Boosting Object Detection Using Feature Selection

Zehang Sun[1], George Bebis[1] and Ronald Miller[2]
[1]Computer Vision Lab. Department of Computer Science, University of Nevada, Reno
[2]Vehicle Design R&A Department, Ford Motor Company, Dearborn, MI
(zehang,bebis)@cs.unr.edu, rmille47@ford.com

Abstract

Feature subset selection has received considerable attention in the machine learning literature, however, it has not been fully explored or exploited in the computer vision area. In this paper, we consider the problem of object detection using Genetic Algorithms (GAs) for feature subset selection. We argue that feature selection is an important problem in object detection, and demonstrate that GAs provide a simple, general, and powerful framework for selecting good sets of features, leading to lower detection error rates. As a case study, we have chosen to perform feature extraction using the popular method of Principal Component Analysis (PCA) and classification using Support Vector Machines (SVMs). We have tested this framework on two difficult and practical object detection problems: vehicle detection and face detection. Experimental results demonstrate significant performance improvements in both cases.

I. Introduction

In recent years, object detection has received an increasing amount of attention in the literature. In this paper, we concentrate on two representative object detection problems using supervised learning: vehicle detection and face detection.

Robust and reliable vehicle detection in images acquired by a moving vehicle (i.e., on-road vehicle detection) is an important problem with application to driver assistance systems or autonomous, self-guided vehicles. This is a very challenging task in general[1]. Research on vehicle detection within the last ten years has been quite active. Matthews et al. use PCA for feature extraction and Neural Networks (NNs) for classification [2]. Goerick et al.[3] employ Local Orientation Coding (LOC) to encode edge information and NNs to learn the characteristics of vehicles. A statistical model was investigated by [4] Schneiderman et al., where PCA and wavelet features were used to represent vehicle and and nonvehicle appearances. A different statistical model was investigated by Weber et al [5]. They represented each vehicle image as a constellation of local features and used the EM algorithm to learn the parameters of the probability distribution of the constellations. An interest operator, followed by clustering, is used to identify a small number of local features in vehicle images. In [6], Papageorgiou et al. proposed using Haar wavelets for feature extraction and SVMs for classification. Sun et al. [1] fused Gabor and Haar wavelet features to improve the detection accuracy [1].

Similarly to on-road vehicle detection, face detection from a single image is a difficult task due to variability in scale, location, orientation, pose, race, facial expression, and occlusion. Rowley [7] proposed a NN-based face detection method, where pre-processed image intensity values were used to train a multilayer NN to learn the face and nonface patterns from face and nonface examples. Sung et al. [8] developed a distribution-based system which consists of two components, (i) a distribution-based model for face/nonface and (ii) a multilayer NN. SVMs were first applied to face detection by Osuna et al.[9]. In that work, the inputs to the SVM were again pre-processed image intensity values as in [7]. SVMs have also been used to detect faces using wavelet features [6]. Viola et al. [10] have recently developed a very fast face detection system using very simple features and the AdaBoost learning algorithm. Two recent comprehensive surveys on face detection can be found in [11], [12].

A. Feature Selection

The majority of real-world object detection problems require supervised learning where each training instance is associated with a class label. Building an object detection system under this scenario involves two main steps: (i) extracting a number of features and (ii) training a classifier using the extracted features to distinguish among different class instances. In most cases, relevant features are often unknown *a priori*. Often, a large number of features are extracted to better represent the target concepts, however, without employing some kind of feature selection strategy, many of them could be either redundant or even irrelevant to the classification task. As a result, the classifier might not be able to reach optimum performance.

It would be ideal if we could use only those features which have great separability power while ignoring or paying less attention to the rest. For example, in order to allow a vehicle detector to generalize nicely, it would be nice if we could exclude features encoding fine details which might be present in particular vehicles only. Finding out which feature to use for the task at hand is referred to as feature selection. Specifically, the problem of feature selection can be defined as follows: given a set of d features, select a subset of size m that leads to the smallest classification error.

A number of feature selection approaches have been proposed in the literature (Jain et al [13], Yang et al [14] for comprehensive surveys). According to the search strategy involved and expected results, feature selection algorithms fall into one of the three categories: (i) optimal feature selection, (ii) heuristic feature selection, and (iii) randomized feature selection.

Exhaustive search is the most straightforward approach to the optimal solution However, the number of possible subsets grows combinatorially, which makes the exhaustive search impractical for even moderate size of features. Sequential Forward Selection (SFS) and Sequential Backward Selection (SBS) are two well-known heuristic feature selection schemes [15]. Combining SFS and SBS gives birth to the "plus l-take away r" feature selection method[16], which first enlarges the feature subset by l using SFS and then deletes r features using SBS. Sequential Forward Floating search (SFFS) and Sequential Backward Floating Search (SBFS) [17] are generalizations of the"plus l - take away r" method . The values of l and r are determined automatically and updated dynamically in SFFS and SBFS. Since these strategies make local decisions, they cannot be expected to find globally optimal solutions.

Randomized search is another feature selection strategy. The relief algorithm [18] and several extension of it [19] are the typical randomized search approaches. Recently, GAs [20] have attracted more and more attention as an optimization tool for feature selection. Siedlecki et al [21] presented one of the earliest studies of GA-based feature selection in the context of k-nearest-neighbor classifiers. Yang et al [14] proposed a feature selection approach based on GAs using a NN classifier. However, by using the test set in the fitness function evaluation, they introduced some bias. Chtioui et al [22] investigated a GA-based feature selection scheme in a seed discrimination problem. In Sun et al. [23], [24] used GAs to select gender-orientated features to boost gender classification.

B. Proposed Work

In this paper, we propose using GAs to select good feature subsets to improve object detection. This is in contrast to common approaches in the literature which use all the features or a subset selected manually or based on some heuristics. An exemption to the above is the recent work of Viola et al. [10] where increasingly more complex classifiers are combined in a cascade using the AdaBoost algorithm. The boosting process they used selects a weak classifier at each stage of the cascade which can been seen as a feature selection process. The proposed approach has the advantage that it is simple, general, and powerful. The work proposed here has similarities with the work of Sun et al. [23], [24], however, the size of the two classes considered here (e.g., object vs non-object) are in principle very different.

To demonstrate the proposed approach, we have considered the well known methods of PCA for feature extraction and SVM for classification. Feature extraction using PCA has received considerable attention in the computer vision area [25], [11], [12]. Feature extraction using PCA entails to representing an image in a low dimensional space spanned by the principal components of the covariance matrix of the data. Although PCA provides a way to represent an image in an optimum way, several studies have shown that not all of the principal eigenvectors encode useful information for classification purposes (e.g., Swets et al [26] have reported that several principal eigenvectors seem to encode mostly lighting information). Here, GAs are used to select a subset of features from the low dimensional representation of the image by disregarding certain eigenvectors that do not seem to encode important information about the target concepts. This framework has been tested on two difficult object detection problems: vehicle detection and face detection.

The rest of the paper is organized as follows: In Section B, we present a brief overview of eigenspace representations. Section C presents a brief review on SVMs. Feature selection in the context of face and vehicle detections are addressed in Section D. In section E, we present the genetic search approach for eigen-feature selection. The proposed framework is experimentally tested in Section F(vehicle detection) and Section G. Finally, Section H summarizes the main results of the paper and presents possible directions for future work.

II. Eigenspace Representation

Eigenspace representations of images use PCA [25] to linearly project an image in a low-dimensional space. This space is spanned by the principal components (i.e., eigenvectors corresponding to the largest eigenvalues) of the distribution of the training images. After an image has been projected in the eigenspace, a feature vector containing the coefficients of the projection is used to represent the image. We refer to these features as eigen-features. Here, we just summarize the main ideas [25]:

Representing each image $I(x, y)$ as a $N \times N$ vector Γ_i, first the average face Ψ is computed:

$$\Psi = \frac{1}{R}\sum_{i=1}^{R}\Gamma_i \qquad (1)$$

where R is the number of faces in the training set. Next, the difference Φ of each face from the average face is computed: $\Phi_i = \Gamma_i - \Psi$. Then the covariance matrix is estimated by:

$$C = \frac{1}{R}\sum_{i=1}^{R}\Phi_i\Phi_i^T = AA^T, \qquad (2)$$

where, $A = [\Phi_1\Phi_2 \dots \Phi_R]$. The eigenspace can then be defined by computing the eigenvectors μ_i of C. Since C is very large ($N \times N$), computing its eigenvector will be very expensive. Instead, we can compute ν_i, the eigenvectors of A^TA, an $R \times R$ matrix. Then μ_i can be computed from ν_i as follows:

$$\mu_i = \sum_{j=1}^{R}\nu_{ij}\Phi_j, j = 1 \dots R. \qquad (3)$$

Usually, we only need to keep a smaller number of eigenvectors R_k corresponding to the largest eigenvalues. Given a new image, Γ, we subtract the mean ($\Phi = \Gamma - \Psi$) and compute the projection:

$$\widetilde{\Phi} = \sum_{i=1}^{R_k} w_i \mu_i. \tag{4}$$

where $w_i = \mu_i^T \Gamma$ are the coefficients of the projection. In this paper, $\{w_i\}$ are our eigen-features.

The projection coefficients allow us to represent images as linear combinations of the eigenvectors. It is well known that the projection coefficients define a compact image representation and that a given image can be reconstructed from its projection coefficients and the eigenvectors (i.e., basis). The eigenspace representation of images is very powerful and has been used in various applications such as image compression and face recognition.

III. Support Vector Machines

SVMs are primarily two-class classifiers that have been shown to be an attractive and more systematic approach to learning linear or non-linear decision boundaries [27]. Given a set of points, which belong to either of two classes, *SVM* finds the hyperplane leaving the largest possible fraction of points of the same class on the same side, while maximizing the distance of either class from the hyperplane. This is equivalent to performing structural risk minimization to achieve good generalization [27]. Assuming l examples from two classes

$$(x_1, y_1)(x_2, y_2)...(x_l, y_l), \quad x_i \in R^N, y_i \in \{-1, +1\} \tag{5}$$

finding the optimal hyper-plane implies solving a constrained optimization problem using quadratic programming. The optimization criterion is the width of the margin between the classes. The discriminate hyperplane is defined as:

$$f(x) = \sum_{i=1}^{l} y_i a_i k(x, x_i) + b \tag{6}$$

where $k(x, x_i)$ is a kernel function and the sign of $f(x)$ indicates the membership of x. Constructing the optimal hyperplane is equivalent to find all the nonzero a_i. Any data point x_i corresponding to a nonzero a_i is a support vector of the optimal hyperplane.

Suitable kernel functions can be expressed as a dot product in some space and satisfy the Mercer's condition [27]. By using different kernels, *SVMs* implement a variety of learning machines (e.g., a sigmoidal kernel corresponding to a two-layer sigmoidal neural network while a Gaussian kernel corresponding to a radial basis function (*RBF*) neural network). The Gaussian radial basis kernel is given by

$$k(x, x_i) = \exp(-\frac{\| x - x_i \|^2}{2\delta^2}) \tag{7}$$

The Gaussian kernel is used in this study (i.e., our experiments have shown that the Gaussian kernel outperforms other kernels in the context of our application).

IV. Feature Selection

We represent each image in terms of a set of eigenfeatures. It has been found in several studies that different eigenvectors encode different kind of information. For example, the first few eigenvectors seem to encode lighting while other eigenvectors seem to encode some local features. We have made very similar observations in our case by analyzing the eigenvectors obtained from our training sets. Fig.1, for example, shows some of the eigenvectors computed from our vehicle detection training data. Obviously, eigenvectors 2 and 4 encode illumination information, while eigenvectors 8 and 12 encode some local information. Similar comments can be made for the eigenvectors derived from our face detection training data set Fig.2. Once again eigenvectors 2 and 5 seem to encode mostly illumination while eigenvectors 8, 9 and 22 seem to encode mostly local information. Eigenvector 150 seems to encode mostly noise in both cases.

Although many eigen-features might be are very important for recognition purposes, they might actually confuse the classifier in other applications such as in detection. For instance, the general shape might be more important information for detecting a vehicle in images caught under unconstrained environments than the illumination or some local features. In this paper, we consider using GAs to select a good subset of eigen-features in order to boost object detection performance.

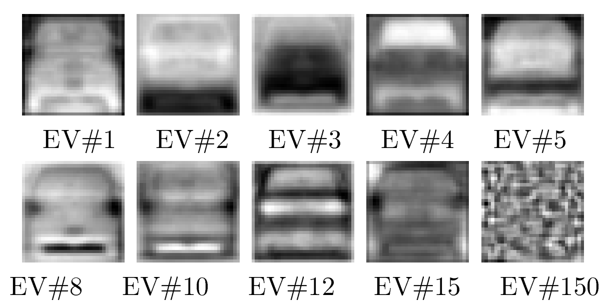


Fig. 1. Eigenvectors from vehicle detection dataset

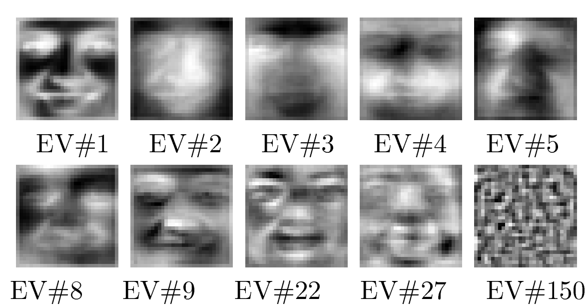


Fig. 2. Eigenvectors from face detection dataset

V. Genetic Feature Selection

A. A Brief Review of GAs

GAs are a class of optimization procedures inspired by the mechanisms of natural selection [20]. GAs operate iteratively on a population of structures, each of which represents a candidate solution to the problem, encoded as a string of symbols (chromosome). A randomly generated set of such strings forms the initial population from which the GA starts its search. Three basic genetic operators guide this search: selection, crossover and mutation

B. Overview of the Proposed Method

The main steps of the proposed method are as follows:
(i) Eigen-feature extraction using PCA
(ii) Optimal eigen-feature subset selection using GAs
(iii) Training of the SVMs
(vi) Classification of novel images

A binary encoding scheme is used to represent the presence or absence of a particular eigenvector from the linear expansion of the training images. Each individual in a generation represents an eigen-feature subset which is used to train SVMs. The performance of the SVMs classifier on the validation data set is used to provide a measure of fitness used to guide the GA.

C. Encoding

Each image is represented as a vector of eigen-features which are the coefficients of the linear expansion of the image in the eigenspace. In our encoding scheme, the chromosome is a bit string whose length is determined by the number of eigenvectors. Each eigenvector, computed using PCA, is associated with one bit in the string. If the ith bit is 1, then the i^{th} eigenvector is selected, otherwise, that component is ignored. Each chromosome thus represents a different eigen-feature subset.

D. Fitness Evaluation

The goal of feature subset selection is to use fewer features to achieve the same or better performance. Therefore, the fitness evaluation contains two terms: (a) accuracy and (b) number of features used. Only the features in the eigen-feature subset encoded by an individual are used to train the SVMs classifier. The performance of the SVMs is estimated using a validation data set and used to guide the GA. Each feature subset contains a certain number of features. If two subsets achieve the same performance, while containing different number of features, the subset with fewer features is preferred. Between accuracy and feature subset size, accuracy is our major concern. Combining these two terms, the fitness function is given as:

$$fitness = 10^4 Accuracy + 0.5 Zeros \tag{8}$$

where *Accuracy* is the accuracy rate that an individual achieves, and *Zeros* is the number of zeros in the chromosome. The accuracy ranges roughly from 0.5 to 1 (i.e., the first term assumes values in the interval 5000 to 10000). The number of zeros ranges from 0 to L where L is the length of the chromosome (i.e., the second term assumes values in the interval 0 to 100 since $L = 200$ here).

Overall, the higher the accuracy is, the higher the fitness is. Also, the fewer the number of features used the higher the number of zeros and as a result, the higher the fitness. It should be noted that individuals with higher accuracy will outweigh individuals with lower accuracy, no matter how many features they contain.

E. Initial Population

In general, the initial population is generated randomly, (e.g., each bit in an individual is set by flipping a coin). In this way, however, we will end up with a population where each individual contains the same number of 1's and 0's on the average. To explore subsets of different numbers of features, the number of 1's for each individual is generated randomly. Then, the 1's are randomly scattered in the chromosome.

F. Crossover

In general, we do not know how the eigenfeatures depend on each other. If dependent features are far apart in the chromosome, it is more probable that traditional 1-point crossover, will destroy the schemata. To avoid this problem, uniform crossover is used here.

G. Mutation

Mutation is a very low probability operator and just flips a specific bit. It plays the role of restoring lost genetic material. Our selection strategy was cross generational. Assuming a population of size N , the offspring double the size of the population and we select the best N individuals from the combined parent-offspring population.

VI. Genetic Feature Subset Selection For Vehicle Detection

In this subsection, we consider the problem of vehicle detection from gray-scale images. A first step of any vehicle detection system is hypothesizing the locations in images where vehicles are present. Then, verification is applied to test the hypotheses. Both steps are equally important and challenging. Approaches to generate the hypothetical locations of vehicles in images include using motion information, symmetry, shadows, and vertical/horizontal edges. Our emphasis here is on improving the performance of the verification step by selecting some representative features.

A. Vehicle Data

The images used in our experiments were collected in Dearborn, Michigan during two different sessions, one in the Summer of 2001 and one in the Fall of 2001. To ensure a good variety of data in each session, the images were caught during different times, different days, and on five different highways. The training set contains subimages of rear vehicle views and non-vehicles which were extracted manually from the Fall 2001 data set. A total of 1051 vehicle subimages and 1051 non-vehicle subimages were extracted (see Figure 3). In [6], the subimages were aligned by wrapping the bumpers to approximately the same position. We have not attempted to align the data in our case since alignment requires detecting certain features on the vehicle accurately. Moreover, we believe that some variability in the extraction of the subimages can actually improve performance. Each subimage in the training and test sets was scaled to 32×32 and preprocessed to account for different lighting conditions and contrast followed the method suggested in [28].

To evaluate the performance of the proposed approach, the average error (ER) was recorded using a three-fold cross-validation procedure. Specifically, we split the training dataset randomly three times (*Set1*, *Set2* and *Set3*) by keeping 80% of the vehicle subimages and 80% of the non-vehicle subimages (i.e., 841 vehicle subimages and 841 non-vehicle subimages) for training. The rest 20% of the data was used for validation during feature selection. For testing, we used a fixed set of 231 vehicle and non-vehicle subimages which were extracted from the Summer 2001 data set.

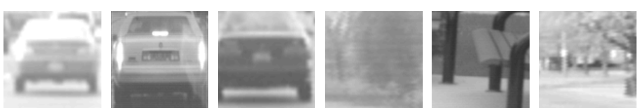


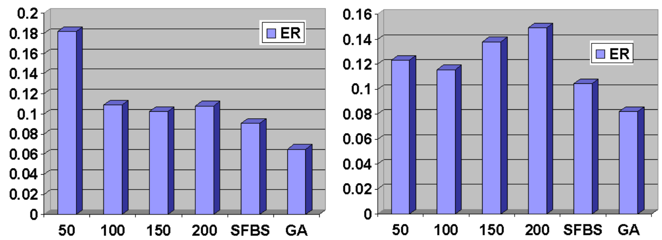
Fig. 3. Examples of vehicle and nonvehicle images used for training.

B. Experimental Results

We have performed a number of experiments and comparisons to demonstrate the importance of the feature selection for vehicle detection. First, SVMs were tested using manually selected eigen-features. We ran several experiments by varying the number of eigenvector from 50 to 200, Fig.4.a summarizes the results. By using the top 50, 100, 150, and 200 eigenvectors, the error rates are 18.21%, 10.89%, 10.24%, and 10.80% respectively.

For comparison purposes, we also implemented the SFBS feature selection method. SFBS is an advanced version of $plus\ l - take\ away\ r$ method that first enlarges the feature subset by l features using forward selection and then removes r features using backward selection. In contrast to the $plus\ l - take\ away\ r$, the number of forward and backward steps in SFBS is dynamically controlled and updated by the classifier's performance. The average number of the features selected by SFBS was 87, and the error rate was 9.07%, which is only slightly better than the best results (10.24%) using manually selected features.

In the last experiment, we used GAs to select an optimum subset of eigen-features. The GA parameters we used are as follows: population size: 2000, number of generations:200, crossover rate:0.66 and mutation rate: 0.04. In all cases, the GA converged in less than 200 generations. Fig.4.a shows the error rate using the GA-selected feature subsets. Using the feature subset, the SVM achieved a 6.49% error rate, which is better than both using the manually selected feature subsets or subsets based on SFBS. In terms of the number of features contained in the feature subsets, SFBS preserves 87 features, which is 43.5% of the complete feature set while GAs keep only 46 features, 23% of the compete feature set.



(a) (b)

Fig. 4. Detection error rates of various methods. a. Vehicle detection results, b. Face detection results.

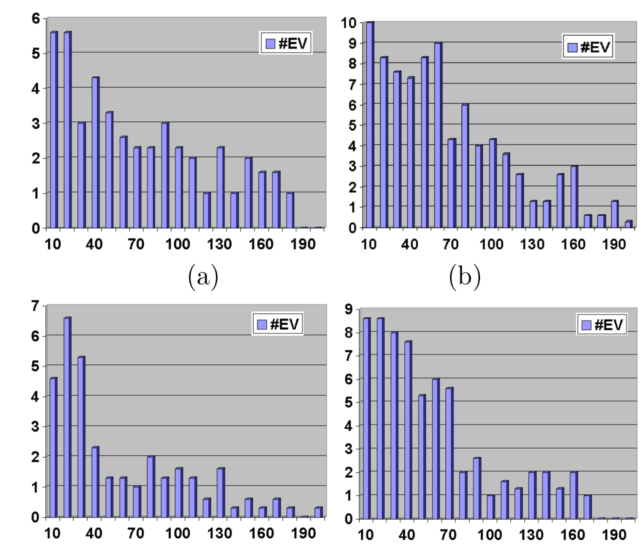


Fig. 5. The distributions of eigenvectors selected by (a) GAs for vehicle detection , (b) SFBS for vehicle detection, (c) GAs for face detection, (d) SFBS for face detection.

To get an idea regarding the optimal set of eigenvectors selected by GAs or SFBS, we compute the histogram (see Fig.5), showing the average distribution of the selected eigenvectors, (i.e. over the three training sets). The x-axis corresponds to the eigenvectors, ordered by their eigenvalues, and has been divided into intervals of length 10.

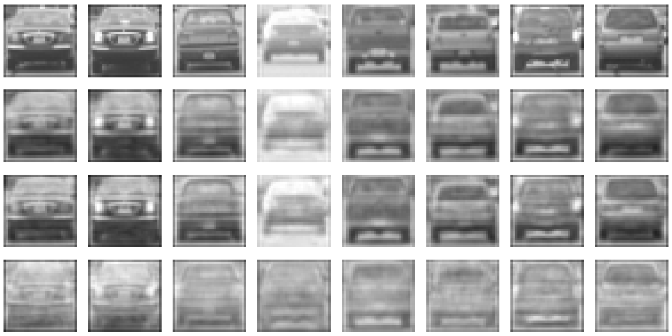


Fig. 6. Reconstructed images using the selected feature subsets. First row: original images; Second row: using top 50 eigenvectors; Third row: using the eigenvectors selected by SFBS; Fourth row: using the eigenvectors selected by GAs.

The y-axis corresponds to the average number of times an eigenvector within some interval has been selected by the GAs (or SFBS correspondingly) in the final solution. Fig.5.a-b illustrate that the feature subsets selected by GAs is sparser than those by SFBS. As we have discussed in Section D, different eigenvectors seems to encode different kind of information. For visualization purposes, we have reconstructed the vehicle subimages using the selected eigenvectors only (Fig.6). Two interesting comments can be made through observing the reconstructed images using feature subsets selected by GAs. First of all, it is obvious that the reconstructed images contain much less details - they all look fairly similar to each other. These features can be thought as features that represent the"concept vehicle", but not individual vehicles. In contrast, the reconstructed images using top 50 eigenvectors or feature subset selected by SFBS method do reveal more vehicle identity information as can be seen from the images in the second and third rows. Second, the eigenvectors encoding features unimportant to representing the vehicle class, such as illumination, seem to have been discarded by the GA. This is obvious by observing the reconstructed images in the fourth row - all of them seem to have been normalized with respect to illumination (notice in particular the image in the fourth column).

VII. Genetic Feature Subset Selection For Face Detection

To detect faces in an image, a fixed window is run across the input image. Each time, the contents of the window are fed to a classifier which verifies whether there is a face in the window or not. To account for differences in face size, the input image is represented at different scales and the same procedure is repeated at each scale. Alternatively, candidate face locations in an image can be found using color, texture, or motion information. As it was the case for vehicle detection, here we concentrate on the verification part only.

A. Face Data

The training set contains 616 faces and 616 nonfaces subimages which were extracted manually from a gender dataset and the CMU face detection dataset [7]. For testing, we used a fixed set of 268 face and non-face subimages which were also extracted from disjoint set of images from the CMU face detection data set. Each subimage in the training and test sets was scaled to 32×32 and preprocessed to account for different lighting conditions and contrast [28].

To evaluate the performance of the proposed approach, we used a three-fold cross-validation procedure, splitting the training dataset randomly three times (*Set1*, *Set2* and *Set3*) by keeping 84% of the face subimages and 84% of the non-face subimages (i.e., 516 vehicle subimages and 516 non-face subimages) for training. The rest 16% of the data was used for validation during feature selection.

Fig. 7. Examples of face and nonface images used for training.

B. Experimental Results

For comparison purposes, SVMs were first tested using manually selected eigen-features. We ran several experiments, as we did for vehicle detection, by varying the number of eigenvectors from 50 to 200, Fig.4.b summarizes the results. By using the top 50, 100, 150, and 200 eigenvectors, the error rates are 12.31%, 11.57%, 13.81%, and 14.93% respectively.

SFBS was also applied on the face detection data. The average number of features selected by SFBS was 68, and error rate was 10.45%, which is better than the best results (11.57%) using manually selected features.

In the last experiment, we used GAs to select an optimum subset of eigen-features for SVMs. The GA parameters we used are exactly the same to those used in the vehicle detection experiments. Fig.4.b shows the error rate using the GAs selected feature subsets. The proposed method achieves 8.21% error rate, which is better than those using manually selected feature subsets and features selected using SFBS. In terms of number of features contained in the feature subsets, SFBS preserved 68 features, which is 34% of the complete feature set while GAs kept only 34 features, that is, 17% of the compete feature set.

The average distributions of the selected eigenvectors by GAs and SFBS are shown in Fig.5.c-d. The reconstructed images using the selected eigenvectors are shown in Fig.8. It can be observed that the reconstructed images using the GA-selected features appear more blurred (i.e., have less details) than the original images or the ones reconstructed using the manually selected or SFBS-based eigeneatures. Obviously, identity information has not been preserved which might be the key to successful face detection. Another interesting observation is that the reconstructed face images using the GA-selected features look more normalized (e.g., notice the image in the fifth column where it seems that feature selection has disregarded the effects of rotation).

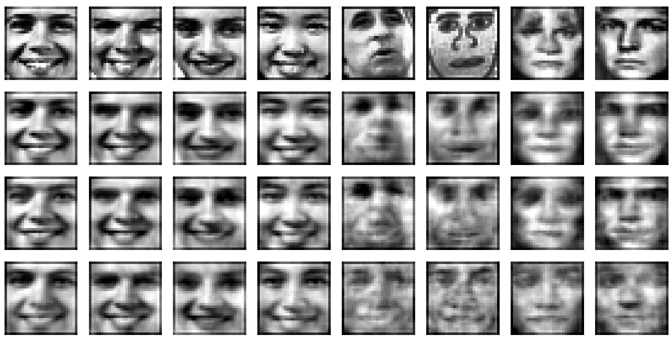


Fig. 8. Reconstructed images using the selected feature subsets. First row: original images; Second row: using top 50 eigenvectors; Third row: using the eigenvectors selected by SFBS; Fourth row: using the eigenvectors selected by GAs.

VIII. Conclusions

We have proposed using feature selection for boosting the performance of object detection. In particular, we proposed using GAs to select detection-specific feature subsets in order to improve the performance of object detection. Our results demonstrate that GAs are capable of removing detection-irrelevant features, outperforming the traditional approach of using the complete feature set. We have tested the proposed method on two difficult object detection problems: vehicle detection and face detection. Experimental results show that the proposed method does boost the performance of both systems using SVMs for classification. For future work, we plan to investigate the proposed feature selection scheme across different feature extraction spaces. We envision that GAs will be able to select the complementary information offered by different feature extraction methods and improve the object detection system performance further.

Acknowledgements: This research was supported by Ford Motor Company under grant No.2001332R, the University of Nevada, Reno under an Applied Research Initiative (ARI) grant, and in part by NSF under CRCD grant No.0088086.

References

[1] Z. Sun, G. Bebis, and R. Miller, "Improving the performance of on-road vehicle detection by combining gabor and wavelet features," *The IEEE Fifth International Conference on Intelligent Transportation Systems* , September, 2002, Singapore.

[2] N. Matthews, P. An, D. Charnley, and C. Harris, "Vehicle detection and recognition in greyscale imagery," *Control Engineering Practice*, vol. 4, pp. 473–479, 1996.

[3] C. Goerick, N. Detlev and M. Werner, "Artificial neural networks in real-time car detection and tracking applications," *Pattern Recognition Letters*, vol. 17, pp. 335–343, 1996.

[4] H. Schneiderman and T. Kanade, "Probabilistic modeling of local appearance and spatial relationships for object recognition," *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 45–51, 1998.

[5] M. Weber, M. Welling, and P. Perona, "Unsupervised learning of models for recognition," *European Conference on Computer Vision*, pp. 18–32, 2000.

[6] C. Papageorgiou and T. Poggio, "A trainable system for object detection," *International Journal of Computer Vision*, vol. 38, no. 1, pp. 15–33, 2000.

[7] H. Rowley, S. Baluja, and T. Kanade, "Neural network-based face fetection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 22–38, 1998.

[8] K. Sung and T. Poggio, "Example-base learning for view-based human face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 39–51, 1998.

[9] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: An application to face detection," *Proc. of Computer Vision and Pattern Recognition*, 1997.

[10] P. Viola and M. Jones, "Rapid object detection using a boosted cascacd of simple features," *Proc. Computer Vision and Pattern Recogntion*, 2001.

[11] M. Yang, D. Kriegman and N. Ahuja, "Detection faces in images:a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 34–58, 2002.

[12] E. Hjelmas and B. Low, "Face detection:a survey," *Computer Vision and Image Understanding*, vol. 83, pp. 236–274, 2001.

[13] A.Jain D. Zongker, "Feature selection: Evaluation, application, and small sample performance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 153–158, 1997.

[14] J. Yang and V. Honavar, "Feature subset selection using a genetic algorithm," *A Data Mining Perspective (H. Motoda and H. Liu eds) Chapt 8, Kluwer*, 1998.

[15] T.Marill and D.Green, "On the effectiveness of receptors in recognition systems," *IEEE Transactions on Information Theory*, vol. 9, pp. 11–17, 1963.

[16] S. Stearns, "On selecting features for pattern classifiers," *The Third International Conference of Pattern Recognition*, pp. 71–75, 1976.

[17] P.Pudil, J. Novovicova and J. Kittler, "Floating search methods in feature selection," *Pattern Recognition Letter*, vol. 15, pp. 1119–1125, 1994.

[18] K. Kira and L. Rendell, "A practical approach to feature selection," *The Ninth International Conference on Machine Learning*, pp. 249–256, 1992.

[19] L. Wiskott, J. Fellous, N. Kruger and C. Malsburg, "Estimating attributes: Analysis and extension of relief," *European Conference on Machine Learning*, pp. 171–182, 1994.

[20] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison Wesley, 1989.

[21] W. Siedlecki and J. Sklansky, "A note on genetic algorithm for large-scale feature selection," *Pattern recognition letter*, vol. 10, pp. 335–347, 1989.

[22] Y. Chtioui, D. Bertrand and D. Barba, "Feature selection by a genetic algorithm. application to seed discrimination by artificial vision," *J. Sci. Food Agric.*, vol. 76, pp. 77–86, 1998.

[23] Z. Sun, X. Yuan, G. Bebis, and S. Louis, "Neural-network-based gender classification using genetic eigen-feature extraction,," *IEEE International Joint Conference on Neural Networks*, May,2002.

[24] Z. Sun, X. Yuan, G. Bebis, and S. Louis, "Genetic feature subset selection for gender classification: A comparison study," *IEEE Workshop on Applications of Computer Vision*, December,2002.

[25] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, pp. 71–86, 1991.

[26] D. Swets, J. Weng, "Using discriminant eigenfeatures for image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 831–836, 1996.

[27] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer Verlag, 1995.

[28] G. Bebis, S. Uthiram, and M. Georgiopoulos, "Face detection and verification using genetic search," *International Journal on Artificial Intelligence Tools*, vol. 9, no. 2, pp. 225–246, 2000.