Since we have proposed and employed eigenvectors to separate the feature spaces of the objects and/or viewpoints, representation of the kernel feature spaces is also exhibited in this section. Some comparisons have also been made with conventional distance based classifiers in order to evaluate our proposed method. Therefore, the results are mainly presented in threefold: representation of kernel spaces, viewpoint classification and car and non-car classification. It is worth mentioning that all the experiments have been performed employing both PCA and KPCA. These have given us a comparison of results between the PCA and KPCA where KPCA has claimed to have better performance for separating the nonlinear features than PCA. The results have also shown false positive rates each time with the successful classification rates.

The training part of the data was used for computing the feature spaces and base learners, while the other part was employed for testing. For classifying car and non-car images, we employed the eigendimension matching algorithm which classifies them in their feature spaces. By considering only the maximum and minimum ranges of the feature space, we are able to classify images that lie within this range as Cars and the rest as Non-cars. We then compare the success of the eigendimension matching classifier against the conventional distance classifiers, Euclidean and Mahalanobis.

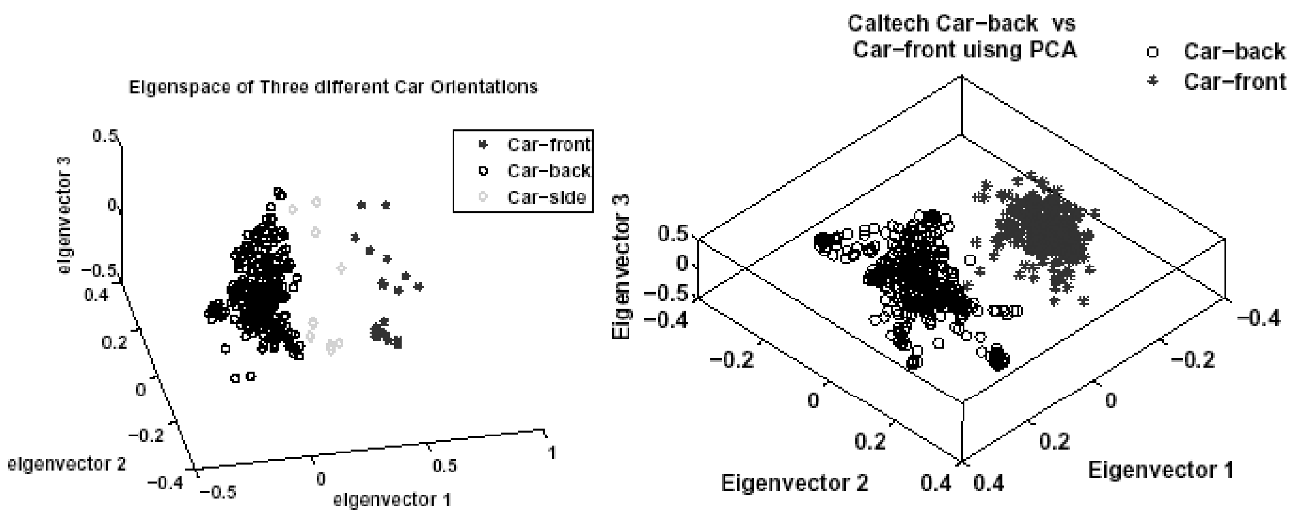


Figure 5: Representation of car views by PCA.

Table 3
Car viewpoint classification results by PCA

Data Sets	*Eigendimension Matching*	
	Classification Rate	*False Positives*
Smart Cars-Car-back	94%	3%
Smart Cars-Car-front	89%	5%
Smart Cars-Car-side	88%	5%
Caltech-Car-rear	96%	2%
RTA-Car-front	98%	1%

5.2.1 Car viewpoint classification with PCA

The objective of the feature space representation is to describe how the car's orientations of the images appears in the eigenspace. Fig. 5 (left) shows the graphical representation of the feature space by indicating three different locations of the respective views with the Smart Cars database. Fig. 5 (right) represents a comparison between Caltech's car-back and RTA's car-front. One can easily observe that the respective views have clearly separated so that they can readily be recognized. Table 3 and Table 4 show the classification rates achieved by the classifiers. Fig. 6 shows the classification rates in four different methods. The mean eigenspace method has also been placed where a mean is taken of some selected data sets and it is used for developing the feature space [22].

5.2.2 Car viewpoint classification with KPCA

This particular experiment shows how KPCA separates the respective features and it then becomes easy to employ our eigendimension matching classifier for the purpose of classification. Fig. 7 (left) shows the clustered spaces of the car orientations of the Smart Cars database that includes all image views of the cars. Fig. 7 (right) illustrates the comparison of the Caltech's car-back and RTA's car-front. These two figures clearly show the difference of employing KPCA for non-linear data sets where each data set is well separated from the other.

Table 4
Comparison results between Euclidean and Mahalanobis distance-based classifiers

Data Sets	*Euclidean Distance*		*Mahalanobis Distance*	
	Classification Rate	*False Positives*	*Classification Rate*	*False Positives*
Smart Cars-Car-back	56%	17.5%	87.7%	4%
Smart Cars-Car-front	62.2%	12%	49%	20%
Smart Cars-Car-side	76.5%	6.8%	10%	30%
Caltech-Car-rear	65%	10%	85%	5%
RTA-Car-front	82.3%	6%	96%	2%

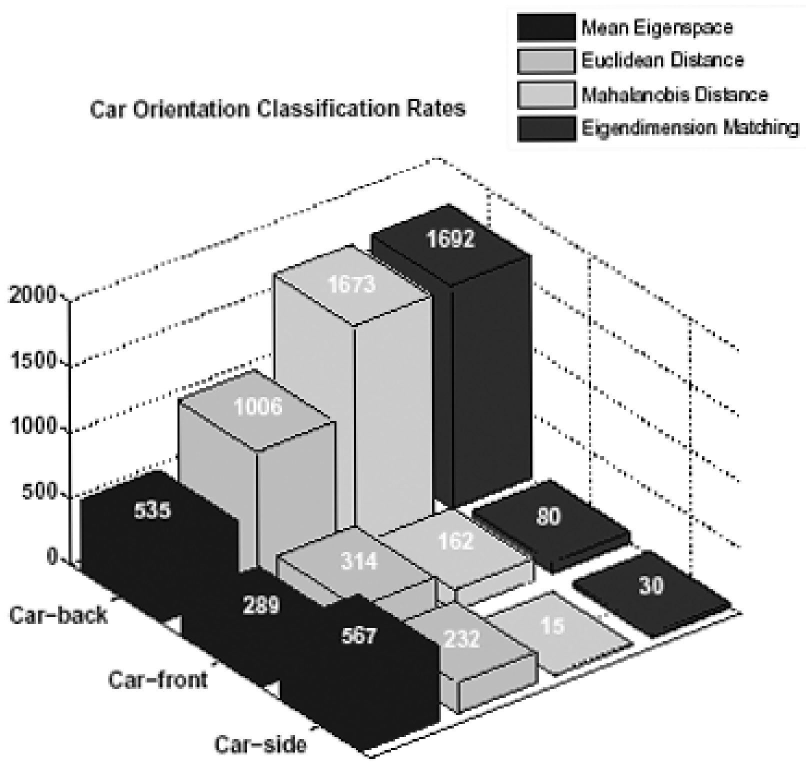


Figure 6: Classification rates of car viewpoints by the four different classifiers.

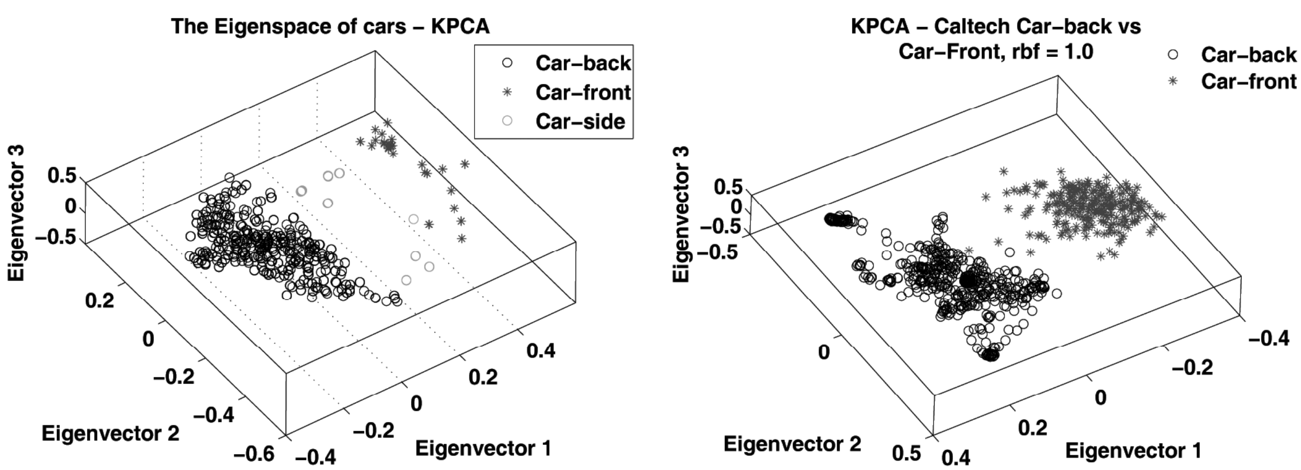


Figure 7: Clustered spaces of the car views by KPCA.

Table 5 presents the results for evaluating the Eigendimension Matching classifier using KPCA. Table 6 outlines results achieved by applying KPCA to the Euclidean and Mahalanobis distance based approaches.

Table 5
Results of the proposed method in KPCA

Data Sets	*Eigendimension Matching*	
	Classification Rate	*False Positives*
Smart Cars-Car-back	98%	1.2%
Smart Cars-Car-front	88%	7%
Smart Cars-Car-side	88%	11%
Caltech-Car-rear	96.9%	2.5%
RTA-Car-front	99.1%	0.1%

Table 6
Results of Euclidean and Mahalanobis distance based classifiers in KPCA

Data Sets	*Euclidean Distance*		*Mahalanobis Distance*	
	Classification Rate	*False Positives*	*Classification Rate*	*False Positives*
Smart Cars-Car-back	65%	34%	88%	9%
Smart Cars-Car-front	67%	35%	59%	42%
Smart Cars-Car-side	74.1%	27.3%	21%	73%
Caltech-Car-rear	65.8%	32%	87%	10%
RTA-Car-front	84%	15%	96.2%	1.6%

5.2.3 Car vs Non-car Classification with PCA

In this experiment, we have employed 1000 Non-car data samples for training the system and 3000 samples for the testing, as shown in Table 2. The objective of this investigation is to highlight the effectiveness of the proposed methods for car and non-car classification. Feature spaces of the Car and non-car images have been placed in Fig. 8 (top left) where they are clearly divided into their respective feature spaces. However, it may not always be possible to create such separated feature spaces due to various problems such as occlusion. Fig. 8 (top right) represents the RTA's Car-front and UIUC's Non-car feature space. Fig. 8 (lower row) illustrates the feature space of Caltech's Car-front and UIUC's Non-car. The obtained classification results employing the Eigendimension Matching approach on a number of data sets have been placed in Table 7. Table 8 gives details of the results achieved by applying PCA to the Euclidean and Mahalanobis distance based approaches.

Table 7
Evaluation of the Eigendimension Matching approach in PCA

Data Sets	*Eigendimension Matching*	
	Classification Rate	*False Positives*
Smart Cars-Car vs Non car	92%	6%
Smart Cars-Car vs UIUC	88%	13%
Caltech vs UIUC	86%	14%
RTA vs UIUC	98%	2.3%

Table 8
Evaluation of the Euclidean and Mahalanobis distance based classifiers in PCA

Data Sets	*Euclidean Distance*		*Mahalanobis Distance*	
	Classification Rate	*False Positives*	*Classification Rate*	*False Positives*
Smart Cars-Car vs Non car	85%	17.5%	66%	27%
Smart Cars-Car vs UIUC	79.4%	16%	55%	21%
Caltech vs UIUC	68.3%	34%	65%	20%
RTA vs UIUC	93%	8%	94%	5%

5.2.4 Car vs Non car Classification with KPCA

As the Gaussian Kernel plays an important role for clustering the feature spaces, the results from tuning the kernel have been highlighted in this subsection. Fig. 9 shows a sequential development of tuning the feature space of the Smart Cars database's Car and Non-car images by varying the radial basis function. Then, an illustration of the RTA's Car-front and UIUC's Non-car feature space has been given in Fig. 10(left). Finally, the Fig. 10(right) illustrates the feature space of Caltech's Car-front and UIUC's Non-car. A result of evaluations of the Eigendimension Matching approach using KPCA listed in Table 9. Table 10 details the result achieved by applying KPCA to the Euclidean and Mahalanobis distance based approaches.

Table 9
Evaluation of the Eigendimension Matching classifier in KPCA

Data Sets	*Eigendimension Matching*	
	Classification Rate	*False Positives*
Smart Cars-Car vs Non car	94%	5.2%
Smart Cars-Car vs UIUC	89%	9%
Caltech vs UIUC	87%	12%
RTA vs UIUC	98.2%	1%

Table 10
Evaluation of the Euclidean and Mahalanobis distance based classifiers in KPCA

Data Sets	*Euclidean Distance*		*Mahalanobis Distance*	
	Classification Rate	*False Positives*	*Classification Rate*	*False Positives*
Smart Cars-Car vs Non car	86.2%	12%	70%	4%
Smart Cars-Car vs UIUC	80%	16%	65%	15%
Caltech vs UIUC	70%	26%	75%	20%
RTA vs UIUC	94%	7%	95.2%	4.9%

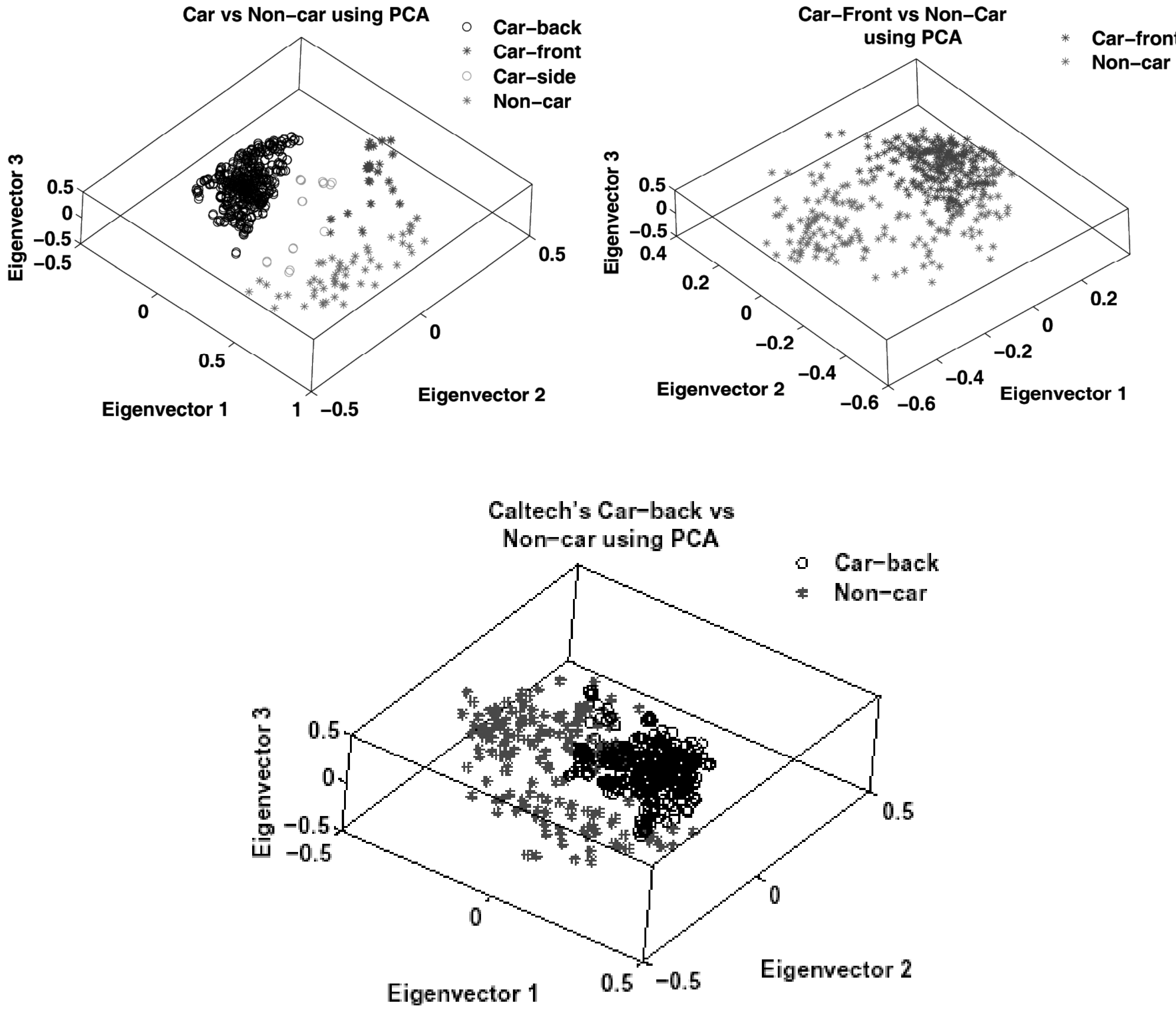
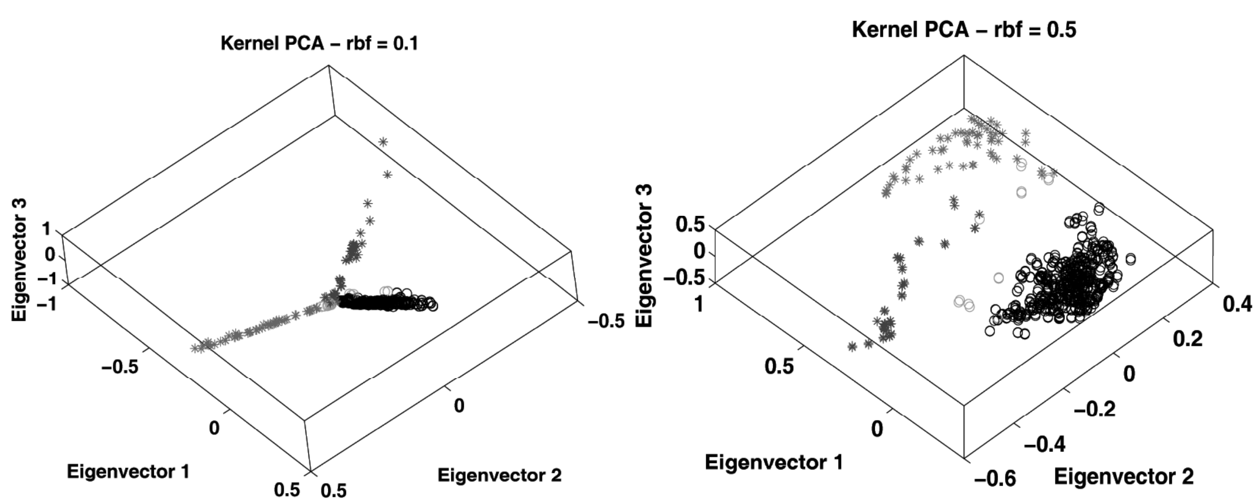


Figure 8: Representation of car and non car images in their feature space.



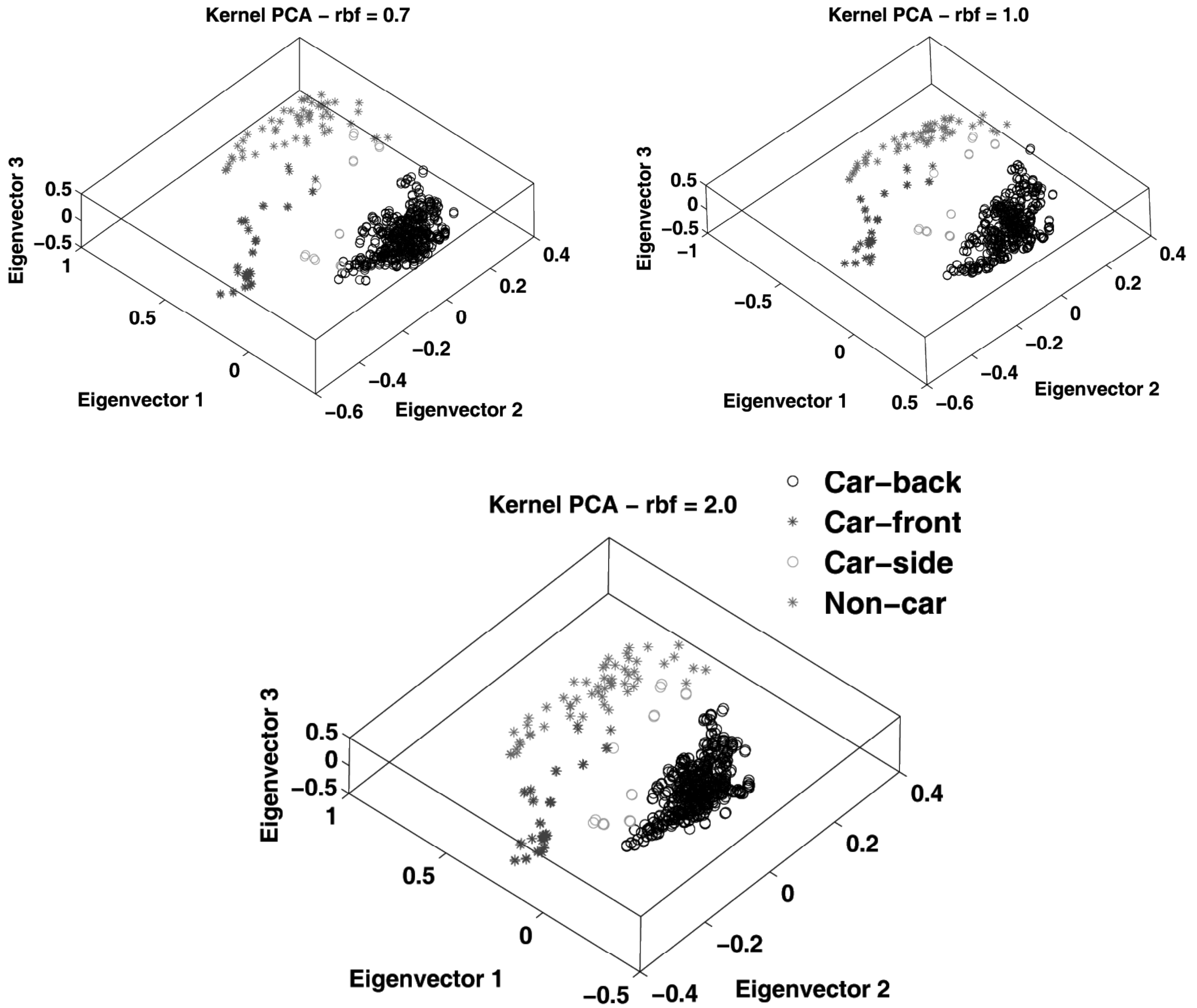


Figure 9: Representation of kernel feature of Smart Cars database.

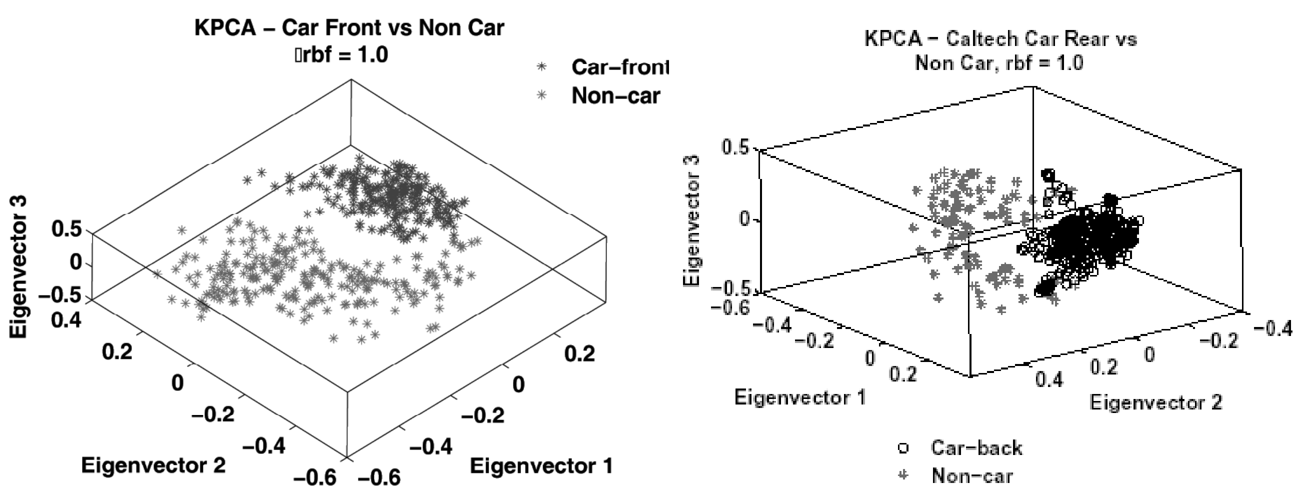


Figure 10: Representation of kernel feature between smart Car and UIUC's Non-car images.

6. DISCUSSION

Various issues related to the experimental complexity and obtained results have been discussed in this section. Among many issues, we mainly highlight the performances obtained in the experiments and comment the eigendimensions we employed for the matching algorithm. There are many problems that opens up in this paper and we will concentrate on these issues in our future work.

6.1 Effectiveness of PCA

As mentioned previously, PCA extracts the linear relationships that exist within a dataset. The experiment for the car viewpoint representation suggests that there were strong linear relationships between the different orientations of cars. This can be easily deduced from the fact that viewpoint classification achieved higher classification rates in comparison to car and non-car classification. It can also be noted that in car and non-car classification, the classification rates are lower because the non-car data set contains samples that do not have a high linear intra-class relationship. Hence, when linear relationships are non-existent in the data, PCA's effectiveness is limited.

6.2 Effectiveness of KPCA

The non-linear feature extractor shows a significant improvement over the linear feature extractor in PCA. It is important to realize that it is crucial to tune the Gaussian Kernel when employing KPCA to attain effective performance. We could visually observe the clear separation that exists in the feature space of the different data sets using KPCA. The classifiers' performance for car and non-car classification, in particular, was much better because KPCA could handle the non-linear intra-class relationships that existed in the UIUC non-car data set. We can also note that because of the presence of linear relationships in the data sets of the earlier experiment, the classification results of PCA and KPCA do not differ much. However, on the whole, as expected KPCA produced an improved feature space for the classification.

6.3 Variations Associated with Each of the Classifiers

Since minimum distance between two subspaces are important for classifying images in the PCA based classification, the limitations of the distance-based classifiers are graphically shown in Fig. 11. Fig. 11 (left) and Fig. 11 (right) show the distribution of the minimum distances between the training samples, i.e. car's views and testing samples, i.e. non-car images using Euclidean and Mahalanobis distances, respectively. It is quite difficult to select their distance threshold, since the minimum distances are not well separated. Consequently, poor classification results were obtained from the conventional methods. It should also be noted that Mahalanobis classifier requires a significantly larger training dataset than the other classifiers. This explains the reason behind Mahalanobis classifier's inability to distinguish the comparatively lower number of car-side samples. The obtained classification results from eigendimension matching, on the other hand, proved to be more effective than the conventional distance-based methods.

6.4 Selection of Eigenvectors

All the images in the data sets vary because of differences in illumination, small changes in the viewpoint and occlusion, none of which are relevant to the task of identifying the test image. The problem, of course, is knowing which eigenvectors correspond to useful information and which are simply meaningless variation. By looking at the images of specific eigenvectors, it is sometimes possible to determine what features are encoded in that eigenvector. Removing specific eigenvectors could in fact improve performance, by removing noise.

It is also worth mentioning that the eigendimension matching classifier needs only 3-7 eigenvectors whereas

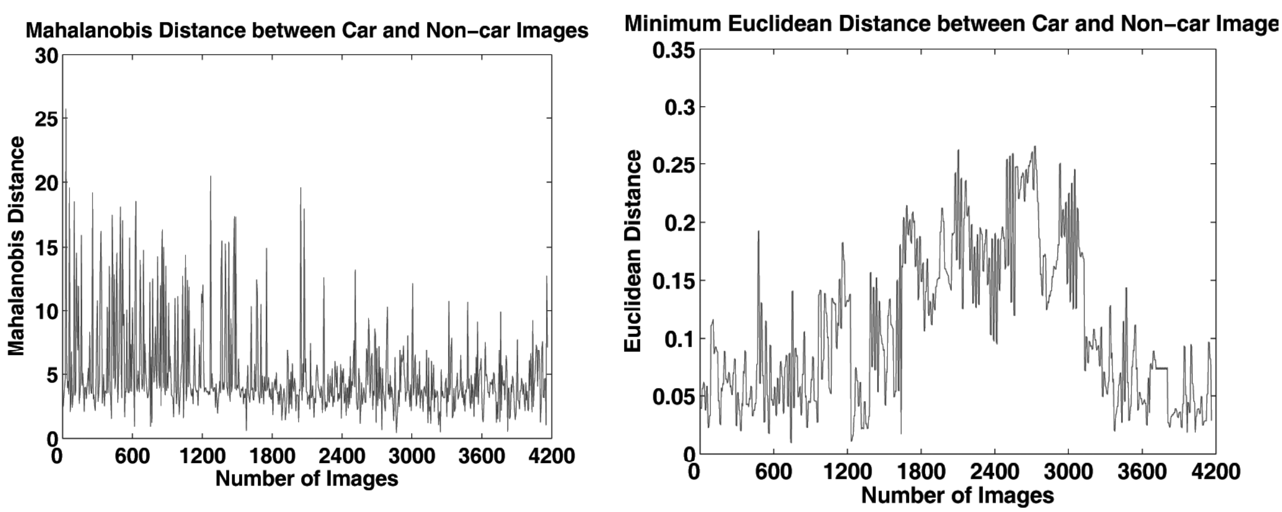


Figure 11: Graphical distribution of minimum distances between training and testing samples.

conventional methods require many more than this. Fig. 12 shows a relationship between requirement of eigenvectors and image characterisation that is suitable for the conventional methods.

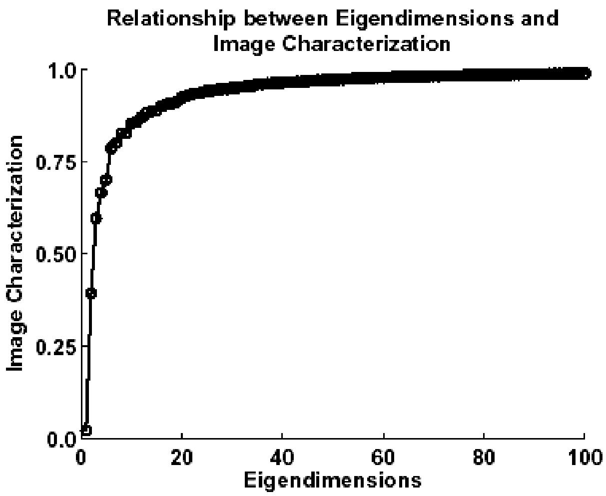


Figure 12: Relationship between eigenvectors and image characterization.

6.5 Increasing the Number of Classes

An increase in the number of classes will result in a more cluttered feature space. This in turn may have an adverse effect on the classification results. However, if the feature space is tuned to be well separated by choosing the appropriate rbf values, the eigendimension matching classification algorithm will produce better results.

6.6 Computational Time

Our training time in Matlab implementation software varies from 3 to 5 minutes with respect to the data sets that is much faster than the reported in [2], and the testing code runs approximately 2-3 images/sec under P4, 3.20 GHz processor with 1GB RAM.

7. CONCLUSION

Kernel based feature selection of non-linear objects and their classification have been introduced in this paper. The proposed kernel tricks have shown its robustness in extracting non-linear features. Eigenvectors have been used for multi-class object classification. These have provided a complete kernel based object classification system which can be useful for detecting the moving vehicles. This novel classification method distinguished moving vehicles from its viewpoints by comparing the feature space.

It introduced eigenvectors as a classifier by claiming that eigenvectors can work independently as a classifier. A series of results were obtained towards attaining the objectives and the representations of feature space clusters using both PCA and KPCA with respect to the data sets were graphically shown. The success of our classifier was also compared to the conventional methods. Our approaches achieved successful classification rates of up to 99.1% whereas the results of Euclidean and Mahalanobis distance-based classifiers were only up to 67%.

REFERENCES

[1] S. Agarwal. Learning to detect objects in images via a sparse part-based representation. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 26(11): 1475-1490, 2004.

[2] S. Ali and M. Shah. A supervised learning framework for generic object detection in images. In *International Conference on Computer Vision*, pages 1347-1354, 2005.

[3] G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12(10), 2000.

[4] Y. Bogomolov and *et al*. Classification of moving targets based on motion and appearance. In *British Machine Vision Conference*, 2003.

[5] J. R. del Solar and P. Navarrete. Eigenspace-based face recognition: A comparative study of different approaches. *IEEE Transaction on Sys., Man. and Cyb.*, 16(7): 817-830, 2005.

[6] C. W. D.J. Pack. Comparing two occluded object recognition systems using subspace graph search methods. In *In't Conf. on Signal and Image Processing*, 2003.

[7] R. C. Gonzalez and P. Wintz. *Digital Image Processing*. Addison-Wesley Publishing Company Limited, 1986.

[8] S. Gopte. Detection and classification of vehicles. *IEEE Transaction on Intelligent Transportation System*, 3(1): 37-47, 2002.

[9] G. Guo, S. Z. Li, and K. Chan. Face recognition by support vector machines. In *The 4th IEEE International Conference on Automatic Face and Gesture Recognition*, 2000.

[10] M. Hagg and H. Nagel. Combination of edge element and optical flow estimate for 3d model based vehicle tracking in traffic image sequences. *International Journal on Computer Vision*, 35(3): 295-319, 1999.

[11] B. Heisele and T. P. P. Ho. Face recognition with support vector machines: Global versus component-based approach. In *International Conference on Computer Vision*, pages 688-673, 2001.

[12] N. Kanhere and *et al*. Vehicle segmentation and tracking from a low-angle off-axis camera. In *International Conference on Computer Vision and Pattern Recognition*, pages 1152-1157, 2005.

[13] K. Kim, S. Park, and H. Kim. Kernel principal component analysis for texture classification. *IEEE Signal Processing Letters*, 8(2): 39-41, 2001.

[14] Z. W. Kim and J. Malik. Fast vehicle detection with probabilistic feature grouping and its application to vehicle tracking. In *International Conference on Computer Vision*, pages 521-528, 2003.

[15] D. Koller. Model based object tracking in monocular image sequences of road traffic scenes. *International Journal of Computer Vision*, 10(3): 257-281, 1993.

[16] D. Koller. Robust multiple car tracking with occlusion reasoning. In *European Conference on Computer Vision*, pages 189-196, 1994.

[17] C. Liu. Gabor-based kernel pca with fractional power polynomial models for face recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2004.

[18] Q. Liu, R. Huang, H. Lu, and S. Ma. Face recognition using kernel based fisher discriminant analysis. In *Proceedings of the 5th IEEE International Conference on Automatic Face and Gesture Recognition*, 2002.

[19] D. R. Magee. Tracking multiple vehicles using foreground, background and motion models. *Image and vision computing*, 22(2): 143-155, 2004.

[20] B. Moghaddam. Principal manifolds and probabilistic subspaces for visual recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 24(6): 780-788, 2002.

[21] H. Murase and S. K. Nayar. Visual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision*, 14(5): 39-50, 1995.

[22] M. M. Rahman and S. Ishikawa. Human motion recognition using an eigenspace. *Pattern Recognition Letters*, 26(6): 687-697, 2005.

[23] M. M. Rahman, K. Nakamura, and S. Ishikawa. Recognizing human behaviors employing universal eigenspace. In *International Conference on Pattern Recognition*, 2002.

[24] S. Romdhani and S. Gong. A multi-view non-linear active shape model using kernel pca. In *British Machine Vision Conference*, pages 483-492, 1999.

[25] H. Schneiderman and T. Kanade. Object detection using the statistics of parts. In *International Conference on Computer Vision*, 2002.

[26] B. Schölkopf, A. J. Smola, and K.-R. Müller. Non-linear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10: 1299-1319, 1998.

[27] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 1991.

[28] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *International Conference on Computer Vision and Pattern Recognition*, pages 586-591, 1991.

[29] C. J. Twining and C. J. Taylor. Kernel principal component analysis and the construction of non-linear active shape models. In *British Machine Vision Conference*, 2001.

[30] M. Vidal-Naquet and S. Ullman. Object recognition with informative features and linear classification. In *International Conference on Computer Vision*, 2003.

[31] Y. Weiss. Segmentation using eigenvectors: a unifying view. In *International Conference on Computer Vision*, pages 975-982, 1999.

[32] M. H. Yang and N. Ahuja. Face recognition using kernel eigenfaces. In *International Conference on Image Processing*, pages 37-40, 2000.

[33] T. Zhao and R. Nevatia. Car detection in low resolution aerial image. In *International Conference on Computer Vision*, pages 710-717, 2001.