IMAGE DENOISING USING SPARSE REPRESENTATIONS

SeyyedMajid Valiollahzadeh[(a)]*, Hamed Firouzi[(a)], Massoud Babaie-Zadeh[(a)], and Christian Jutten[(b)]

[(a)]Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran,
[(b)]GIPSA-lab, Grenoble, France

Abstract. The problem of removing white zero-mean Gaussian noise from an image is an interesting inverse problem to be investigated in this paper through sparse and redundant representations. However, finding the sparsest possible solution in the noise scenario was of great debate among the researchers. In this paper we make use of new approach to solve this problem and show that it is comparable with the state-of-art denoising approaches.

1 Introduction

Being a simple inverse problem, the denoising is a challenging task and basically addresses the problem of estimating a signal from the noisy measured version available from that. A very common assumption is that the present noise is additive zero-mean white Gaussian with standard deviation σ. Many solutions have been proposed for this problem based on different ideas, such as spatial adaptive filters, diffusion enhancement [1], statistical modeling [2], transfer domain methods [3], [4], order statistics [5] and yet many more. Among these methods, methods based on with sparse and redundant representations has recently attracted lots of attentions [8]. Many researchers have reported that such representations are highly effective and promising toward this stated problem [8]. Pioneered by Donoho [5], sparse representations firstly examined with unitary wavelet dictionaries leading to the well-known shrinkage algorithm [5]. A major motivation of using overcomplete representations is mainly to obtain translation-invariant property [6]. In this respect, several multiresolutional and directional redundant transforms are introduced and applied to denoising such as curvelets, contourlets, wedgelets, bandlets and the steerable wavelet [5] [8].

The aim of all such transforms is to provide a redundant sparse decomposition of the signal. In parallel, beside providing a suitable redundant transform, representation of a signal with these transforms is also of high value, since such a representation is not necessarily unique. Several methods are then proposed to

* This work has been partially supported by Iran National Science Foundation (INSF) under contract number 86/994 and also by ISMO and French embassy in Iran in the framework of a Gundi-Shapour collaboration program.

find the best possible representation of a signal from a redundant, overcomplete dictionary obtained by these transforms, namely Matching Pursuit(MP), Basis Pursuit(BP), FOCUSS, and Smoothed ℓ^0- Norm (SL0) [7]. All these approaches basically try to find the sparsest possible solution among all the possible representations a signal can obtain. As an alternative point of view to obtain the sparse representation, example-based dictionary learning of K-SVD which is introduced by Aharon, *et. al.* [8] attempts to find the sparse dictionary over the image blocks. When using the Bayesian approach to address this inverse problem with the prior of sparsity and redundancy on the image, it is the dictionary to be used that we target as the learned set of parameters. Instead of the deployment of a pre-chosen set of basis functions like the curvelet or contourlet, this process of dictionary learning can be done through examples, a corpus of blocks taken from a high-quality set of images and even blocks from the corrupted image itself. This idea of learning a dictionary that yields sparse representations for a set of training image blocks has been studied in a sequence of works [8] and specifically the one using K-SVD has shown to outperform in both providing the sparse representation and capability of denoising. While the work reported here is based on the same idea of sparsity and redundancy concepts, a different method is used to solve the sparsest possible solution in presence of noise. An example-based dictionary learning such as K-SVD along with here used technique can provide better solutions in estimation of the original clean signal.

The paper is organized as follows. In section 2, we briefly present modeling of the scenario in decomposing a signal on an overcomplete dictionary in the presence of noise. In section 3 we discuss this algorithm in the real image denoising task. At the end we conclude and give a general overview to future's work.

2 FINDING THE SPARSE REPRESENTATION IN PRESENCE OF NOISE

Consider the problem of estimation of $\mathbf{x}$ from the observed signal

$$\mathbf{y} = \mathbf{x} + \mathbf{n}$$

where $\mathbf{n}$ denotes the observation noise. Assume that $\mathbf{x}$ has a sparse representation over the dictionary $\boldsymbol{\Phi}$, i.e. $\mathbf{x} = \boldsymbol{\Phi}\boldsymbol{\alpha}$ with a small $\|\boldsymbol{\alpha}\|_0^0$ (the number of nonzero elements of a vector) and also assume that a good estimation on the energy of the present noise, $\|\mathbf{n}\|_2^2 \leq \epsilon^2$ is provided.

The sparsest representation we are looking for, is simply

$$P_0 : \qquad \min \|\boldsymbol{\alpha}\|_0^0 \qquad \text{subject to} \qquad \|\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\alpha}\|_2^2 \leq \epsilon^2 \tag{1}$$

Note that the above-stated problem rarely has a unique solution [11], since once the sparsest solution is found, many feasible variants of it sharing the same support can be built. Since the above-stated problem is highly nonconvex and hard to deal with, many researchers pursue a strategy of convexification with

- Initialization: let $\boldsymbol{\alpha} = \lambda(\mathbf{I} + \boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\mathbf{y}$
  (This is equivalent to the solution when the σ tends to be infinity)
  i.e.:

  $$\text{argmin}_{\boldsymbol{\alpha}} \quad \|\boldsymbol{\alpha}\|_2^2 + \lambda\|\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\alpha}\|_2^2$$
- Choose a suitable decreasing sequence for σ, $[\sigma_1 \dots \sigma_J]$.
- for $n = 1, \dots, J$:
  1. Let $\sigma = \sigma_n$.
  2. find $\boldsymbol{\alpha}_\sigma^{opt} = \text{argmin}_{\boldsymbol{\alpha}} \quad (m - F_\sigma(\boldsymbol{\alpha})) + \lambda\|\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\alpha}\|_2^2$
     using any kind of optimization tool ,
     say steepest decent with fixed number of iterations
- Final answer is $\boldsymbol{\alpha} = \boldsymbol{\alpha}^{opt}$.

Fig. 1. Algorithm for finding the sparse coefficients in presence of noise.

replacing ℓ^0 norm with ℓ^1- norm. so simply try to solve the following problem instead:

$$P_1 : \qquad \min \|\boldsymbol{\alpha}\|_1 \qquad \text{subject to} \qquad \|\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\alpha}\|_2^2 \leq \epsilon^2 \tag{2}$$

where $\|\boldsymbol{\alpha}\|_1 = \sum \boldsymbol{\alpha}_i$ is the ℓ^1-norm of $\boldsymbol{\alpha}$. Note that the replacing ℓ^0-norm by other convex cost functions such as ℓ^1-norm is only asymptotic and the equivalence does not always hold [9]. Hereafter, motivated by the recently stated work of Mohimani, *et. al.* [7] we seek to find the sparsest possible answer without such a replacement and instead, attempt to relax the replacing ℓ^0- norm by a continuous, differentiable cost function, say $F_\sigma(\boldsymbol{\alpha}) = \sum_i \exp(-\alpha_i^2/2\sigma^2)$.

This function tends to count the number of zero elements of a vector. So, as stated in [7] the above problem can be converted to:

$$P_0 : \qquad \min_{\boldsymbol{\alpha}}(m - F_\sigma(\boldsymbol{\alpha})) \qquad \text{subject to} \qquad \|\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\alpha}\|_2^2 \leq \epsilon^2 \tag{3}$$

The above optimization task can be converted to optimizing the Lagrangian:

$$P_0 : \qquad \min_{\boldsymbol{\alpha}}(m - F_\sigma(\boldsymbol{\alpha})) + \lambda\|\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\alpha}\|_2^2 \tag{4}$$

So that the constraint becomes a penalty and the parameter λ is dependent on ϵ. Solution toward this problem was recently proposed in [12] and it is shown that for a proper choice of λ, these two problems are equivalent. The σ parameter determines the smoothness of the approximated cost function. By gradual decrease in this parameter it is highly probable to skip trapping in local minimum. The overall algorithm which is used through this paper is shown in Fig. 1 is a slight modification of the same idea presented in [12].

Once the sparsest solution of (3) has been found with the stated algorithm summarized in Fig. 1, we can retrieve the approximate image by $\hat{\mathbf{x}} = \boldsymbol{\Phi}\hat{\boldsymbol{\alpha}}$.

3 IMAGE DENOISING

The problem of estimation of X from an observed noisy version of it under the sparsity prior has two essential issues: first, to find a dictionary $\boldsymbol{\Phi}$ which permits a sparse representation regarding the fact that the sample are noisy and second to find the coefficients of this sparse representation. The second phase was what explained so far. As it was shown by Aharon [8], *et. al.*, the K-SVD learning is a very efficient strategy which leads to satisfactory results. This method along with all other types of dictionary learning fails to act properly [8] when the size of dictionary grows. Beside that, the computational complexity and thus time needed for training will grow awesome.

When we are dealing with larger size images we are still eager to apply this method but as stated it is computationally costly and both dictionary learning and optimization to find the coefficients of sparse representation are sometimes intractable. To overcome this difficulty, an image with size $\sqrt{N} \times \sqrt{N}$ is divided to blocks of size of $\sqrt{n} \times \sqrt{n}$. These blocks are chosen highly overlapped for two reasons: first, to avoid blockiness and second to have better estimate in noise removal process. Then a dictionary is tried to be found over these blocks and all these blocks are cleaned with algorithm of Fig. 1. Let $\mathbf{L}_{ij}$ be a matrix representing each block to be located in (ij)-th position of the image. $\mathbf{L}_{ij}$ is a matrix of size $n \times N$ which provides the location information of all possible blocks of the images. So in this respect, the noise removal process changes to:

$$\{\hat{\mathbf{X}}, \hat{\boldsymbol{\alpha}}\} = \text{argmin}_{\mathbf{X},\boldsymbol{\alpha}} \lambda \|\mathbf{Y} - \mathbf{X}\|_2^2 + \sum_{ij} \gamma \|\boldsymbol{\alpha}_{ij}\|_0^0 + \sum_{ij} \|\boldsymbol{\Phi}\boldsymbol{\alpha} - \mathbf{L}_{ij}\mathbf{X}\|_2^2 \qquad (5)$$

in which $\mathbf{X}$ is the original image to be estimated and the $\mathbf{Y}$ is the observed available noisy version of it. This equation is similar to (1) with this slight difference that local analysis was taken into account and a linear combination of ℓ^0-norm and ℓ^2-norm of all sparse representation and error between the original signal and the reconstructed one tried to be minimized. In this process, visible artifacts may occur due to blocking phenomena. To avoid this, we choose the blocks with overlap and at the end average the results in order to prevent blockiness artifact. After determining all the approximated coefficients, we estimate the original image by solving the following equation:

$$\hat{\mathbf{X}} = \text{argmin}_{\mathbf{X}} \lambda \|\mathbf{Y} - \mathbf{X}\|_2^2 + \sum_{ij} \|\boldsymbol{\Phi}\boldsymbol{\alpha} - \mathbf{L}_{ij}\mathbf{X}\|_2^2 \qquad (6)$$

This quadratic equation has the solution:

$$\hat{\mathbf{X}} = (\lambda \mathbf{I} + \sum_{ij} \mathbf{L}_{ij}^T \mathbf{L}_{ij})^{-1} (\lambda \mathbf{Y} + \sum_{ij} \mathbf{L}_{ij}^T \boldsymbol{\Phi}\hat{\boldsymbol{\alpha}})^{-1} \qquad (7)$$

This estimated modified image can be interpreted as a relaxed averaging between the noisy observed image with the cleaned estimated one. The summarized overall algorithm is shown is Fig. 2.

- Goal: denoise a given image $\mathbf{Y}$ from additive white Gaussian noise with variance of $\|\mathbf{n}\|_2^2$
- parameters:
  n block-size ,k dictionary, λ Lagrangian multiplier.the task is to optimize

  $$\{\hat{\mathbf{X}}, \hat{\boldsymbol{\alpha}}\} = \text{argmin}_{\mathbf{X},\boldsymbol{\alpha}} \lambda \|\mathbf{Y} - \mathbf{X}\|_2^2 + \sum_{ij} \gamma \|\boldsymbol{\alpha}_{ij}\|_0^0 + \sum_{ij} \|\boldsymbol{\Phi}\boldsymbol{\alpha} - \mathbf{L}_{ij}\mathbf{X}\|_2^2$$

- train a dictionary $\boldsymbol{\Phi}$ of size $n \times k$ using K-SVD.
- find the sparse noisy coefficients of $\boldsymbol{\alpha}$ using algorithm stated in Fig. 1.
- Final estimation is $\hat{\mathbf{X}} = (\lambda \mathbf{I} + \sum_{ij} \mathbf{L}_{ij}^T \mathbf{L}_{ij})^{-1} (\lambda \mathbf{Y} + \sum_{ij} \mathbf{L}_{ij}^T \boldsymbol{\Phi}\hat{\boldsymbol{\alpha}})^{-1}$.

Fig. 2. The final denoising algorithm.

4 Experimental results:

In this work, the underlying dictionary was trained with the K-SVD method and once the learning is done, the image blocks was represented sparsely via Fig. 1. The algorithm of section 2 was used for such a representation. The overall denoising method explained above was examined with numerous test images mainly of size 256×256 and 512×512 with different noise levels. Blocks of size 8×8 was driven by the synthesis noisy image and a dictionary of size 64×256 was learned through this blocks using K-SVD method. Then we applied the algorithm of Fig. 1 to represent each block on the provided dictionary, while the similar approach done by Aharon [8] make use of Orthogonal Matching Pursuit (OMP) [10] for this stage. The tested images are all the same ones as those used in the denoising experiments reported in [8], in order to enable a fair comparison. Table 1 summarizes the denoising results in the same database of images. In a quite large experiments we found sparser solution and better quality of representations. Every result reported is an average over 5 experiments, having different realizations of the noise. To show a comparison in sparsity yielded with different methods coefficients in representations of a sample block with OMP and the stated algorithm was depicted in Fig. 3. The quite same results is valid for other blocks as well.

The denoised blocks were averaged, as described in Fig.2 .In Fig. 5 the results of the overall algorithm for the image "Barbara" for $\|\mathbf{n}\|_2 = 20$ is shown. By refereing to Table 1, as it is seen, when the level of noise grows, our approach outperforms K-SVD with OMP and we can conclude the mentioned algorithm is suitably designed for noisy cased with known energy.

Also a comparison was done with other types of sparse coding phase such as FOCUSS and SL0 [8] and yet the proposed algorithm outperforms them. A sample comparison has been done in Fig. 4. In this experiment after providing the dictionary, the sparse representation coefficients are found with different approaches. The coefficients of the original clean signal, the signal corrupted

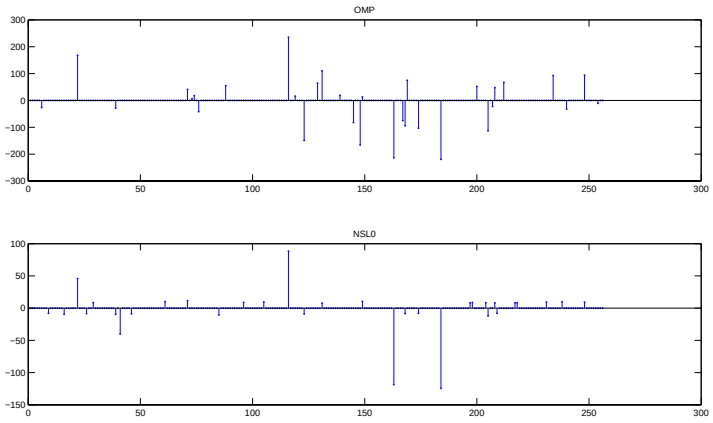


Fig. 3. Coefficients of a sample block represented with OMP above and in bottom .the latter, leads to the same result or sparsely superior one.

with additive white gaussian noise of $v\|\mathbf{n}\|_2 = 20$, recovered block via OMP and the recovered signal via Fig. 1 is depicted in Fig. 4 and as it can be seen the our recovered signal resembles more to the original signal.

5 Discussions and Conclusions

In this paper a simple algorithm for denoising application of an image was presented leading to state-of-the-art performance, equivalent to and sometimes surpassing recently published leading alternatives. It is basically on the basis of sparse representation of an image in the presence of noise. The stated algorithm considers local approach, splits the noisy observed image to several blocks and learns a dictionary over these blocks and attempts to find the best possible sparse representation of each block with this dictionary. In order to find the cleaned image some averaging is needed to avoid the blocking effect in boundaries. Experimental results show satisfactory recovering of the image. Future theoretical work on the general behavior of this algorithm is on our further research agenda.

References

1. Gilboa, G. , Sochen, N. and Zeevi, Y.: Forward-and-Backward Diffusion Processes for Adaptive Image Enhancement and Denoising. IEEE Trans on Image Processing. IEEE Tran on Image Processing, Vol. 11, No. 7, pp 689-703,(JULY 2002)
2. Mihcak, M. K. , Kozintsev, I. , Ramchandran K. and Moulin, P.: Low complexity image denoising based on statistical modeling of wavelet coefficients. IEEE Trans on Image Processing. IEEE Signal Processing Letters,Vol 6, No 12 pp 300-303 (Dec 1999)

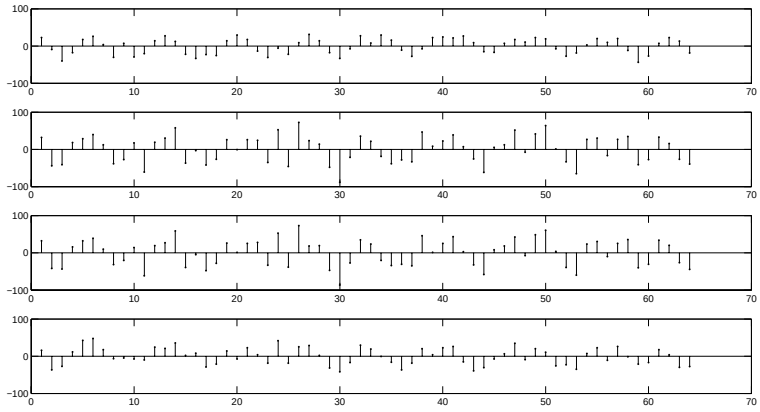


Fig. 4. Coefficients of a sample block. From top to bottom: the original clean signal, the signal corrupted with additive white Gaussian noise of $\|\mathbf{n}\|_2 = 20$, recovered block via OMP and the recovered block with the algorithm of Fig. 1.

3. Grace Chang, S. , Yu, B. and Vetterli, M. : Adaptive wavelet thresholding for image denoising and compression. IEEE Trans on Image Processing. IEEE Tran on Image Processing, Vol 9, No 9, pp 1532-1546, (Sep 2000)
4. Starck, J. , Candes, E. and Donoho, DL: The curvelet transform for image denoising. IEEE Trans on Image Processing, Vol 11, Issue 6, pp 670 -684, 2002
5. Donoho, D. L.: De-noising by soft thresholding. IEEE Trans. Inf. Theory, vol. 41, no. 3, pp. 613-627, May 1995.
6. Coifman, R. and Donoho, D. L.: Translation invariant de-noising. in Wavelets and Statistics, Lecture Notes in Statistics. New York: Springer- pp. 125-150, 1995
7. Mohimani, H. , Babaei-Zadeh, M. and Jutten, C.: A Fast approach for overcomplete sparse decomposition based on smoothed L0 norm. ,to appear in IEEE Trans on Signal Processing
8. Aharon, M. , Elad, M. and Bruckstein, A.: K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. IEEE Trans on Signal Processing, vol. 54, No. 11, Nov 2006
9. Candes, E. J. , Wakin, M. B., and Boyd, S.: Enhancing Sparsity by Reweighted l1 Minimization. Journal of Fourier Analysis and Applications, special issue on sparsity, October 2008
10. Davis, G. ,Mallat, S. and Zhang, Z.:Adaptive time-frequency approximations with matching pursuits, Tech. Rep. TR1994-657, 1994.
11. Donoho, D.L.: Elad, M.; Temlyakov, V.N.:Stable recovery of sparse overcomplete representations in the presence of noise , Information Theory, IEEE Transactions on Volume 52, Issue 1, pp 6-18, Jan. 2006 .
12. Firouzi, H. ,Farivar, M. ,Babaie-Zadeh, M. and Jutten, C.: Approximate Sparse decomposition Based on Smoothed L-0 norm submitted to ICASSP2009 available in http://www.arxiv.org.

Fig. 5. From left to right: original image, noisy image with zero-mean white gaussian noise of $\|\mathbf{n}\|_2 = 20$, the cleaned image via sparse representation described.

Table 1. Summary of denoising PSNR results. In each column the bottom is corresponding to our approach and the above is corresponding to the K-SVD with OMP. the bold one corresponds with better response.

$\frac{\sigma}{PSNR}$	LENA	BARBARA	BOAT	Fgrpt	House	Peppers	Average	σ_{PSNR}
2/42.11	**43.58**	**43.67**	**43.14**	**42.99**	**44.47**	**43.33**	**44.47**	**43.33**
	42.11	42.38	42.17	41.85	42.92	42.51	42.92	42.51
5/34.15	**38.60**	**38.08**	**37.22**	**36.65**	**39.37**	**37.78**	**39.37**	**37.78**
	38.18	37.41	36.68	36.17	38.25	37.08	38.25	37.08
10/28.13	**35.47**	34.42	**33.64**	**32.39**	**35.98**	34.28	35.98	**34.28**
	35.42	**34.51**	33.62	32.31	35.60	**34.53**	35.60	**34.53**
15/24.61	33.70	32.36	31.73	30.06	34.32	32.22	34.32	32.22
	33.91	**32.79**	**32.13**	**30.258**	**34.40**	**32.79**	**34.40**	**32.79**
20/22.11	32.38	30.83	30.36	28.47	33.20	30.82	33.20	30.82
	33.46	**32.01**	**31.29**	**29.16**	**34.19**	**31.58**	**34.19**	**31.58**
25/20.17	31.32	29.60	29.28	27.26	32.15	29.73	32.15	29.73
	32.72	**31.01**	**30.46**	**28.90**	**33.61**	**30.83**	**33.61**	**30.83**
50/14.15	27.79	25.47	25.95	23.24	27.95	26.13	27.95	26.13
	28.98	**26.93**	**27.30**	**24.43**	**28.69**	**27.70**	**28.69**	**27.70**
75/10.63	25.80	23.01	23.98	19.97	25.22	23.69	25.22	23.69
	26.93	**24.71**	**25.33**	**21.69**	**26.83**	**24.28**	**26.83**	**24.28**
75/10.63	24.46	21.89	22.81	18.30	23.71	21.75	23.71	21.75
	26.32	**23.55**	**24.36**	**22.19**	**25.08**	**23.14**	**25.08**	**23.14**

A NEW TREND IN OPTIMIZATION ON MULTI OVERCOMPLETE DICTIONARY TOWARD INPAINTING

SeyyedMajid Valiollahzadeh[(a)], *Mohammad Nazari*[(b)], *Massoud Babaie-Zadeh* [(a)], *Christian Jutten*[(c)]

[(a)]Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran
[(b)]Department of Electrical Engineering, Amirkabir University of Technology, Tehran, Iran
[(c)]Laboratoire des Images et des Signuax, Institute National Polytechnique de Grenoble, France

ABSTRACT

Recently, great attention was intended toward overcomplete dictionaries and the sparse representations they can provide. In a wide variety of signal processing problems, sparsity serves a crucial property leading to high performance. Inpainting, the process of reconstructing lost or deteriorated parts of images or videos, is an interesting application which can be handled by suitably decomposition of an image through combination of overcomplete dictionaries. This paper addresses a novel technique of such a decomposition and investigate that through inpainting of images. Simulations are presented to demonstrate the validation of our approach.

Index Terms— Sparse representations, Inpainting, Texture, Cartoon, Total variation.

1. INTRODUCTION

Sparse signal decomposition of signals on an over overcomplete Dictionary was of great interest among researchers in past few years and serve many interesting applications [1]. The main assumption over these signals is that they are linear mixtures of building atoms and also only a few of these atoms will participate in the reconstruction. In the context of image processing an interesting decomposition application would be separating texture from non-texture part to be used in areas from compression to analysis and synthesis of an image[2][3].

Inpainting consists in problems like filling the holes, reconstructing lost or deteriorated parts of images or videos, removal of scratches in old photos, removal of unwanted text or graphic and is an interesting inverse problem with lots of research momentum [4] in recent years dealing highly with such decomposition. Pioneered by the work of Sapiro et al [5], total variation was used in this respect taking mainly the geometrical contents into consideration. Since images contain both geometrical and textural information, decomposition should be done in two layers.

This approach has been presented in [6]-[7] and the layers to which an image is decomposed are called texture and cartoon. The inpainting process is done in each layer separately and afterwards the output will be formed by summing up these layers. The crucial part in this approach is layer decomposition and will extend the notions of total variations. By this trend, if any failure in the inpainting of each layer is presented, superimposing of two layers will lead in less visual artifact and hence quite satisfactory result.

In some recent work sparsity was taken into account as additional criteria to decompose an image to these layers. To this end, we need two dictionaries, mutually incoherent, one to represent the texture and the other for the cartoon. Both should provide the sparse representation for the corresponding layer image while yielding nonsparse for the other. Combination of these two dictionaries into one and performing the (Basis Pursuit denoising) BPDN [1] algorithm seeking the sparsest solution has shown to perform well and even can be improved by further applying the total-variation regularization.

Elad *et al*. [8, 9] proposed an inpainting algorithm capable of filling in holes in either texture or cartoon content, or any combination thereof extending employment of separation by sparsity, so that the missing samples fit naturally into the layer separation framework. The main advantageous point of this approach is the global treatment trend toward the image rather the local one. Also it deploys general overcomplete dictionaries which can be better established for a typical image content.

What is presented in this paper is quite similar on the basis of sparse representations, but modeling the overall problem as a specific optimization is better relaxed. Inspired by the work of Mohimani, *et al.* [10] for finding

[1] This work has been partially supported by Iran National Science Foundation (INSF) under contract number 86/994, by Iran Telecommunications Research Center (ITRC), and also by ISMO and French embassy in Iran in the framework of a Gundi-Shapour collaboration program

the sparsest solution of an Underdetermined System of Linear Equations (USLE) through the smoothed ℓ^0-norm, we extend this approach in two dimensional models to solve the prior modeling. The outline of the paper is as follows. In section 2, we briefly present the modeling scenario to decompose a signal over two incoherent dictionaries. In section 3 we model the inpainting problem and present the final algorithm. We discuss some simulation results to validate the proposed algorithm in section 4 and finally conclusion and summary of later work is discussed in the last section.

2. MAIN IDEA

Let the input image **c** containing *N* total pixels, be presented as a one-dimensional vector. This image is to be decomposed over two distinct dictionaries, **A** and **B,** the former corresponding to texture and the latter to cartoon. Both provide sparse representation for the image of their kind and non-sparse for the other, written formally as:

$$\mathbf{c}_1 = \mathbf{A}\,\mathbf{s}_1 \qquad (\mathbf{s}_1 \text{ is sparse}) \tag{1}$$

$$\mathbf{c}_2 = \mathbf{B}\,\mathbf{s}_2 \qquad (\mathbf{s}_2 \text{ is sparse}) \tag{2}$$

Sparsity of a vector S is quantified by its ℓ^0-norm, denoted by $\|\mathbf{s}\|_0$, defined by the number of its nonzero elements. There are two assumptions over these dictionaries [8,9]: firstly, these two dictionaries should be incoherent, i.e. the texture dictionary is not able to represent the cartoon image sparsely and vice versa. Secondly, the dictionary assigned to texture should be such that if the texture appears in parts of the image and is otherwise zero, representation is still sparse, implying somehow that it should employ a multiscale and local analysis of the image content.

Now, we seek a sparse representation over the combined dictionary:

$$\{\mathbf{s}_1,\mathbf{s}_2\} = \underset{\mathbf{s}_1,\mathbf{s}_2}{\operatorname{argmin}} \left\{ \|\mathbf{s}_1\|_0 + \|\mathbf{s}_2\|_0 \right\} \quad \text{Subject to:} \quad \mathbf{A}\mathbf{s}_1 + \mathbf{B}\mathbf{s}_2 = \mathbf{c} \tag{3}$$

The problem is non-convex and seemingly intractable due to combinatorial search it needs, however inspired by the work of Mohimani et al [10], we can find S_1, S_2 as it using smoothed ℓ^0-norm. Smoothed ℓ^0-norm of a vector α is an approximation to its ℓ^0-norm and is defined as:

$$F_\sigma(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \exp(-\alpha_i^2 / 2\sigma^2) \tag{4}$$

where α is a parameter determining a tradeoff between the accuracy of approximation and the smoothness of $F_\sigma(\boldsymbol{\alpha})$. Minimizing the ℓ^0 norm of $\boldsymbol{\alpha}$ subject to $\mathbf{b} = \boldsymbol{\Phi}\boldsymbol{\alpha}$ then requires then to maximize $F_\sigma(\boldsymbol{\alpha})$ for a small value of σ. For a small σ, $F_\sigma(\boldsymbol{\alpha})$ is highly non-smooth with lots of local maxima. To overcome this difficulty we use a decreasing sequence of σ and make use of maximizer of $F_\sigma(\boldsymbol{\alpha})$ as a starting point to find the next (smaller) sigma [10]. Moreover, the algorithm initially starts with minimum ℓ^2 norm solution of $\mathbf{b} = \boldsymbol{\Phi}\boldsymbol{\alpha}$, which corresponds to the maximizer of $F_\sigma(\boldsymbol{\alpha})$ when $\sigma \to \infty$.

Using similar idea ,we want to minimize a cost function $J_\sigma(\mathbf{s})$ -which will be introduced in the next section- subject to $\mathbf{A}\mathbf{s}_1 + \mathbf{B}\mathbf{s}_2 = \mathbf{c}$.The minimization should be done for small σ and in order to avoid trapping in local minima we use a sequence of $\left[\sigma_1,...,\sigma_{k_{\max}}\right]$ and then minimize $J_\sigma(\mathbf{s})$ for each σ, with the starting point yielded by the maximizer of the previous (longer) σ.
Moreover the process is initialized by:

$$\begin{bmatrix}\mathbf{A} & \mathbf{B}\end{bmatrix}\begin{bmatrix}\mathbf{s}_1\\ \mathbf{s}_2\end{bmatrix} = \mathbf{c} \Rightarrow \quad \begin{bmatrix}\mathbf{s}_1\\ \mathbf{s}_2\end{bmatrix} = \begin{bmatrix}\mathbf{A} & \mathbf{B}\end{bmatrix}^{\dagger}\mathbf{c} = \begin{bmatrix}\left(\mathbf{P}_{\mathbf{B}}^{\perp}\mathbf{A}\right)^{\dagger}\\ \left(\mathbf{P}_{\mathbf{A}}^{\perp}\mathbf{B}\right)^{\dagger}\end{bmatrix}\mathbf{c} \tag{5}$$

where, $\mathbf{P}_{\mathbf{A}}^{\perp}$ and $\mathbf{P}_{\mathbf{B}}^{\perp}$ are the orthogonal projections of the corresponding matrices:

$$\mathbf{P}_{\mathbf{A}}^{\perp} = \mathbf{I} - \mathbf{A}^T\left(\mathbf{A}\mathbf{A}^T\right)^{-1}\mathbf{A} \qquad \mathbf{P}_{\mathbf{B}}^{\perp} = \mathbf{I} - \mathbf{B}^T\left(\mathbf{B}\mathbf{B}^T\right)^{-1}\mathbf{B} \tag{6}$$

Then we use L iterations of the steepest ascent algorithm, followed by a projection onto the feasible which is:

$$\textit{Update}\ \begin{bmatrix}\mathbf{s}_1\\ \mathbf{s}_2\end{bmatrix} = \begin{bmatrix}\mathbf{s}_1\\ \mathbf{s}_2\end{bmatrix} - \begin{bmatrix}\mathbf{A} & \mathbf{B}\end{bmatrix}^{\dagger}(\mathbf{c} - \mathbf{A}\mathbf{s}_1 - \mathbf{B}\mathbf{s}_2) \tag{8}$$

For more explanation about choosing the sequence see [10].

3. Modeling inpainting and the final proposed algorithm

Suppose that missing pixels of the image are masked with a diagonal mask matrix **M** (of which has value'1' over the existing pixels and '0' over the missing pixels) we propose restoring the image by optimizing the following problem:

$$\{\mathbf{s}_1^{Opt},\mathbf{s}_2^{Opt}\} = \arg\min_{\mathbf{s}_1,\mathbf{s}_2}\left\{\|\mathbf{s}_1\|_0 + \|\mathbf{s}_2\|_0 + \lambda\left\|\mathbf{M}\left(\mathbf{c} - \mathbf{A}\mathbf{s}_1 - \mathbf{B}\mathbf{s}_2\right)\right\|_2^2 + \gamma\,\mathrm{TV}\{\mathbf{A}\mathbf{s}_1\}\right\} \tag{9}$$

in which we have $\mathrm{TV}\{\mathbf{x}\} = \|\nabla \mathbf{x}\|_1$. So the recovered image would be:

$$\hat{\mathbf{c}} = \mathbf{A}\mathbf{s}_1^{Opt} + \mathbf{B}\mathbf{s}_2^{Opt} \tag{10}$$

– *Initialization* :

+*Let* $\begin{bmatrix}\mathbf{s}_1\\ \mathbf{s}_2\end{bmatrix} = \begin{bmatrix}\mathbf{A} & \mathbf{B}\end{bmatrix}^{\dagger}\mathbf{c}$

+*Choose a suitable sequence for* $\sigma = \left[\sigma_1, ..., \sigma_{N_{\max}}\right]$

–*For* $n = 1, ..., N$

+*Maximize the function* $J_{\sigma_n}(\mathrm{S})$ *using steepest descent algorithm*

–*For* $k = 1, ..., L$

- $\begin{bmatrix}\Delta\mathbf{s}_1\\ \Delta\mathbf{s}_2\end{bmatrix} =$

$$\begin{bmatrix} s_{11}^{(n)} \exp\left(-\left(s_{11}\right)^2 \Big/ 2\sigma_n^2\right) & \cdots & s_{1m_1}^{(n)} \exp\left(-\left(s_{1m_1}\right)^2 \Big/ 2\sigma_n^2\right) \\ s_{21}^{(n)} \exp\left(-\left(s_{21}\right)^2 \Big/ 2\sigma_n^2\right) & \cdots & s_{2m_2}^{(n)} \exp\left(-\left(s_{2m_2}\right)^2 \Big/ 2\sigma_n^2\right) \end{bmatrix}$$

$$+ 2\lambda \begin{bmatrix}\mathbf{A}^T\left(\mathbf{c} - \mathbf{A}\mathbf{s}_1\right)\\ \mathbf{B}^T\left(\mathbf{c} - \mathbf{B}\mathbf{s}_2\right)\end{bmatrix}.$$

- $\begin{bmatrix}\mathbf{s}_1\\ \mathbf{s}_2\end{bmatrix} = \begin{bmatrix}\mathbf{s}_1\\ \mathbf{s}_2\end{bmatrix} - \begin{bmatrix}\mu_1\Delta\mathbf{s}_1\\ \mu_2\Delta\mathbf{s}_2\end{bmatrix}$

$\left(\begin{array}{l}\mu_1 \text{ and } \mu_2 \text{ can be chosen by a line-search minimizing}\\ \text{the overal penalty function or fixed stepsize.}\end{array}\right)$

- *Calculate* $\mathbf{c}_1 = \mathbf{A}\mathbf{s}_1$
- *Apply the TV Corrolation on the* $\mathbf{c}_1$

–*Reconstruct* $\mathbf{c}_1 = \mathbf{c}_1 - \mu\dfrac{\partial \mathrm{TV}\{\mathbf{c}_1\}}{\partial \mathbf{c}_1}$

$$= \mathbf{c}_1 - \mu\nabla\left(\frac{\nabla\mathbf{c}_1}{|\nabla\mathbf{c}_1|}\right)$$

- *Update* $\lambda = \lambda - \dfrac{1}{N}\lambda_{\max}$
- *Update* $\begin{bmatrix}\mathbf{s}_1\\ \mathbf{s}_2\end{bmatrix} = \begin{bmatrix}\mathbf{s}_1\\ \mathbf{s}_2\end{bmatrix} - \begin{bmatrix}\mathbf{A} & \mathbf{B}\end{bmatrix}^{\dagger}\left(\mathbf{c} - \mathbf{A}\mathbf{s}_1 - \mathbf{B}\mathbf{s}_2\right)$

+$s = s_{n-1}$

–*Final coefficients are* $\mathbf{s}_1$ *and* $\mathbf{s}_2$.

Figure 1: The proposed algorithm for decomposition

The term $\mathrm{TV}\{\mathbf{A}\mathbf{s}_1\}$ essentially computes the image $\mathbf{A}\mathbf{s}_1$ (supposed to be piecewise smooth), applies the absolute gradient field and summing up ℓ^1 norm to avoid blockiness and force the image be smooth thus support the separation process.

These coefficients to be found can be relaxed as stated in the previous part:

$$J_\sigma = \left(M_1 - F_\sigma\left(\mathbf{s}_1\right)\right) + \left(M_2 - F_\sigma\left(\mathbf{s}_2\right)\right) + \lambda\left\|\mathbf{M}\left(\mathbf{c} - \mathbf{A}\mathbf{s}_1 - \mathbf{B}\mathbf{s}_2\right)\right\|_2^2 + \gamma\,\mathrm{TV}\left\{\mathbf{A}\mathbf{s}_1\right\} \tag{11}$$

M_1 and M_2 are the length of $\mathbf{s}_1, \mathbf{s}_2$ coefficients ,not necessarily equivalent. The overall algorithm is shown in Fig 1. The parameters γ and λ are found experimentally [9].

4. EXPERIMENTAL RESULTS

In this section, we apply the algorithm of Fig1 for the reconstruction of gray level still images where some parts are missing. In proposed algorithm, we briefly present the scenario to decompose a signal over two incoherent dictionaries. Our approach in this work is to choose two known transforms, one to represent the texture and the other for the cartoon.

With regards to the actual choice, for the cartoon representation, we used curvelet transform and for the texture; we used local-DCT transform. These dictionaries are nice choice of transform according to our experience dependent on this problem. We must remind that type of sparse transformation may vary from one image to another [8] but must be mutually independent.

In fig 2, we show the representation result of the proposed algorithm for the Barbara image. Left image was obtained using the curvelet transform with six resolution levels and right one is the output of local-DCT representation with a block size 32×32. We must mention that resolution levels in curvelet and optimal block size in local-DCT transformation were obtained experimentally.

Figure 2: The representation result in last iteration of proposed algorithm for the Barbara image.(left) Output of curvelet transform with six resolution levels. (right) Output of local-DCT representation with a block size 32×32.

The parameters we had used in our simulations are: $N = 5$ (number of decreasing value of σ), $\lambda_{\max} \in [1, 2]$ and $L = 10$ (number of iterations of the steepest ascent algorithm). Note that for calculating the computational complexity of the proposed inpainting algorithm, we can ignore L iterations of the steepest ascent calculation, therefore it is governed by the number of applying the two forward and the inverse transforms.

In fig 2, (top left) we show the original Barbara image; on top right the target regions are masked in white. Region filling via our inpainting method using curvelet and local-DCT dictionaries are illustrated on bottom left. The result of our algorithm around Barbara's eyes shows no trace of the original holes, and seems natural on bottom right.

Figure 2: The reconstruction of the masked image.
(top left) Original image. (top right) The target regions are masked in white. (bottom left) Region filling via the proposed inpainting algorithm. (bottom right) The result of our algorithm around Barbara's eyes.

5. CONCLUSIONS

In this paper we presented a novel approach for inpainting. It is basically on the basis of decomposition of an image to texture and cartoon layers via sparse combinations of atoms of predetermined dictionaries. The stated algorithm with consideration of total-variation regularization attempts to fill in the holes in each layer separately and superimposes these layers as a final solution. Experimental results show the efficiency of the proposed algorithm in finding the missing samples. Future theoretical work on the general behaviour of this algorithm along with learning of dictionaries through examples adapted to each layers are two further topics in our current research agenda.

6. REFERENCES

[1] S.S. Chen, D.L. Donoho, M.A. Saunder, “Atomic decomposition by basis pursuit”, SIAM J. Sci. Comput. 20, pp. 33–61, 1998.

[2] J.S. De Bonet, “Multiresolution sampling procedure for analysis and synthesis of texture images”, in: Proceedings of SIGGRAPH, 1997.

[3] A.A. Efros, T.K. Leung, “Texture synthesis by non-parametric sampling”, in: IEEE International Conference on Computer Vision, Corfu, Greece, pp. 1033–1038, September 1999.

[4] V. Caselles, M. Bertalmio, G. Sapiro, C. Ballester, “Image inpainting, in: Comput. Graph. (SIGGRAPH 2000), pp. 417–424, July 2000.

[5] M. Bertalmio, L. Vese, G. Sapiro, S. Osher, “Simultaneous structure and texture image inpainting”, IEEE Trans. Image Process. 12, pp. 882–889, 2003.

[6] L. Vese and S. Osher, “Modeling textures with total variation minimization and oscillating patterns in image processing,” Journal of Scientific Computing vol. 19, pp. 553–577, 2003.

[7] J. Aujol, G. Aubert, L. Blanc-Feraud, and A. Chambolle, “Image decomposition: Application to textured images and SAR images,” Tech. Rep. ISRN I3S/RR-2003-01-FR, INRIA - Project ARIANA, Sophia Antipolis, 2003.

[8] J.-L. Starck, M. Elad, D.L. Donoho, “Image decomposition via the combination of sparse representations and a variational approach”, IEEE Trans. Image Process. Vol 14,No10, pp. 1570-1582, oct. 2005.

[9] M. Elad,J.-L. Starck , P. Querre, D.L. Donoho, "Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA)", *Journal on Applied and Computational Harmonic Analysis*, Vol. 19, pp. 340-358, November 2005.

[10] G. H. Mohimani, M.Babaie-Zadeh, C. Jutten, “A Fast approach for overcomplete sparse decomposition based on smoothed L0 norm”, to appear in IEEE Trans on Signal Processing. Available at: http://ee.sharif.edu/~SLzero/

16th Iranian Conference on Electrical Engineering
ICEE 2008
Tarbiat Modares University
May 13-15, 2008

Sparse Component Analysis (SCA) in Random-valued and Salt and Pepper Noise Removal

Hadi. Zayyani, Seyyedmajid. Valliollahzadeh
Sharif University of Technology
zayyani2000@yahoo.com, valliollahzadeh@yahoo.com

Massoud. Babaie-Zadeh
Sharif University of Technology
mbzadeh@yahoo.com

Abstract: *In this paper, we propose a new method for impulse noise removal from images. It uses the sparsity of images in the Discrete Cosine Transform (DCT) domain. The zeros in this domain give us the exact mathematical equation to reconstruct the pixels that are corrupted by random-value impulse noises. The proposed method can also detect and correct the corrupted pixels. Moreover, in a simpler case that salt and pepper noise is the brightest and darkest pixels in the image, we propose a simpler version of our method. In addition to the proposed method, we suggest a combination of the traditional median filter method with our method to yield better results when the percentage of the corrupted samples is high.*

Keywords: Image denoising, salt and pepper noise, sparse component analysis, median filter.

1. Introduction

Impulse noise is caused by malfunctioning pixels in camera sensors, faulty memory locations in hardware or transmission in a noisy channel. The salt and pepper noise and the random valued-noise are the two common types of impulsive noises. In the salt and pepper noise, the salt noise is assumed to have the brightest gray level and the pepper noise has the darkest value of the gray level in the image. This assumption can help us to know the corrupted pixels in the images. In these cases the only hard task is to recover the original pixel of the image. But, in the general case of random-valued impulse noise, there is not any pre-assumption about the value of the impulsive noise. Therefore, the image denoising task in these cases is to detect the corrupted pixels and then correct them by the original pixel of the image. So, image denoising for random-valued impulse noises is more difficult than fixed salt and pepper image denoising. In this paper, we focus on the random value impulsive noise. However, we also present a version of our method in the case of salt and pepper noise.

The median filter is the most popular nonlinear filter for removing impulse noise [1]. However, when the noise level is high or when the random noise is available, some details and edges are smeared by the filter and the performance of the median filter decreases. Different remedies of the median filter have been proposed so far. They are the adaptive median filter [2], the median filter based on homogeneity [3], centre-weighted median filters [4] a generally family called decision-based methods. The so-called "decision-based" methods first identify possible noisy pixels and then replace them by using the median filter or its variants, while leaving all other pixels unchanged. Some of these two-stage methods deal with salt and pepper noise [5] and the others with the case of random-valued impulse noises [6].

In this paper, we do not separate the detection and correction steps similar to "decision-based" methods mentioned earlier. We use the compressibility of the images in the DCT domain which is used for image compression in JPEG standard. This compressibility gives us the necessary equation to exactly recover the impulsive noises or errors. Therefore, we use the transformed image to recover the noisy pixels. To recover the noisy pixels (or finding errors), we encounter an Underdetermined System of Linear Equations (USLE) whose sparse solution is to be found. This USLE problem can be solved by means of Sparse Component Analysis (SCA)

methods [7]. In the SCA context, m sparse sources (which the most of their samples are nearly zero) and n linear observations of them are available. The goal is to find these sparse sources from the observations. The relation between the sources and the observations are:

$$\mathbf{x} = \mathbf{As} \tag{1}$$

where $\mathbf{x}$ is the $n \times 1$ observation vector and $\mathbf{s}$ is the $m \times 1$ source vector and $\mathbf{A}$ is the $n \times m$ mixing matrix. m is the number of sources and n is the number of observations. In SCA, it is assumed that the number of sources is greater than the number of observations ($m > n$). So, the number of unknowns is larger than the equations. Therefore, this Underdetermined Linear System of Equations (ULSE) has infinite number of solutions. Fortunately, under conditions stated in [10], the sparsest solution of this problem is unique. This condition is that the number of active sources (non zero source) should be less than half of the number of observations ($\|\mathbf{s}\|_0 < 0.5n$). By this assumption, the sparsest solution is unique and different algorithms to find this solution have been already proposed, including Basis-Pursuit (BP) [9], FOCUSS [10], smoothed-l^0 [11] and EM-MAP method [12].The aim of this paper is to use the SCA methods in application of noise removal, especially for salt and paper noise and random-valued noise. The organization of the paper is as follows. Firstly, our SCA method is introduced in section 2, then this method in combination with popular median filtering is studied in section 3, and at last the simulation results will be discussed.

2. The proposed SCA method

2.1 Basic Idea

The basic idea is that, the representation of the image in the DCT domain is sparse because the most of the coefficients in the DCT domain are zero or near zero. We assume the noisy model as:

$$\mathbf{X}_{N\times N} = \mathbf{S}_{N\times N} + \mathbf{E}_{N\times N} \tag{2}$$

where $\mathbf{S}_{N\times N}$ is the original image and $\mathbf{E}_{N\times N}$ is the impulsive noise and $\mathbf{X}_{N\times N}$ is the noisy image (sub image). If we apply the DCT transform to both sides of equation (2), we have:

$$T(\mathbf{X}) = T(\mathbf{S}) + T(\mathbf{E}) \tag{3}$$

where $\boldsymbol{T}$ is the DCT transform and has the following form:

$$T(\mathbf{S}) = \mathbf{TST}' \tag{4}$$

where $\boldsymbol{T}$ is the DCT transform matrix as defined below [1]:

$$t(x,y) = \alpha(x)\cos((2y+1)\frac{x\pi}{2N})$$
$$\alpha(x) = \begin{cases} \sqrt{\frac{1}{N}} & x = 0 \\ \sqrt{\frac{2}{N}} & x \neq 0 \end{cases} \tag{5}$$

We know that the block of $T(\mathbf{S})$ have many almost zero coefficients. To order this two dimensional matrix to a one dimensional vector with zeros at the end of the vector, we define the zigzag transform. This transform changes a two dimensional matrix to a one dimensional vector, similar to the JPEG standard. We assume that the coefficients of $Z(T(\mathbf{X}))$ are zero from $n+1$ to m. In this case m is the number of pixels in a sub image of size N and so is equal to $m = N^2$. Moreover, n is determined with the compression ratio of the sub image. If the compression ratio of the sub image is defined as CR, then the value of n is equal to $n = \frac{m}{CR}$. The general idea is to use this zeros to find the impulse noises (or errors). At first, we present the general case where the degraded pixels have random values and then switch to a simpler case where the salt and pepper assumption of noise are available.

2.2 Random value impulsive noise

By defining $\tilde{\mathbf{X}} \square Z(T(\mathbf{X}))|_{n+1:m}$, and the previous assumption that transformed original image in the DCT domain is sparse, i.e. $Z(T(\mathbf{S}))|_{n+1:m} = 0$, we will have the following reconstruction formula to find the impulsive noises (or errors):

$$\tilde{\mathbf{X}} = Z(T(\mathbf{E}))|_{n+1:m} \tag{6}$$

If we are able to write the right hand of equation (6) in the linear form of $Z(T(\mathbf{E}))|_{n+1:m} = \mathbf{H}Z(\mathbf{E})$, then the problem of finding errors, converts to a classical SCA formulation as:

$$\tilde{\mathbf{X}} = \mathrm{H}Z(\mathbf{E}) \quad (7)$$

Solving this SCA problem leads to zigzag transform of the errors. Taking the inverse zigzag transform yields the error image (both its value and its position). After subtracting the error image from the noisy image, the estimation of the original image is obtained. We call this method as "SCA method". The block diagram of this method is depicted in Fig. 1.

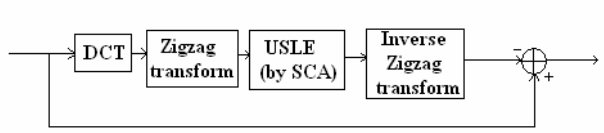


Fig. 1 The block diagram of our method

At first, we should find the matrix $\mathbf{H}$ in terms of the DCT transform. To compute the matrix $\mathbf{H}$, we use the 2-D transform equation in the general form [1]:

$$T(\mathbf{E})|_{(u,v)} = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} E(x,y)t(x,y,u,v) \quad (8)$$

Note that the i'th element of the $Z(T(\mathbf{E}))$ equal to:

$$Z(T(\mathbf{E}))|_i = T(\mathbf{E})|_{(u(i),v(i))} \quad (9)$$

where we can imagine the $[u(i),v(i)]$ as the inverse zigzag transform of the i'th 1-D element. From equations (8) and (9), we can write:

$$Z(T(\mathbf{E}))|_i = [t(0,0,u(i),v(i)),t(0,1,u(i),v(i),..,t(N,N,u(i),v(i))]Z(\mathbf{E}) \quad (10)$$

Therefore, $Z(T(\mathbf{E}))$ can be written as $\mathbf{G}Z(\mathbf{E})$, where the matrix $\mathbf{G}$ is:

$$G_{ij} = t(u(j),v(j),u(i),v(i)) \quad (11)$$

From equations (6), (7), (9) and the preceding discussion, the matrix $\mathbf{H}$ is $\mathbf{H} = \mathbf{G}(n+1:m,1:m)$ where we use MATLAB matrix notation. The matrix $\mathbf{G}$ is obtained simply from equation (11) and knowing that the DCT transform is separable of the form $t(x,y,u,v) = t(x,u)t(y,v)$. So, we have:

$$G_{ij} = t(u(j),u(i)).t(v(j),v(i)) \quad (12)$$

where $t(u(j),v(j),u(i),v(i))$ is defined in equation (5). Finally, the SCA problem in equation (7) can be solved by means of any SCA method such as MP, BP (or known as Linear Programming), smoothed-l^0 or EM-MAP. Since we should divide the image into the sub images and then solve the correspondence SCA problem with different $\tilde{\mathbf{X}}$ and $\mathbf{H}$, so a fast method for SCA is a necessity. Among the various methods, BP (or equivalently LP) and EM-MAP is rather complicated. Moreover, the MP method does not yield the accurate sparse solution of a SCA problem. However, a recently developed method called smoothed-l^0 [11] has the ability to provide a very fast and accurate estimation of the sparse solution. So, in our simulations we use this method.

2.3 Salt and pepper impulsive noise

In the salt and pepper impulsive noise, it is usually assumed that the salt noise is the maximum gray level (255) and the pepper noise is the minimum gray level (0) [5]. So, the places of noisy pixels are easily found by a simple comparison to these values (assuming that our image has not pixels with gray level 0 and 255). In [5], an adaptive median filter is used to detect the noisy pixels. But, in our paper, we assume that our image does not have pixels with gray level 0 and 255, and the noisy pixels are known by a simple comparison with the upper and lower gray levels. So, the only problem is to recover the original gray level of noisy pixels. Therefore, we propose a simpler version of our method. In this case we start from equation (7). Since the positions of errors are known, we can omit the columns of the matrix $\mathbf{H}$ which we know that there is not any error at those places. So, equation (7) converts to the following formula:

$$\tilde{\mathbf{X}} = \mathrm{H}_{truncated} Z(\mathbf{E})_{nonzero} \quad (13)$$

After solving the above equation which is equal to solving a linear system of equations, the nonzero errors are obtained. In this case, the number of errors must be less than the size of the $\tilde{\mathbf{X}}$ vector which is equal to $m-n$. The solution in these cases can be obtained via pseudo-inverse (where the unknowns are smaller than equations). We call this method the "Salt-Pepper SCA method" (SP-SCA).

3. The combined median-SCA-median method

Because of the good properties of the nonlinear filtering and especially median filtering in the image denoising applications, we suggest to use a combination of the traditional median filtering with our SCA method. When the noise level is low, the noisy pixels in a subimage are small and the median value of the sub image is not noisy. But, when the noise level is high the median value itself is a noisy pixel. So, the performance of the median filter is decreased. The median filter can be regarded as a pre-process to reduce the effect of the impulsive noise. After that, we can apply our SCA method. Moreover, in high level noise, this combination also cannot omit all the impulsive noises. Another median filter after our SCA method can omit the remaining impulse noises. So, the block diagram of this combination method is shown in Fig. 2.

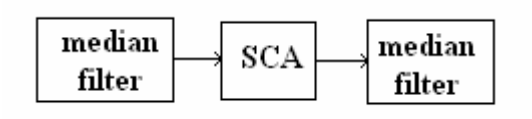


Fig. 2 The block diagram of combination of methods

4. Experiments

Three experiments were done to investigate our SCA method in image denoising when impulsive noise is present. In all experiments, the performance of our SCA method is compared with the median filter and also with the combination of methods. In the first experiment, we use the "SCA method" introduced in Sec. 2.2, and in the second and third experiments, we use the "salt-pepper SCA method" introduced previously in Sec. 2.3.
Our performance measure is the Peak-Signal-to-Noise Ratio (PSNR), defined as:

$$PSNR = 10\log_{10}\left(\frac{255^2}{\frac{1}{MN}\sum_{i,j}(s_{ij}-\hat{s}_{ij})^2}\right) \qquad (14)$$

4.1 Random-valued impulsive noise

In this experiment, random valued impulsive noise with different levels is added to the image. The results of the simulations are shown in Fig. 3. As we can see the combination of the methods has the best result in high level of noise (30% to 60% noise level). In addition to objective measures, the reconstructed images have good results up to 50% impulsive noise. Fig. 4 shows the corrupted image when 50% of pixels are corrupted with random-valued noise. Fig. 5 shows the reconstructed image.

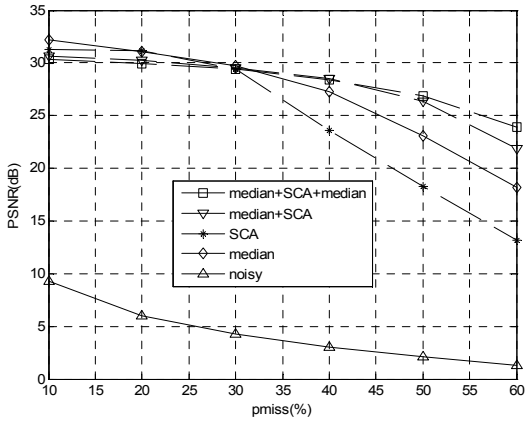


Fig. 3 The results for the random-valued noise

Fig. 4 The 50% random-valued noisy image

Fig. 5 The reconstructed image from 50% random-valued noisy image

4.2 Fixed gray level salt and pepper noise

In this experiment, it is assumed that only fixed gray level salt and pepper noise has corrupted the image (0 for pepper and 255 for salt). In this case, the image is reconstructed by the "salt-pepper SCA method" as introduced in Sec. 2.3. The results of various methods are depicted in Fig. 6. As it can be seen, our combination of methods has slightly better results especially at high noise levels. In this case, we can reconstruct the images ever it is corrupted by 60% salt and pepper noise. The noisy image and the reconstructed image in this case are shown in Fig. 7 and Fig. 8 respectively.

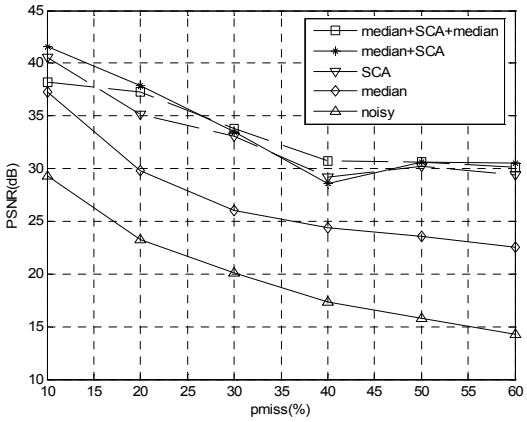


Fig. 6 The result for the fixed gray level salt and pepper noise

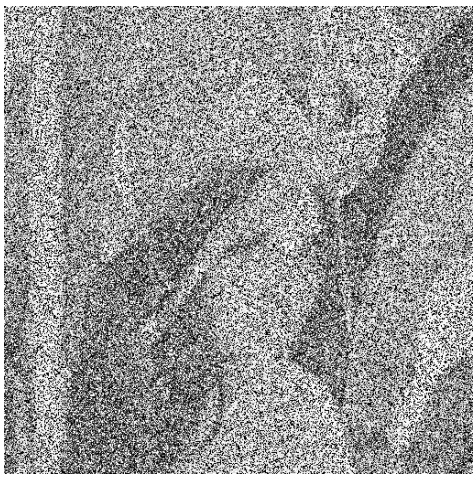


Fig. 7 The 60% fixed salt and pepper noisy image

Fig. 8 The reconstructed image from 60% fixed salt and pepper noise

4.3 Missing sample

In this experiment, we assume that some pixels of the image are missed. So, those pixels are dark and have zero gray level. Similar to the previous experiment, the reconstruction of image is done by the "salt-pepper SCA method" as introduced in Sec. 2.3. The result of the simulations is shown in Fig. 9. In this case, the reconstruction was done appropriately up to 40% of missed samples. The missed-sample image and reconstructed image are shown in Fig. 10 and Fig. 11 respectively.

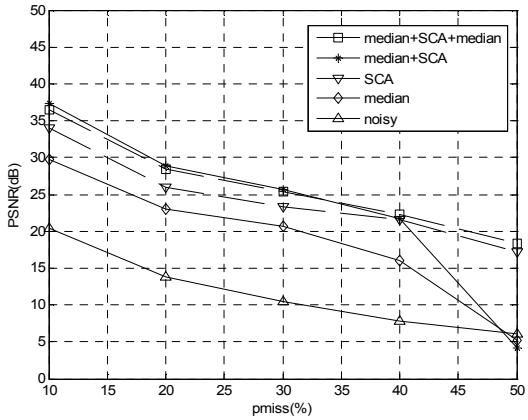


Fig. 9 The result for the missing sample experiment

5. Conclusion

In this paper, a novel method is proposed to remove impulsive noise from images. This method is essentially based on the sparsity of the images in the DCT domain. Using the nearly zeros in the DCT domain, an exact equation is provided to recover the impulse noises (or errors). To solve

this equation, the smoothed-l^0 method [11] is utilized. In addition, in the simple case of fixed gray level salt and pepper noise, we present a new version of our method. To obtain better results when high level of noise is present, a combination of our SCA method with traditional median filtering is suggested. The simulation results show the efficiency of our method in the three cases of impulsive noise (random-value, fixed salt and pepper and missing sample).

Fig. 10 The 40% missed-sample image

Fig. 11 The reconstructed image from 40% missed sample image

Acknowledgement

The authors would thank Advanced Communication Research Institute (ACRI) and Iran National Science Foundation (INSF) for financially supporting this work.

References

[1] A. K. Jain, Fundamentals of Image Processing, Prentice Hall, 1989.

[2] H. Hwang and R. A. Haddad, "Adaptive median filters: New algorithms and results" *IEEE Trans. Image Processing*, Vol. 4, No. 4, pp. 499-502, Apr. 1995.

[3] G. Pok and J. C. Liu and A. S. Nair, "Selective removal of impulse noise based on homogeneity level information" *IEEE Trans. Image Processing*, Vol. 12, No. 1, pp. 85-92, Jan. 2003.

[4] T. Chen and H. R. Wu, "Adaptive impulse detection using center-weighted median filters" *IEEE Signal Processing Letters*, Vol. 8, pp. 1-3, Jan. 2001.

[5] R. H. Chan and C. W. Ho and M. Nikolova, "Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization" *IEEE Trans. Image Processing*, Vol. 14, No. 10, pp. 1479-1485, Oct. 2005.

[6] Y. Dong and S. Xu, "A new directional weighted median filter for removal of random-valued impulse noise" *IEEE Signal Processing Letters*, Vol. 14, pp. 193-196, Mar. 2007.

[7] R. Gribonval and S. Lesage, "A survey of sparse component analysis for blind source separation: principles, perspectives, and new challenges," *Proc. ESANN'06*, pp. 323-330, 2006.

[8] D. L. Donoho, "For most large underdetermined system of linear equations the minimal l^1-norm solution is also the sparsest solution," *Technical Report,* 2004.

[9] S. S. Chen and D. L. Donoho and M. A. Saunders, "Atomic decomposition by basis pursuit" *SIAM Journal on Scientific Computing*, Vol. 20, No. 1, pp. 33-61, 1999.

[10] I. F. Gorodnitski and B. D. Rao, "Sparse signal reconstruction from limited data using FOCUSS: a re-weighted norm minimization algorithm" *IEEE Trans. Signal Processing*, Vol. 45, pp. 600-616, 1997.

[11] G. H. Mohimani and M. Babaie-zadeh and C. Jutten, "Fast sparse representation based on smoothed-l^0 norm," *Proc. ICA'07*, Sep 2007.

[12] H. Zayyani and M. Babaie-zadeh and G. H. Mohimani and C. Jutten, "Sparse component analysis in presence of noise using an iterative EM-MAP algorithm," *Proc. of ICA'07*, Sep 2007

Face Detection Using Adaboosted RVM-based Component Classifier

Ali Reza Bayesteh Tashk, Abolghassem Sayadiyan, SeyyedMajid Valiollahzadeh
Electrical Engineering Department, Amirkabir University of Technology,
15914 Tehran, Iran
Bayesteh_ar@yahoo.com,eea35@aut.ac.ir,valiollahzadeh@yahoo.com

Abstract

In this paper, a new Adaboosted Kernel Classifier algorithm is introduced for face detection application. However, most of the methods used to implement Relevance Vector Machine (RVM), need lengthy computation time when faced with a large and complicated dataset. A new pruning method is used to reduce the computational cost.

The kernel classifier parameters are adaptively chosen. In addition, using Fisher's criterion, a subset of Haar-like features is selected. As a result, our proposed algorithm with its previous counterparts i.e. Support Vector Machine (SVM) and RVM without boosting is compared, which results in a better performance in terms of generalization, sparsity and real-time behavior for CBCL face database.

1. Introduction

Nonlinear classification of data is always of special attention. Face Detection is a problem dealing with such data, due to large amount of variation and complexity brought about by changes in facial appearance, lighting and expression. Feature selection is needed beside appropriate classifier design to solve this problem, like many other pattern recognition tasks.

Tipping's Relevance Vector Machines (RVM) [1] [3] are a Bayesian approach leading to a probabilistic non-linear model with a prior on the weights that promotes sparse solutions. The advantage of RVM over non-Bayesian kernel methods comes from explicit probabilistic formulation that yields predictive distributions for test instances and allows Bayesian model selection [4].

One of the major developments in machine learning in the past decade is the Ensemble method, which finds a highly accurate classifier by combining many moderately accurate component classifiers. Boosting [15] and Bagging [16] are the most common techniques, used to construct Ensemble classifiers. In Comparison with Bagging, Boosting outperforms when the data do not have much noise [17] [18]. Among popular Boosting methods, AdaBoost [6] establishes a collection of weak component classifiers by maintaining a set of weights over training samples and adjusting them adaptively after each Boosting iteration the weights of the misclassified samples by current component classifier will be increased while the weights of the correctly classified samples will be decreased. To implement the weight updates in Adaboost, several algorithms have been proposed [19]. The success of Adaboost can be attributed to its ability to enlarge the margin [5], which could enhance Adaboost's generalization capability. All these factors have to be carefully tuned in practical use of Adaboost. Furthermore, diversity is known to be an important factor which affects the generalization accuracy of Ensemble classifiers [21][19]. In order to quantify the diversity, some methods are proposed [19] [22]. It is also known that in Adaboost there exists an accuracy/diversity dilemma [9], which means that the more accurate two component classifiers become, the less they can disagree with each other. Only when the accuracy and diversity are well balanced, the Adaboost can demonstrate excellent generalization performance. However, the existing Adaboost algorithms do not yet explicitly taken sufficient measurement to deal with this problem.

In this paper we propose a new kernel classifier for face detection. Applying boosted RVM has an advantage of getting accuracy and being Sparse. Boosting will reduce the sparsity in nature, while RVM will compensate this fact. Obtaining accuracy and sparsity will allow the system operate very fast. The rest of the paper is organized as follows Sections 2 describes the feature selection method. In Section 3 we introduce RVM and Adaboost, and then we develop AdaboostRVM. In Section 4, we apply the proposed method for face detection. Finally, we provide discussions and conclusions in Section 5.

2. Feature selection

To find out which features to be used for a particular problem, is referred as feature selection. In this paper, like Viola and Jones [10], we use four types of Haar-like basis functions for feature selection which have been used by Papageorgiou et al [9].

Like their work, we use four types of haar-like feature to build the feature pool. The features can be computed efficiently within integral image. The main objective to use these features is that they can be rescaled easily which avoids to calculate a pyramid of images and yields to fast operation of the system on these features. These features can be seen in Figure 1. Given that the base resolution of the detector is 19x19, the exhaustive set of rectangle features is quite large. Note that unlike the Haar basis, the set of rectangle features is over complete. Like viola, we use image variance σ to correct lighting, which can be got using integral images of both original image and image squared.

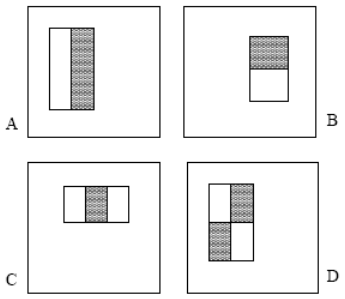


Figure 1. Example rectangle features shown relative to the enclosing detection window. The sum of the pixels which lie within the white rectangles is subtracted from the sum of pixels in the grey rectangles. Two-rectangle features are shown in (A) and (B). Figure (C) shows a three-rectangle feature, and (D) a four-rectangle feature.

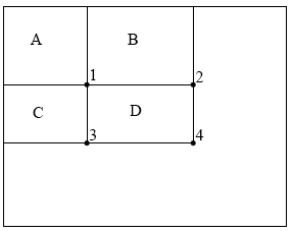


Figure 2.The sum of the pixels within rectangle D can be computed with four array references. The value of the integral image at location 1 is the sum of the pixels in rectangle A. The value at location 2 correspond to the area A+B and so on

Using the integral image any rectangular sum can be computed in four array references (see Figure 2). Clearly the difference between two rectangular sums can be computed using eight references. Since the two-rectangle features defined above involve adjacent rectangular sums they can be computed in six array references, eight in the case of the three-rectangle features, and nine for four-rectangle features.

we use Fisher's score for between-class measurement as:

$$S_i = \left| \frac{m_{i,face} - m_{i,nonface}}{\sigma^2_{i,face} + \sigma^2_{i,nonface}} \right| \qquad (1)$$

By selecting the feature with highest Fisher's scores and smallest spatial correlation, we can retain the most discriminative feature between face and non-face classes

3. Statistical Learning

In this section, we describe boost based learning methods to construct face/nonface classifier, and propose a new boosting algorithm which improves boosting learning.

3.1. AdaBoost Learning

Given a set of training samples, AdaBoost [7] maintains a probability distribution, W, over these samples. This distribution is initially uniform. Then, AdaBoost algorithm calls a WeakLearn algorithm repeatedly in a series of cycles. At cycle T, AdaBoost provides training samples with a distribution w^t to the WeakLearn algorithm.

AdaBoost, constructs a composite classifier by sequentially training classifiers while putting more and more emphasis on certain patterns.

For two class problems, we are given a set of N labeled training examples $(y_1, x_1), ..., (y_N, x_N)$, where $y_i \in \{+1, -1\}$ is the class label associated with example x_i .

For face detection, x_i is an image sub-window of a fixed size containing an instance of the face $(y_i = +1)$ or non-face $(y_i = -1)$ pattern. In the notion of AdaBoost see table 1, a stronger classifier is a linear combination of M weak classifiers.

In boosting learning [15], each example x_i is associated with a weight w_i, and the weights are updated dynamically using a multiplicative rule according to the errors in previous learning so that more emphasis is placed on those examples which are erroneously classified by the weak classifiers learned previously.

Greater weights are given to weak learners having lower errors. The important theoretical property of AdaBoost is that if the weak learners consistently have accuracy only slightly better than half, then the error of the final hypothesis drops to zero exponentially fast. This means that the weak learners need be only slightly better than random.

Furthermore, since proposed AdaBoost with RVM invents a convenient way to control the classification accuracy of each weak learner, it also provides an opportunity to deal with the well-known accuracy/diversity dilemma in Boosting methods. This is a happy accident from the investigation of AdaBoost based on RVM weak learners.

Table 1. The AdaBoost with RVM Algorithm [3] .

1. Input: Training sample
Input: a set of training samples with labels $(y_1, x_1), ..., (y_N, x_N)$, ComponentLearn algorithm, the number of cycles *T*.

2. Initialize: the weights of training samples: $w_i^1 = 1/N$, for all $i = 1, ..., N$

3. Do for $t = 1, ..., T$
(1)Use ComponentLearn algorithm to train the component classifier *ht* on the weighted training sample set.
(2)Calculate the training error of h_t :

$$\varepsilon_t = \sum_{1=1}^{N} w_i^t, y_i \neq h_t(x_i) \quad (2)$$

(3)Set weight of component classifier h_t :

$$h_t : \alpha_t = \frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right) \quad (3)$$

(4)Update the weights of training samples:

$$w_i^{t+1} = \frac{w_i^t \exp\{\alpha_t y_i h_t(x_i)\}}{C_t} \quad (4)$$

where C_t is a normalization constant, and

$$\sum_{i=1}^{N} w_i^{t+1} = 1 \quad (5)$$

4. Output: $f(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$.

3.2. RVM for classification

$$y(X;W) = \sum_{i=1}^{N} w_i K(X,\mathrm{X}_i) + w_0 \quad (6)$$

Where $K(X,\mathrm{X}_i)$ is a kernel function, effectively defining one basis function for each example in the training set.

Relevance vector machine (RVM) is a Bayesian framework for achieving the sparse linear model (6). In sparse model, the majority of the W s are zero. The sparsity of model is based on a hierarchical prior, where an independent Gaussian prior is defined on the weight parameters in the first level:

$$p(W|\alpha) = \prod_{i=1}^{N} N\left(w_i|0,\alpha_i^{-1}\right) \quad (7)$$

Where $\alpha = (\alpha_1,\alpha_2,...,\alpha_N)$ is a vector consisting of N hyper parameters. An independent Gamma hyper prior is used for the variance parameters in the second level:

$$p(\alpha_i) = Gamma(a,b) \quad (8)$$

Where a and b are constants. The key point of this method is using the maximum a posteriori (MAP) instead of maximum likelihood (ML) for the Weight estimation.

Given the N pairs of training data $\{\mathrm{X}_l,t_l\}\Big|_{l=1}^{N}$, the dataset likelihood is defined by applying the logistic sigmoid link function $\sigma(y) = 1/1+e^{-y}$ to $y(X)$ and adopting the Bernoulli distribution for $P(t|X)$:

$$P(t|W) = \prod_{n=1}^{N} \sigma\{y(X_n;W)\}^{t_n}\left[1-\sigma\{y(X_n;W)\}\right]^{1-t_n} \quad (9)$$

Where class label is denoted by $t_l \in \{0,1\}$. The parameters w_i are then obtained by maximizing the posterior distribution of the class labels given the input vectors with respect to prior information. For this maximization, a numerical method is suggested as follows:

1. For the current, fixed, values of α, the most probable' weights W_{MP} are found, giving the location of the mode of the posterior distribution.

Since $P(W|t,\alpha) \propto P(t|W)P(W|\alpha)$ this is equivalent to finding the maximum, over W, of:

$$\log\left\{P(t|W)P(W|\alpha)\right\} = \sum_{n=1}^{N} t_n \log y_n + (1-t_n)\log(1-y_n) - \frac{1}{2}W^T AW \quad (10)$$

With $y_n = \sigma\{y(X_n;W)\}$

2. Laplace's method is simply a quadratic approximation to the log-posterior around its mode. The quantity (10) is differentiated twice to give:

$$\nabla_W \nabla_W \log P(W|t,\alpha)\Big|_{W_{MP}} = -\left(\Phi^T B\Phi + A\right) \quad (11)$$

Where

$B = diag(\beta_1,\beta_2,...,\beta_N)$ $\beta_n = \sigma\{y(X_n)\}\left[1-\sigma\{y(X_n)\}\right]$

The posterior is approximated around W_{MP} by a Gaussian approximation with Covariance:

$$\Sigma = \left(\Phi^T B\Phi + A\right)^{-1} \quad (12)$$

And mean

$$\mu = \Sigma\Phi^T Bt \quad (13)$$

3. Using the statistics Σ and μ of the Gaussian approximation, the hyper parameters α are updated as follows:

$$\alpha_i^{new} = \frac{\gamma_i}{\mu_i^2} \quad (14)$$

where μ_i is the i-th posterior mean weight from(14) and $\gamma_i \equiv 1 - \alpha_i^{old} N_{ii}$ which N_{ii} is the i-th diagonal element of Σ .Since computing the μ and Σ based on above mentioned steps takes so much time, we use incremental DFT-RVM for simplicity on implementation.

3.3. Data Pruning

When we are faced to a large and complicated dataset, the accuracy of RVM classification is not as high as expected and the computation time increases rapidly. Therefore, improving the efficiency of RVM is one important area of study.

Now, we present a simple statistical algorithm to identify the most crucial points of the training data. The basic idea is to model the face class as a multivariate normal distribution, which is especially reasonable if one, models only the upright frontal faces that are properly aligned to one another. Note that the training

face images are all upright, frontal, and properly aligned. Therefore, the density function of the face class is modeled as a multivariate normal distribution as follows:

$$p\left(Y\left|w_f\right.\right) = \frac{1}{\left(2\pi\right)^{N/2}\left|\Sigma\right|^{1/2}} \times \exp\left\{-\frac{1}{2}\left(Y-M\right)^t \Sigma^{-1}\left(Y-M\right)\right\} \quad (15)$$

Where $Y \in \mathbb{R}^N$ the discriminating is feature vector and $M \in \mathbb{R}^N, \sum \in \mathbb{R}^{N\times N}$ are the mean vector and the covariance matrix of the face class w_f, respectively.

Afterwards, we model non-face class PDF with a Gaussian mixture model.

$$p\left(Y\left|w_n\right.\right) = \sum_{i=1}^{M} W_i N\left(Y; M_i, \Sigma_i\right) \quad (16)$$

As a result, the crucial data are introduced as follows:

$$\varepsilon_1 \le Log\left(\frac{p(Y|w_n)}{p(Y|w_f)}\right) \le \varepsilon_2 \quad (17)$$

Where the remaining points obtained above, are the ones hardly separable.

The data obtained according to aforementioned scheme, can now be applied to a learning machine

3.4. Adaboosted RVM-Based Classifier

We combine RVM with Adaboost to improve its capability in classification. A polynomial RVM with kernel $K\left(X, X_l\right) = \left(1 + s\, X \cdot X_l\right)^d$ is used in our experiments [2].

RVM weak classifiers are obtained by selecting the polynomial parameters, s and d, then these weak classifiers (classifier error place in range of %55 to %65) are used for optimizing strong classifiers (Adaboost classifier).

3.5. Face Detection System

We explain our face detection system and show how to construct an Adaboosted RVM-based component classifier for face detection. The learning of a detector is done as follows:

1. A set of simple Haar wavelet features are used as candidate features. There are tens of thousands of such features for a 19x19 window.

2. A subset of them based on fisher's score are selected and the corresponding weak classifiers are constructed, using Adaboosted RVM-based component classifier learning. Data pruning is applied to reduce the number of effective samples but it helps to get higher training speed without losing the accuracy in general.

3. A strong classifier is constructed as a linear combination of the weak ones.

4. Experimental results

We adopt a face image database from the Center for Biological and Computational Learning at Massachusetts Institute of Technology (MIT), which contains 2429 face training samples, 472 face testing samples, and 23,573 non-face testing samples. We randomly collected 15,000 non-face training samples from the images that do not contain faces.

We compared RVM and SVM with the same input vectors and 2nd polynomial kernel without boosting. In this stage we generated the input vector by applying a mask on images in our database.

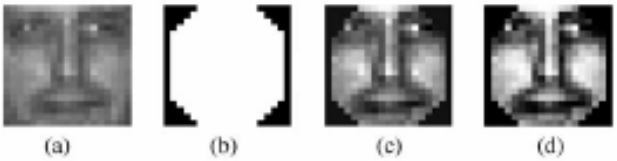


Figure 3.(a) Original face image, (b) The mask, (c) Normalized image and (d) Histogram equalized image

Next we performed normalization and the histogram equalization on resulted image. Figure 3 shows these steps [2]. Then we used the face training samples to calculate 50 Principal Analysis Component (PCA) features.

In the other experiment we calculated 50 Fisher's features and used them as the features of the 2nd polynomial kernel RVM and SVM classifier without boosting.

As we can see in the Figure 4, 50 PCA features outperforms in the terms of accuracy than 50 Fisher's features. This experiment showed RVM is better than SVM classifier.

Our experiment showed that the sparseness of RVM is more than SVM classifier and in testing phase it makes the RVM work fast. Table 2. Compares the sparseness of this approach. Another reason that this method works fast is the advantageous usage of Fisher's feature instead of PCA features. The number of multiplications required for computing Fisher's features are very less than PCA features. Also Figure4 shows that AdaboostRVM by applying pruning performs nearly to AdaboostRVM in accuracy but it reduces the number of samples greatly. Our methods used the highest 50 Fisher's scores features. Figure 4 shows the ROC graph of our method. According to this Figure, it is clear that the performance of the proposed method is much better than the SVM and RVM without boosting.

5. Conclusions

An Adaboosted method is proposed in this paper in order to combine a group of week RVMs which adaptively adjusts the kernel parameters of RVM classifier to get the best result. Experimental results on CBCL database for Face Detection demonstrated that

the proposed AdaboostRVM algorithm performs better than other approaches such as SVM and RVM without being Adaboosted in accuracy and speed.

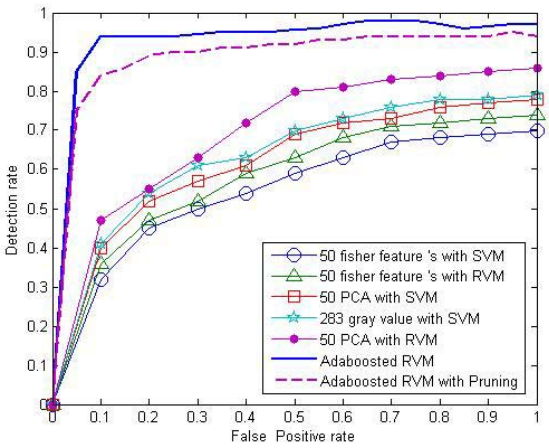


Figure 4. RVM and SVM Comparison

Table 2. Comparison of the sparseness

	SVM	RVM	Adaboosted -RVM	Adaboosted RVM with Pruning
283 gray level	792	--	--	--
50 PCA	766	185	--	--
50Fisher 's feature	529	107	586	427

Experimental results show that AdaboostRVM with pruning, results in a better performance in terms of computational cost and sparsity. Due to this fact that by applying pruning, number of effective samples will be reduced without losing the accuracy noticeably.
Besides these, it is found that proposed AdaboostRVM algorithm demonstrated a better performance on imbalanced classification problems. Based on the AdaboostRVM, an improved version is further developed to deal with the accuracy/diversity dilemma in Boosting algorithms, in raising a better generalization performance. Experimental results indicate that the performance of the Adaboost classifier with RVM is overlay superior to those obtained by the SVM and RVM.

6. References

[1] M.E. Tipping, "Sparse Bayesian Learning and the Relevance Vector Machine", J. Machine Learning Research, vol. 1, 2001, pp. 211-244.

[2] Frank Y., Shouxian Cheng, Gravano, "Improved feature reduction in input and feature spaces", Pattern Recognition 38, 2005, 651-659

[3] Tipping M. E., Faul A., "Fast Marginal Likelihood Maximization for Sparse Bayesian Models", Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics, Jan 3-6, 2003.

[4] Catarina Silva , Bernardete Ribeiro., " Two-level hierarchical hybrid SVM-RVM classification model", Proceedings of the 5th International Conference on Machine Learning and Applications, 2006

[5] Schapire, R. E., Freund, Y., "Boosting the margin: a new explanation for the effectiveness of voting methods", The Annals of Statistics, 26(5), pp.1651–1686, October 1998.

[6] Freund, Y., Schapire, R., Aug 1997 "A decision-theoretic generalization of on-line learning and an application to boosting". Journal of Computer and System Sciences, 55(1):119–139.

[7] Schapire R. E., Y. Singer, "Improved boosting algorithms using confidence-rated predictions,Machine Learning, 37(3), pp.297–336, Dec 1999.

[8] Friedman, J., Hastie, T., R. Tibshirani,. "Additive logistic regression: a statistical view of boosting", Technical report, Department of Statistics, Sequoia Hall, Stanford University, July 1998.

[9] Dietterich, T. G., Aug 2000, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization", Machine Learning, vol. 40, no. 2, Aug 2000, pp. 139–157.

[10] Papageorgiou, C., Oren, M., Poggio, T., "A general Framework for object detection", In International Conference on Computer Vision, 1998.

[11] Viola, P., Jones, M., "Rapid Object Detection Using a Boosted Cascade of Simple Features," Proc. IEEE CS Conf. Computer Vision and Pattern Recognition, Dec. 2001

[12] Rowley, H., Baluja, S., Kanade, T., "Neural network-based face detection", In IEEE Patt. Anal. Mach. Intell., volume 20, pp.22–38,1998.

[13] Li, S. Z., EE, Zhang, Z. Q. ,"FloatBoost Learning and Statistical Face Detection", In IEEE Patt. Anal. Mach. Intell., vol. 26, no. 9, sept. 2004,

[14] Haykin, S., Neural networks: A comprehensive foundation. Prentice Hall, July 1998.

[15] Lienhart, R., Kuranov, A., Pisarevsky, V.,"Empirical analysis of detection cascades of boosted classifiers for rapid object detection", 2003.

[16] schapire. R. E., "The boosting approach to machine learning: An overview", In MSRI Workshop on Nonlinear Estimation and Classification, 2002.

[17] Breiman. L, "Bagging predictors", Machine Learning, 24, pp.123–140, 1996.

[18] Opitz, D., Maclin, R. "Popular ensemble methods: An empirical study", Journal of Artificial Intelligence Research, 11, pp.169–198, 1999.

[19] Bauer, E., Kohavi, R., "An empirical comparison of voting classification algorithms: Bagging, boosting, and variants", Machine Learning, 36(1),pp.105–139 Jul 1999

[20] Kuncheva, L. I., Whitaker, C. J., "Using diversity with three variants of boosting: aggressive, conservative, and inverse", In Proceedings of the Third International Workshop on Multiple Classifier Systems, 2002.

[21] Schwenk , H. and Bengio. Y, "Boosting neural networks", Neural Computation, 12, pp.1869–1887, 2000.

[22] Melville P., Mooney. R. J, "Creating diversity in ensembles using artificial data", Information Fusion, 6(1), pp.99–111, Mar2005.

SPEAKER-INDEPENDENT VOWEL RECOGNITION IN PERSIAN SPEECH

Mohammad Nazari, Abolghasem Sayadiyan, Seyyed Majid Valiollahzadeh

Amirkabir University of Technology
Electrical Engineering Department,
Tehran, Iran, 15914

ABSTRACT

In this paper we discuss the applicability of the kernel-based feature extraction for speaker-independent vowels recognition, focusing on non-linear dimension reduction methods. The Increasing of feature space dimension lead us to improve accuracy of vowels recognition system but we lost realtime system. So, using dimension reduction algorithms, help us to improved accuracy and we study the applicability of this idea to build a quasi-realtime system in Persian speech. In Vowels Recognition and other similar applications that need a mapping technique that introduces representation of low-dimensional features with enhanced discriminatory power and a proper classifier, able to classify those complex features. In this short paper, we combine nonlinear kernel based mapping of data with Support Vector machine (SVM) classifier to improve efficiency of system. The proposed here method is compared, in terms of classification accuracy, to other commonly used Vowels Recognition methods on FarsDat database.

Index Terms— Kernel-Based feature extraction, Speaker-independent Vowel recognition, Persian speech

1. INTRODUCTION

In Recent years, automatic speech processing like Automatic Speech Recognition (ASR) becomes very important and popular since it can contribute to the natural language recognition. ASR technology and other speech processing application have been developed very quickly in many fields, especially in the Internet, telecom and security. In these applications, the vowel recognition generally plays an important role [1].

For example, approaches to large vocabulary continuous speech recognition are based on acoustic modeling of subword units of speech such as context-dependent phones (diphones and triphones) [1] and syllables [2].

In many languages, the Consonant-Vowel (CV) units have the highest frequency of occurrence among different forms of subword units. Therefore, recognition of CV units with a good accuracy is crucial for development of a speech recognition system. Recognition of these subword units is a large-class-set pattern classification problem because of the large number (typically, a few thousands) of units [3]. In this case, if ASR recognizes the vowel with a good accuracy, system can reduce region of search and improve accuracy and time. With this reason, we try to establish a vowel recognition system. In our auditory system, same vowels pronounced by different people with different gender, different age, or by the same person using a different pitch can be recognized.

For the purpose of data reduction and feature extraction in pattern recognition, Principle component analysis (PCA) and linear discriminant analysis (LDA) are introduced as two powerful tools. It is generally believed that, LDA based algorithms outperform PCA based ones in solving problems of pattern classification, since the former optimizes the low-dimensional representation of the objects with focus on the most discriminant feature extraction while the latter achieves simply object reconstruction.

The limited success of these methods should be attributed to their linear nature. Kernel discriminant analysis algorithm, (KDA) [4] generalizes the strengths of the presented LDA. Recently, more effective solutions, called Direct LDA (D-LDA) methods, have been presented for image processing purpose like face recognition [5], [6]. Although successful in many cases, linear methods fail to deliver good performance when face patterns are subject to large variations in viewpoints, which results in a highly non-convex. In this paper, we try to use nonlinear mapping characteristic of kernels with Direct LDA idea from pattern recognition problems for establishing a Speaker-Independent Vowel Recognition system. The kernel techniques while at the same time overcomes many of their shortcomings and limitations [7].

In this work, we first nonlinearly map the original input space to an implicit high-dimensional feature space, where the distribution of Vowel patterns is hoped to be linearized and simplified. Then, KDDA method is introduced to effectively derive a set of optimal discriminant basis vectors in the feature space. And then SVM approach is used for classification.

The rest of the paper is organized as follows. In Section tow, we start the analysis by briefly reviewing KDDA method. Following that in section three, SVM is introduced and analyzed as a powerful classifier. In Section four, a set of experiments are presented to demonstrate the effectiveness of

the KDDA algorithm together with SVM classifier on highly nonlinear, highly complex face pattern distributions. The proposed method is compared, in terms of the classification error rate performance, to other methods like KPCA (kernel based PCA) on the FarsDat speech database. Conclusions are summarized in Section five.

2. KERNEL-BASED FEATURE EXTRACTION

2.1. Kernel Discriminant Analysis (KDA)

In the statistical pattern recognition tasks, Kernel Discriminante Analysis (KDA) is the nonlinear kernel version of LDA to deal with the feature extraction and the classification of nonlinear characteristics [4].

The problem of feature extraction can be stated as follows: Assume that we have a training set,$\{X\}_{i=1}^{M}$ is available. It is further assumed that each input belongs to one of C classes. For a given nonlinear mapping φ , the input space, $\Re^{\aleph}$, can be mapped into the feature, F, $\phi : \Re^{\aleph} \rightarrow F$ space . Note that the feature space could have a much higher, possibly infinite, dimensionality.

Let S_{BTW} and S_{WTH} be the between- and within- class scatter matrices in the feature space F respectively, expressed as follows:

$$S_{BTW}^{\varphi} = \frac{1}{M}\sum_{i=1}^{C} c_i (m_i^{\varphi} - m^{\varphi})(m_i^{\varphi} - m^{\varphi})^T \quad (1)$$

$$S_{WTH}^{\varphi} = \frac{1}{M}\sum_{i=1}^{C}\sum_{j=1}^{C_i} (\varphi(X_{ij}) - m_i^{\varphi})(\varphi(Z_{ij}) - m_i^{\varphi})^T \quad (2)$$

Where m_i^{φ} is the mean of class i and m^{φ} is the average of the ensemble.

$$m_i^{\varphi} = \frac{1}{C_n}\sum_{i=1}^{C} \varphi(X_{ij}) \quad (3)$$

$$m^{\varphi} = \frac{1}{M}\sum_{i=1}^{C}\sum_{j=1}^{C_i} \varphi(X_{ij}) \quad (4)$$

Where C_i is the number of observations of class i. The maximization can be achieved by:

$$J^{\varphi} = \frac{\psi^T S_{BTW}^{\varphi} \psi}{\psi^T S_{WTH}^{\varphi} \psi} \quad (5)$$

The optimal discriminant vectors in feature space F can be obtained by solving the eigenvalue problem:

$$(S_{BTW}^{\varphi})(S_{WTH}^{\varphi})^{-1}\Psi = \Lambda(S_{BTW}^{\varphi})(S_{WTH}^{\varphi})^{-1}\Psi \quad (6)$$

where Λ is eigenvalues matrix and Ψ is eigenvectors matrix.

2.2. Kernel Direct Discriminant Analysis (KDDA)

We saw that KDA can become numerically unstable because of the invertibility problem of the Within-class Scatter Matrix. Furthermore, the non orthogonality of the resulting transformation matrix may prove disadvantageous. These issues give rise to the need for KDDA [4].

The maximization process in Eq.(5) is not directly linked to the classification error which is the criterion of performance used to measure the success of the vowel recognition procedure. Modified versions of the method, such as the Direct LDA (D-LDA) approach, use a weighting function in the input space, to penalize those classes that are close and can potentially lead to misclassifications in the output space [4].

Most LDA based algorithms including D-LDA [5] utilize the conventional Fisher's criterion denoted by Eq.(7). The introduction of the kernel function allows us to avoid the explicit evaluation of the mapping. Any function satisfying Mercer's condition can be used as a kernel, and typical kernel functions include polynomial function, radial basis function (RBF) and multi-layer perceptrons [Vapnik, 1995].

$$\Psi = \arg\max_{\Psi} \frac{|\Psi^T(S_{BTW}^{\varphi})\Psi|}{|\Psi^T(S_{BTW}^{\varphi})\Psi + \Psi^T(S_{WTH}^{\varphi})\Psi|} \quad (7)$$

The KDDA method implements an improved D-LDA in a high-dimensional feature space using a kernel approach. KDDA introduces a nonlinear mapping from the input space to an implicit high dimensional feature space, where the nonlinear and complex distribution of patterns in the input space is "linearized" and "simplified" so that conventional LDA can be applied.

In Generalized discriminant analysis (GDA), to remove the null space of , it is required to compute the pseudo inverse of the kernel matrix K, which could be extremely ill-conditioned when certain kernels or kernel parameters are used. Pseudo inversion is based on inversion of the nonzero eigenvalues.

3. SVM BASED APPROACH FOR CLASSIFICATION

The principle of Support Vector Machine (SVM) [Vapnik, 1995] [8] relies on a linear separation in a high dimension feature space where the data have been previously mapped, in order to take into account the eventual non-linearities of the problem.

3.1. Support Vector Machines (SVM)

Subheadings should appear in lower case (initial word capitalized) in boldface. They should start at the left margin on a separate line. If we assume that, the training set $X = (x_i)_{(i=1)}^{l} \subset \Re^R$ where $l$the number of training vectors, R is the number of modalities, is labeled with two class targets$Y =$

$(y_i)_{(i=1)}^{l}$, where :

$$y_i \in \{-1, +1\} \qquad \Phi : \Re^R \rightarrow F \tag{8}$$

Maps the data into a feature space F. Vapnik has proved that maximizing the minimum distance in space F between $\Phi(X)$ and the separating hyper plane $H(w, b)$ is a good means of reducing the generalization risk. Where:

$$H(w, b) = \{f \in F | < w, f >_F + b = 0\} \tag{9}$$

where $<>$ is inner product. Vapnik also proved that the optimal hyper plane can be obtained solving the convex quadratic programming (QP) problem:

$$Maximize: \qquad \frac{1}{2} \parallel w \parallel^2 + c \sum_{i=1}^{l} \zeta_i \tag{10}$$

$$With: \qquad y_i(< w, \Phi(X) > + b) \geq i - \zeta_i \tag{11}$$

Where constant C and slack variables x are introduced to take into account the eventual non-separability of $\Phi(X)$ into F.

In practice this criterion is softened to the minimization of a cost factor involving both the complexity of the classifier and the degree to which marginal points are misclassified, and the tradeoff between these factors is managed through a margin of error parameter (usually designated C) which is tuned through cross-validation procedures.

Although the SVM is based upon a linear discriminator, it is not restricted to making linear hypotheses. Non-linear decisions are made possible by a non-linear mapping of the data to a higher dimensional space. The phenomenon is analogous to folding a flat sheet of paper into any three-dimensional shape and then cutting it into two halves, the resultant non-linear boundary in the two-dimensional space is revealed by unfolding the pieces.

The SVM's non-parametric mathematical formulation allows these transformations to be applied efficiently and implicitly: the SVM's objective is a function of the dot product between pairs of vectors; the substitution of the original dot products with those computed in another space eliminates the need to transform the original data points explicitly to the higher space. The computation of dot products between vectors without explicitly mapping to another space is performed by a kernel function.

The nonlinear projection of the data is performed by this kernel functions. There are several common kernel functions that are used such as the linear, polynomial kernel $K(x, y) = < x, y >_{\Re^R} + a^d$ and the sigmoidal kernel $K(x, y) = \tanh(< x, y >_{\Re^R} + a)$, where x and y are feature vectors in the input space. The other popular kernel is the Gaussian (or "radial basis function") kernel, defined as:

$$K(x, y) = \exp(\frac{-|x - y|^2}{2\sigma^2} \tag{12}$$

Where σ is a scale parameter, and x and y are feature-vectors in the input space. The Gaussian kernel has two hyper parameters to control performance C and the scale parameter. In this paper we used radial basis function (RBF).

3.2. Multi-class SVM

The standard Support Vector Machines (SVM) is designed for dichotomic classification problem (two classes, called also binary classification). Several different schemes can be applied to the basic SVM algorithm to handle the K-class pattern classification problem. These schemes will be discussed in this section. The K-class pattern classification problem is posted as follow [9]:

- Given l i.i.d. sample: $(x_1, y_1), ..., (x_l, y_l)$ where x_i, for $i = 1, ..., l$ is a feature vector of length d and $y_i = \{1, ..., k\}$ is the class label for data point x_i.

- Find a classifier with the decision function, $f(x)$ such that $y = f(x)$ where y is the class label for x.

The multi-class classification problem is commonly solved by decomposition to several binary problems for which the standard SVM can be used. For solving the multi-class problem are as listed below:

- Using K one-to-rest classifiers (one-against-all).

- Using $K(K - 1)/2$ pair wise classifiers.

- Extending the formulation of SVM to support the k-class problem.

4. EXPERIMENTAL RESULTS

In this section, first, we briefly describe our database, and then, present simulations results for a speaker independent vowel recognition system.

4.1. DATABASE

We used a subset of clean speech data consisting of 20 male and 20 female utterances with no background noise were extracted from FARSI-DAT (most popular Persian speech database) for the-evaluation experiments.

The training material consisted of 100 word for each speaker. In the database, people do not speak simultaneously. The utterance length is between 400 ms and 5s. We made vowels database by labeling vowels manually in each utterance. Our database focuses on eight important vowels in Persian language (as /a/, /@/, /o/, /e/, /i/, /u/, /au/, /ei/). We used 80 percent of our data for training and 20 percent for evaluation phase.

Specifications of the speech analysis at the acoustic preprocessor are summarized as follows (in Fig.2):

For example, an industry standard Mel-frequency cepstral coefficients (MFCC) front end is typically employed to extract 12 Mel-frequency cepstral coefficients (MFCC) plus the log energy at a frame shift of 12.5 ms. In order to model the

Fig. 1. Speech analysis conditions.

Sampling Frequency	8 kHz
Pre-Emphasis	$1 - 0.98z^{-1}$
Hamming window width	25 ms(200 Point)
Frame period	12.5 ms(100 Point)
LPC analysis order	16-th
Feature parameters	MFCC, Delta MFCC, PLP, Delta PLP, F1, F2

spectral variation of the speech signal, the first and second order derivatives of the 13 coefficients are appended to yield a total of 39 coefficients per frame. Another popular front end that can be used for vowel recognition task based on the perceptual linear prediction (PLP) coefficients, first and second formants (F1 and F2).

4.2. Vowel Recognition System

Our system includes three main stages. In first stage system detects vowels then in second stage extracts features with KDDA feature extraction and in last stage we use SVM as classifier. We will demonstrate the effectiveness of our combined KDDA and SVM proposed method. It is compared with LDA, GDA and pure SVM. We use a radial basis function (RBF) kernel for KDDA and SVM.

We use simple method for vowels detection in continues speech. Like [10], for detecting vowels the modified loudness has to be smoothed over time in order to get a kind of envelope of the modified loudness and the energy envelope.

For each detected vowel, we candidate a fixed length segment then pass this segment to vowel classification phase (KDDA + SVM).

Fig. 2. Speaker independent Vowel Recognition Rate.

Recognition Rate				
	LDA	GDA	SVM	KDDA+SVM
Training Set	93.1 %	94.2 %	92.3 %	96.1 %
Test set	89.2 %	90.8 %	91.4 %	93.9 %

The RBF function is selected for the proposed SVM method and KDDA in the experiments. The selection of scale parameter is empirical.

In addition, in the experiments the training set is selected randomly each time, so there exists some fluctuation among the results. In order to reduce the fluctuation, we do each experiment 15 times and use the average of them. The best result is illustrated in Table 2.

5. DISCUSSIONS AND CONCLUSIONS

A new Vowel Recognition method has been introduced in this paper. The proposed method combines kernel-based methodologies with discriminant analysis techniques and SVM classifier. The kernel function is utilized to map the original vowel patterns to a high-dimensional feature space, where the highly non-convex and complex distribution of patterns is simplified, so that linear discriminant techniques can be used for feature extraction.

Experimental results indicate that the performance of the KDDA algorithm together with SVM is overall superior to those obtained by the pure SVM or GDA approaches. In conclusion, the KDDA mapping and SVM classifier is a general pattern recognition method for nonlinearly feature extraction from high-dimensional input patterns.

We expect that in addition to vowel recognition, KDDA will provide excellent performance in applications where classification tasks are routinely performed, such as phoneme recognition and speaker recognition and verification.

6. REFERENCES

[1] L. Rabiner B. Juang, "Fundamentals of speech recognition," *Prentice Hall*, 1993.

[2] B.A. Ganapathiraju M. Ordowski J. Hamaker, J. Picone and G.R. Doddington, "Syllable-based large vocabulary continuous speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 9, pp. 358–366, May 2001.

[3] C. Chandra Sekhar K. Takeda and F. Itakura, "Recognition of consonant-vowel (cv) units of speech in a broadcast news corpus using support vector machines," in *Pattern Recognition with SVM, Canada*. Springer, 2002, pp. 283–291.

[4] J. Lu K. N. Plataniotis, A. N. Venetsanopoulos, "Face recognition using kernel direct discriminant analysis algorithms," *IEEE Trans. ON Neural Networks*, vol. 14, pp. 117– 126, Jan 2003.

[5] E. C. Liu and H.Wechsler, "Evolutionary pursuit and its application to face recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 570–582, June 2000.

[6] H. Lu S. Ma D. Q. Liu, R. Huang, "Kernel-based optimized feature vectors selection and discriminant analysis for face recognition," in *16th Conferance on Pattern Recognition*. IEEE, 2002, vol. 2, pp. 362–365.

[7] A. Kocsor and L. Toth, "Application of kernel-based feature space transformations and learning methods to phoneme classification," in *ICASSP 98*. IEEE, 1998, vol. 2, pp. 945–948.

[8] V. N. Vapnik, "The nature of statistical learning theory," *Springer-Verlag, New York*, 1995.

[9] R. Tibshirani H. Hastie, "Classification by pairwise coupling," *Technical report, Stanford University and University of Toronto*, 1996.

[10] G. Ruske T. Pfau, "Estimating the speaker rate by vowel detection," in *ICASSP 98*. IEEE, 1998, vol. 2, pp. 945–948.

PROBABILISTIC SVM/GMM CLASSIFIER FOR SPEAKER-INDEPENDENT VOWEL RECOGNITION IN CONTINUES SPEECH

Mohammad Nazari[(a)], *Abolghasem Sayadiyan*[(a)], *SeyedMajid Valiollahzadeh*[(b)]

[(a)]Department of Electrical Engineering, Amirkabir University of Technology, Tehran, Iran
[(b)]Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran

ABSTRACT

In this paper, we discuss the issues in automatic recognition of vowels in Persian language. The present work focuses on new statistical method of recognition of vowels as a basic unit of syllables. First we describe a vowel detection system then briefly discuss how the detected vowels can feed to recognition unit. According to pattern recognition, Support Vector Machines (SVM) as a discriminative classifier and Gaussian mixture model (GMM) as a generative model classifier are two most popular techniques. Current state-of-the-art systems try to combine them together for achieving more power of classification and improving the performance of the recognition systems. The main idea of the study is to combine probabilistic SVM and traditional GMM pattern classification with some characteristic of speech like band-pass energy to achieve better classification rate. This idea has been analytically formulated and tested on a FarsDat based vowel recognition system. The results show inconceivable increases in recognition accuracy. The tests have been carried out by various proposed vowel recognition algorithms and the results have been compared.

Index Terms— Vowel Recognition, Automatic Speech Recognition (ASR), Support Vector Machine (SVM), Gaussian Mixture Model (GMM), Speaker-independent.

1. INTRODUCTION

Some recent researches in speech processing area like Automatic Speech Recognition (ASR) and speaker recognition and verification focus on Vowel Recognition (VR) because of generally spectrally well defined character of vowels. In fact they improve our ability to recognize speech significantly, both by human beings and ASR systems. Therefore the vowel recognition generally plays an important role [1, 2 and 3].

Previous vowel recognition methods like segmental trajectory modeling [3] and HMM using cepstra with their derivatives have some leakage. For example HMM can not model the trajectories of speech signals effectively especially for vowels. In segmental trajectory modeling, the main problem is computational complexity of estimation of transformation matrix to reduce the high correlation within the residual error covariance using Minimum Classification Error (MCE).

In this paper, like traditional segmental modeling methods [5, 6], we proposed a weighted least square estimation to estimate the trajectory feature but for reducing the computational complexity we weaken the updating of transformation matrix then we used the state-of-the-art maximum margin classifier, Probabilistic Vector Machines (PSVM) [9, 10], as a powerfully discriminative function to compensate lack of accuracy. SVM is an effective and accurate discriminative model and it has excellent property of making full use of discriminative information of different classes in the representation pattern variations.

Generative model such as Gaussian Mixture Model (GMM) can construct high performance class models for pattern recognition tasks using statistical information. Earlier works try to combine generative models, particularly GMMs and HMMs, with discriminative framework like SVM [7]. In these systems classifiers are trained to discriminate between individual frames of data then the likelihood scores of each frame are combined using an averaging step [8] to give an overall utterance score from which the authenticity of the speaker may be determined. In this paper we introduce a solution to combine weighted GMM for selecting the SVM training data set to prevail over an important weakness of SVM in large scale databases. Therefore we use GMM score (likelihood) for classifying the easy-to-find members of classes and keep other hard-to-find members, we can reduce number of support vectors.

The rest of the paper is organized as follows. In Section two, we start the analysis by briefly reviewing the SVM and GMM. Following that in section three, we introduce our method as a powerful classifier. In Section four, a set of experiments are presented to demonstrate the effectiveness of our classifier. The proposed method is compared, in terms of the classification error rate performance, to other methods like pure SVM for Speaker-Independent Vowel Recognition on the FarsDat speech database. Conclusions are summarized in Section five.

2. SVM CLASSIFIER WITH GMM TRAINING SET SELECTION

2.1. Support Vector Machines (SVM)

The principle of Support Vector Machine (SVM) relies on a linear separation in a high dimension feature space where the data have been previously mapped considering the eventual non-linearities. Assuming that the training set $X = (x_i)_{i=1}^{l} \subset \mathrm{R}^R$ is labeled with two class targets $Y = (y_i)_{i=1}^{l}$ with l the number of training vectors, R the real line and R number of modalities:

$$y_i \in \{-1,+1\} \quad \Phi : \mathrm{R}^R \rightarrow F \tag{1}$$

Y ,Maps the data into a feature space F. it has been proved that maximizing the minimum distance in space F between $\Phi(X)$ and the separating hyper plane $H(w,b)$ is a good means of reducing the generalization risk [10].

$$H(w,b) = \{ f \in F \mid < w, f >_F + b = 0 \} \tag{2}$$

Where, <> is inner product Also, it has been proved that the optimal hyper plane can be obtained solving the convex quadratic programming (QP) problem [10]:

$$\begin{aligned} &\text{Minimize} \quad \frac{1}{2}\|w\|^2 + c\sum_{i=1}^{l}\xi \\ &\text{with} \quad y_i(< w, \Phi(X) > + b) \geq 1 - \xi \quad i = 1,...,l \end{aligned} \tag{3}$$

Where constant C and slack variables x are introduced to take into account the eventual non-separability of $\Phi(X)$ into F. Practically, this criterion is softened to the minimization of a cost factor involving both the complexity of the classifier, the degree to which marginal points are misclassified, and the tradeoff between these factors through a margin of error parameter (usually designated C) which is tuned through cross-validation procedures. There are several common kernel functions that are used such as the linear, polynomial kernel, sigmoidal kernel and the most popular one, Gaussian (or "radial basis function") kernel, defined as:

$$K(x,y) = \exp\left(\frac{-|x-y|^2}{(2\sigma^2)}\right) \tag{4}$$

Where σ is a scale parameter and x, y are feature-vectors in the input space. The Gaussian kernel has two hyper parameters of C and σ to control the overall performance. In this paper we used radial basis function (RBF).

2.2. Probabilistic SVM

Given training examples $\hat{x}_i \in \Re^n, i = 1,...,m$, labeled by $\hat{y}_i \in \{+1,-1\}$, the binary Support Vector Machine (SVM) computes a decision function $f(x)$ such that $sign(f(x))$ can be used to predict the label of any test example x. Instead of predicting the label, many applications require a posterior class probability $P(y=1|x)$. Platt [9] proposes to approximate the posterior by a sigmoid function:

$$P(y=1|x) \approx P_{A,B}(x) \equiv \frac{1}{1+\exp(Af(x)+B)} \tag{5}$$

The best parameters (A,B) are then estimated by solving the following regularized maximum likelihood problem with a set of labeled examples $\{(x_i, y_i)\}_{i=1}^{l}$ (with N_+ of the y_i's positive and N_- for negative ones):

$$\min_{z=(A,B)} F(z) = -\sum_{i=1}^{l}\left(t_i \log(p_i) + (1-t_i \log(1-p_i))\right), \tag{6}$$

$$for \;\; p_i = P_{A,B}(x_i), and \;\; t_i = \begin{cases} \frac{N_+ + 1}{N_+ + 2} & if \quad y_i = +1 \\ \frac{1}{N_- + 2} & if \quad y_i = -1 \end{cases}, \qquad i = 1,...,l$$

2.3. Gaussian Mixture Models

Gaussian Mixture Models (GMM) provides a good approximation of the originally observed feature probability density functions by a mixture of weighted Gaussians. The mixture coefficients were computed using an Expectation Maximization (EM) algorithm. Each emotion is modeled in a separate GMM and decision is made on the basis of maximum likelihood model. We used diagonal covariance GMMs as baseline classifier.The outputs of GMM are:

$$P_{GMM}(x|C_i) = \sum_{m=1}^{M} c_{im} N(x, \mu_{im}, \Sigma_{im}) \tag{7}$$

Where:

$$N(x, \mu_{im}, \Sigma_{im}) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \times \exp\left[-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right] \tag{8}$$

Here, c_{im}, μ_{im} and Σ_{im} are the weight, mean and variance, respectively, of the m-th mixture for class i. The GMM reflects the intra-class information.

3. VOWEL RECOGNITION SYSTEM OVERVIEW

To make a model practical, it is necessary to develop training and recognition algorithms precisely. Our system based on two important steps, first step is vowel detection and second for vowel classification. In following we briefly describe our steps.

3.1. Vowel detection and recognition

The purpose of this step is creation of system for detection of vowels then finds the best boundaries of vowels. In fact this step is a pre-processing for classification step. Outputs of vowel detection block include two boundaries (start and end point of vowel) and average likelihood score of each vowel's segment. The basis of the suggested model is a linear fusion of estimated score of GMM's with probabilistic SVM and traditional band-pass energy for achieving better

performance and accuracy. In rest of this section we describe proposed vowel recognition system.

3.2. Soft GMM Fitting

The main idea of soft segment modeling on a phoneme recognition system is proposed in [5] and improved in [6]. In this segmentation method, considers neighbor segment's vectors in estimating each segment's probability distribution function (PDF) with suitable weight using a GMM. The importance of soft segmentation approach may come into view in the boundary estimation and the recognition phase. In the training phase, the adjacent segments are playing role in GMM parameter estimation.

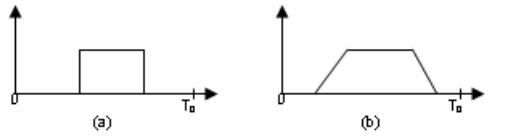


Figure 1. Soft segment modeling versus hard segment modeling. (a)Hard (b) Soft segment modeling.

We proposed to compute this score (normalized between 0 and 1), called confidence measure (CM), to indicate reliability of any recognition decision made by vowel detection system. CM can be computed for every recognized vowel to indicate how likely it is correctly recognized and how much we can trust the results for the utterance.

3.3. Probabilistic SVM Training with GMM's Output

The overall training and recognition block diagram of the developed system is depicted in figure 2. In this proposed method, we introduce a method that how we can use GMM for selecting the SVM training data set. An important weakness of SVM in large scale databases is time consuming in real time recognition because of its large numbers of support vectors.

In this case, if we use GMM confidence measure for choosing the training dataset, we achieve the best support vectors. We discuss vowel (ω_1) and non-vowel (ω_2) training system, The GMM score is the difference between the log likelihoods of the two models,

$$l(X) = \log P(X \mid \omega_1) - \log P(X \mid \omega_2) \tag{9}$$

The decision boundary is:

$$l(X) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} \log(P_1) - \log(P_2) \tag{10}$$

Where P_i is the A priori probability of ω_i. If we add ε margin for GMM score,

$$\begin{cases} X \in \omega_1, & if \quad l(X) < \log\left(\dfrac{P_1}{P_2}\right) - \varepsilon \\ X \in \omega_2, & if \quad l(X) > \log\left(\dfrac{P_1}{P_2}\right) + \varepsilon \\ X \text{ pass to SVM classifier} & if \quad otherwise \end{cases} \tag{11}$$

That ε is calculated experimentally. It is clear that the value of ε is very important for generalization characteristic of classifier.

3.4. Classification with linear combined models' outputs

In this section, we proposed a linear model for vowel recognition based on combining the outputs of soft GMM models (vowel and non-vowel classes), the probabilistic output of PSVM and band-pass energy. The overall training block diagram of the developed system is depicted in figure 1. We suggested for calculation of vowel boundaries first we must estimate $P(Vowel \mid X)$:

$$\begin{aligned} P(Vowel \mid X) &\cong (0.3)G(X) \\ &+ (0.5)P_{GMM}(X \mid Vowel) + (0.2)P_{PSVM}(X) \end{aligned} \tag{12}$$

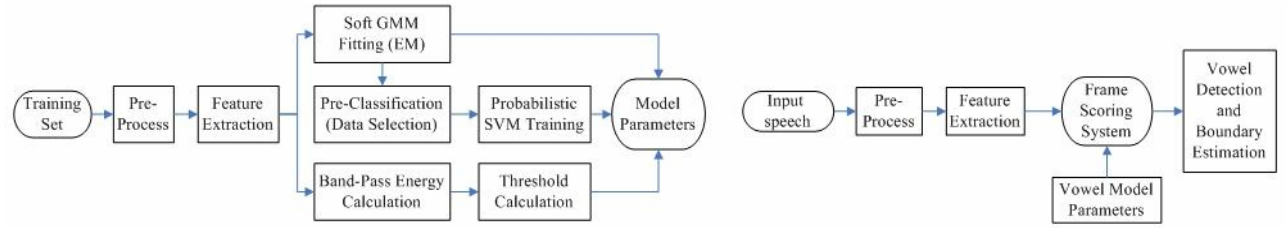
Where $P(Vowel \mid X)$ is probability of input vector X is member of vowel class, $P_{GMM}(X \mid Vowel)$ is output of soft GMM fitted to vowel class, $P_{PSVM}(X)$ is output of PSVM and $G(X)$ is band-pass energy of frame. Like vowel class we can calculate $P(NonVowel \mid X)$ for non vowel class:

$$\begin{aligned} P(NonVowel \mid X) &\cong (0.3)(1 - G(X)) \\ &+ (0.5)P_{GMM}(X \mid NonVowel) + (0.2)P_{PSVM}(X) \end{aligned} \tag{13}$$

Where $P_{GMM}(X \mid NonVowel)$ is the output of soft GMM fitted to non-vowel class. The underlying goal of classifier combination theory is to identify the conditions under which the combination of an ensemble of classifiers yields improved performance compared to the individual classifiers. We can find the vowels boundaries with contact points of $P(Vowel \mid X)$ and $P_{GMM}(X \mid NonVowel)$ curves.

4. EXPERIMENTAL RESULTS

The proposed method has been verified on a subset of clean speech data consisting of 30 male and 25 female utterances with no background noise were extracted from FARSI-DAT (most popular Persian speech database) for the-evaluation experiments. The training material consisted of 30 complete sentences for each speaker. We made vowels database by labeling vowels manually in each utterance. Our database focuses on eight important vowels in Persian language (as /a/, /@/, /o/, /e/, /i/, /u/, /au/, /ei/). We used 80 percent of our data for training and 20 percent for evaluation phase. Specifications of the speech analysis at the acoustic pre-processor are summarized as follows (in Table 1):



(a) (b)

Figure 1 – Block Diagram of (a) Training phase (b) Vowel Recognition system.

In training phase, we searched for the best number of mixtures in soft GMM model experimentally. For vowel class, GMM have been trained with 80 mixtures and for non-vowel class, with 170 mixtures. Although this may increase the computational cost, it would be ignorable in comparison with Viterbi search computational cost.

In recognition phase, the results compared by equivalent system using HMM. In overall, our method improved about 1% in averaged recognition rate. Accuracy matrix for proposed speaker independent vowel recognition is illustrated in Table 2.

Sampling Frequency	8 kHz
Pre-Emphasis	$1 - 0.98\ z^{-1}$
Hamming window width	25 ms(200 Point)
Frame period	12.5 ms(100 Point)
LPC analysis order	16-th
Feature parameters	MFCC, Delta MFCC, Delta Log-Energy

Table 1. Speech analysis conditions

		Uttered Vowel						
		/a/	**/@/**	**/o/**	**/e/**	**/i/**	**/u/**	**/au/**
Recognized Vowel	**/a/**	94.9	2.1	0.3	0.5	0.1	0.1	0.1
	/@/	1.4	95.2	0.1	0.8	0.3	0.4	0.2
	/O/	0.8	0.4	96.1	3.7	0.4	0.2	0.4
	/e/	0.4	0.5	2.1	93.2	0.3	0.1	0.1
	/i/	0.5	1.1	0.3	0.4	95.1	1.1	1.4
	/u/	0.9	0.4	0.7	0.6	2.9	96.1	0.6
	/au/	1.1	0.3	0.4	0.8	0.9	2.1	97.2

Table 2. Accuracy matrix for Vowel Recognition system

5. DISCUSSIONS AND CONCLUSIONS

A simple and efficient statistical Vowel Recognition method has been introduced in this paper. This method improved accuracy of vowel recognition with combining GMM, SVM and Band-pass Energy. The main feature of this model is the toleration of gradual inter-segmental conversion. The model is very promising in both recognition rate and computational complexity aspects. The proposed method has the ability to reduce support vectors significantly. This reduction leads us to improve the speed of SVM classifier also using the GMM help us achieving more accuracy. The main advantage of this model is a drastic reduction of recognition time. The remained open problems are the soft window shape, fast methods for both GMM recognition and training, and the coefficients of each combined classifiers (e.g. SVM, GMM and Band-Pass Energy) on this duration modeling approach, which their studies are all in progress now.

6. REFERENCES

[1] Rabiner, L., Juang, B. “Fundamentals of Speech Recognition” Prentice Hall, Englewood Cliffs, NJ (1993).

[2] Ch. Sekhar, K. Takeda, and F. Itakura, “Recognition of Consonant-Vowel (CV) Units of Speech in a Broadcast News Corpus Using Support Vector Machines” in Pattern Recognition with SVM, pp. 283–291, Canada, Springer, 2002.

[3] B. Zhao, T. Schultz, "Toward Robust Parametric Trajectory Segmental Model for Vowel Recognition", ICASSP, Vol. 4, 2002.

[4] A. Ganapathiraju, J. Hamaker, J. Picone, M. Ordowski, and G. R. Doddington. "Syllable-based large vocabulary continuous speech recognition", IEEE Transactions on Speech and Audio Processing, 9(4):358–366, May 2001.

[5] M. Ostendorf, V. Digalakis, O. Kimball, “From HMM To Segment Models: A Unified View of Stochastic Modeling of Speech Recognition”, IEEE Transactions on Speech and Audio Processing, Vol. 4, No. 5, PP. 360-378, Sep. 1996.

[6] Razzazi, F., Sayadiyan, A., “Evaluation of soft segment modeling on a phoneme recognition system”, in proc. ICECS03. Vol. 1, pp. 140-143, Dec. 2003.

[7] F. Hou, B. Wang. “Text-independent Speaker recognition using probabilistic SVM with GMM adjasment”, in Proc. NLP-KE conf., pp. 305-308, 2003.

[8] S. Fine, J. Navratil and R. A. Gopinath. Hybrid GMM/SVM Approach to Speaker Identification, Proc. ICASSP 2001.

[9] Platt, J., “Probabilistic outputs for support vector machines and comparison to regularized likelihood methods” In: A. Smola, P. Bartlett, B. Scholkopf, and D. Schuurmans (eds.): “Advances in Large Margin Classiffiers”, Cambridge, MA, 2000.

[10] V.N. Vapnik. The nature of statistical learning theory, Second Edition, New York: Springer-Verlag, 1999.

FACE DETECTION USING ADABOOSTED SVM-BASED COMPONENT CLASSIFIER

Seyyed Majid Valiollahzadeh, Abolghasem Sayadiyan
Electrical Engineering Department, Amirkabir University of Technology,
Tehran, Iran, 15914
valiollahzadeh@yahoo.com, eea35@aut.ac.ir

Mohammad Nazari
Electrical Engineering Department, Amirkabir University of Technology,
Tehran, Iran, 15914
mohnazari@aut.ac.ir

Keywords: Face Detection, Cascaded Classifiers, ComponentLearn, Adaboost, Support Vector Machine (SVM).

Abstract: Recently, Adaboost has been widely used to improve the accuracy of any given learning algorithm. In this paper we focus on designing an algorithm to employ combination of Adaboost with Support Vector Machine (SVM) as weak component classifiers to be used in Face Detection Task. To obtain a set of effective SVM-weaklearner Classifier, this algorithm adaptively adjusts the kernel parameter in SVM instead of using a fixed one. Proposed combination outperforms in generalization in comparison with SVM on imbalanced classification problem. The proposed here method is compared, in terms of classification accuracy, to other commonly used Adaboost methods, such as Decision Trees and Neural Networks, on CMU+MIT face database. Results indicate that the performance of the proposed method is overall superior to previous Adaboost approaches.

1 INTRODUCTION

Nonlinear classification of data is always of special attention. Face Detection is a problem dealing with such data, due to large amount of variation and complexity brought about by changes in facial appearance, lighting and expression. Feature selection is needed beside appropriate classifier design to solve this problem, like many other pattern recognition tasks.

One of the major developments in machine learning in the past decade is the Ensemble method, which finds a highly accurate classifier by combining many moderately accurate component classifiers. Two of the commonly used techniques for constructing Ensemble classifiers are Boosting [schaphire, 2002] and Bagging [Breiman, 1996]. In Comparison with Bagging, Boosting outperforms when the data do not have much noise [Opitz, 1999] [Bauer, 1999] .Among popular Boosting methods, AdaBoost [Freund, 1997] establishes a collection of weak component classifiers by maintaining a set of weights over training samples and adjusting them adaptively after each Boosting iteration: the weights of the misclassified samples by current component classifier will be increased while the weights of the correctly classified samples will be decreased. To implement the weight updates in Adaboost, several algorithms have been proposed [Kuncheva, 2002]. The success of AdaBoost can be attributed to its ability to enlarge the margin [schapire, 1998], which could enhance AdaBoost's generalization capability. Decision Trees [Dietterich, 2000] or Neural Networks [Schwenk, 2000] [Ratsch, 2001] have already been employed as component classifiers for AdaBoost. These studies showed good generalization performance of these AdaBoost. However, determining the suitable tree size is a question when Decision Trees are used as component classifiers. Also, controlling the complexity in order to avoid over fitting will remain a question, when Radial Basis Function (RBF) Neural Networks are used as component classifiers.

Moreover, we have to decide on the optimum number of centers and also on setting the width values of the RBFs. All these factors have to be carefully tuned in practical use of AdaBoost. Furthermore, diversity is known to be an important factor which affects the generalization accuracy of Ensemble classifiers [Melville, 2005][Kuncheva, 2002]. In order to quantify the diversity, some methods are proposed [Kuncheva, 2003] [Windeatt, 2005]. It is also known that in AdaBoost exists an accuracy/diversity dilemma [Dietterich, 2000], which means that the more accurate two component classifiers become, the less they can disagree with each other. Only when the accuracy and diversity are well balanced, can the AdaBoost demonstrate excellent generalization performance. However, the existing AdaBoost algorithms do not yet explicitly taken sufficient measurement to deal with this problem. Support Vector Machine [Vapnick, 1998] was developed based on the theory of Structural Risk Minimization. By using a kernel trick to map the training samples from an input space to a high dimensional feature space, SVM finds an optimal separating hyper plane in the feature space and uses a regularization parameter, C, to control its model complexity and training error. One of the popular kernels used by SVM is the RBF kernel, including a parameter known as Gaussian width, σ. In contrast to the RBF networks, SVM with the RBF kernel (RBFSVM in short) can automatically determine the number and location of the centers and the weight values [Scholkopf, 1997]. Also, it can effectively avoid over fitting by selecting proper values of C and σ. From the performance analysis of RBFSVM [Valentini, 2004], we know that σ is a more important parameter compared to C: although RBFSVM cannot learn well when a very low value of C is used, its performance largely depends on the σ value if a roughly suitable C is given. This means that, over a range of suitable C, the performance of RBFSVM can be conveniently changed by simply adjusting the value of σ.

The proposed here method is compared, in terms of classification accuracy, to other commonly used Adaboost methods, such as Decision Trees and Neural Networks, on CMU+MIT face database. Results indicate that the performance of the proposed method is overall superior to those of traditional adaboost approaches.

2 FEATURE SELECTION

In this paper, like Viola and Jones [Viola and Jones 2001], we use four types of Haar-like basis functions for feature selection which have been used by Papageorgiou et al [Papageorgiou et al 1998].

Like their work, we use four types of haar-like feature to build the feature pool. The feature can be computed efficiently with integral image. The main objective to use these features is that they can be rescaled easily which avoids to calculate a pyramid of images and yields to fast operation of the system on these features. These features can be seen in figure1. Given that the base resolution of the detector is 32x32, the exhaustive set of rectangle features is quite large, over 180,000. Note that unlike the Haar basis, the set of rectangle features is overcomplete. For each scale level, we rescale the features and record the relative coordinate of the rescaled features to the top-left of integral image in look-up-table (LUT). After looking up the value of the rescaled rectangle's coordinate, we calculate features with relative coordinate. Like viola, we use image variance σ to correct lighting, which can be got using integral images of both original image and image squared. Rescaling needs to round rescaled coordinates to nearest integer, which would degrade the performance of viola's features [Lienhart 2003]. Like R. Lienhart [Lienhart 2003], we normalize the features by acreage, and thus reduce the rounding error.

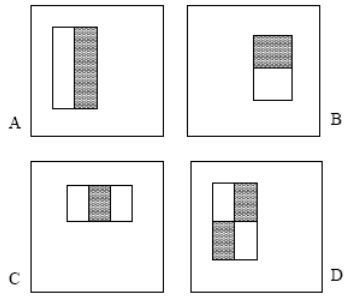


Figure 1: Example rectangle features shown relative to the enclosing detection window. The sum of the pixels which lie within the white rectangles is subtracted from the sum of pixels in the grey rectangles. Two-rectangle features are shown in (A) and (B). Figure (C) shows a three-rectangle feature, and (D) a four-rectangle feature.

Using the integral image any rectangular sum can be computed in four array references (see Figure 2). Clearly the difference between two rectangular sums can be computed in eight references. Since the two-rectangle features defined above involve adjacent rectangular sums they can be computed in

six array references, eight in the case of the three-rectangle features, and nine for four-rectangle features.

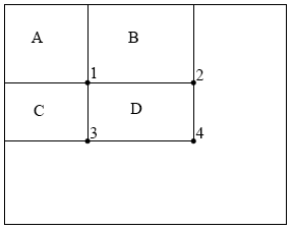


Figure 2: The sum of the pixels within rectangle D can be computed with four array references. The value of the integral image at location 1 is the sum of the pixels in rectangle A. The value at location 2 is A+B, at location 3 is A+C, and at location 4 is A+B+C+D. The sum within D can be computed as 4+1-(2+3).

3 STATISTICAL LEARNING

In this section, we describe boost based learning methods to construct face/nonface classifier, and propose a new boosting algorithm which improves boosting learning.

3.1 AdaBoost Learning

Given a set of training samples, AdaBoost [Schapire and Singer 1999] maintains a probability distribution, W, over these samples. This distribution is initially uniform. Then, AdaBoost algorithm calls Weak Learn algorithm repeatedly in a series of cycles. At cycle T, AdaBoost provides training samples with a distribution w^t to the WeakLearn algorithm.

AdaBoost, constructs a composite classifier by sequentially training classifiers while putting more and more emphasis on certain patterns.

For two class problems, we are given a set of N labeled training examples $(y_1, x_1),...,(y_N, x_N)$, where $y_i \in \{+1,-1\}$ is the class label associated with example x_i.

For face detection, x_i is an image sub-window of a fixed size (for our system 24x24) containing an instance of the face $(y_i = +1)$ or non-face $(y_i = -1)$ pattern. In the notion of AdaBoost see Algorithm 1, a stronger classifier is a linear combination of M weak classifiers.

In boosting learning [9, 26, 10], each example x_i is associated with a weight w_i, and the weights are updated dynamically using a multiplicative rule according to the errors in previous learning so that more emphasis is placed on those examples which are erroneously classified by the weak classifiers learned previously.

Greater weights are given to weak learners with lower errors. The important theoretical property of AdaBoost is that if the weak learners consistently have accuracy only slightly better than half, then the error of the final hypothesis drops to zero exponentially fast. This means that the weak learners need be only slightly better than random.

Furthermore, since proposed AdaBoost with SVM invents a convenient way to control the classification accuracy of each weak learner, it also provides an opportunity to deal with the well-known accuracy/diversity dilemma in Boosting methods. This is a happy accident from the investigation of AdaBoost based on SVM weak learners.

Algorithm 1. The AdaBoosAlgorithm [Schapire and Singer] .

1. Input: Training sample
Input: a set of training samples with labels $(y_1, x_1),...,(y_N, x_N)$, ComponentLearn algorithm, the number of cycles *T*.

2. Initialize: the weights of training samples: $w_i^1 = 1/N$, for all $i = 1,..., N$

3. Do for $t = 1,...,T$
(1)Use ComponentLearn algorithm to train the component classifier h_t on the weighted training sample set.
(2)Calculate the training error of h_t:

$$\varepsilon_t = \sum_{1=1}^{N} w_i^t, y_i \neq h_t(x_i).$$

(3)Set weight of component classifier h_t:

$$h_t : \alpha_t = \frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$$

(4)Update the weights of training samples:

$$w_i^{t+1} = \frac{w_i^t \exp\{-\alpha_t y_i h_t(x_i)\}}{C_t}$$

where C_t is a normalization constant, and

$$\sum_{i=1}^{N} w_i^{t+1} = 1$$

4. Output: $f(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$.

3.2 SVM Based Approach for Classification

The principle of Support Vector Machine (SVM) relies on a linear separation in a high dimension feature space where the data have been previously

mapped, in order to take into account the eventual non-linearities of the problem.

If we assume that, the training set $X = (x_i)_{i=1}^{l} \subset \mathrm{R}^R$ where l is the number of training vectors, R stands for the real line and R is the number of modalities, is labelled with two class targets $Y = (y_i)_{i=1}^{l}$, where :

$$y_i \in \{-1,+1\} \quad \Phi : \mathrm{R}^R \to F \tag{1}$$

Maps the data into a feature space F. Vapnik has proved that maximizing the minimum distance in space F between $\Phi(X)$ and the separating hyper plane $H(w,b)$ is a good means of reducing the generalization risk.

Where:

$$H(w,b) = \{f \in F \mid < w, f >_F + b = 0\}, \quad (<> \textit{is} \text{ inner product}) \tag{2}$$

Vapnik also proved that the optimal hyper plane can be obtained solving the convex quadratic programming (QP) problem:

$$\text{Minimize} \quad \frac{1}{2}\|w\|^2 + c\sum_{i=1}^{l}\xi_i \qquad \text{with} \quad y_i(< w, \Phi(X) > + b) \geq 1 - \xi_i \quad i = 1,...,l \tag{3}$$

Where constant C and slack variables x are introduced to take into account the eventual non-separability of $\Phi(X)$ into F.

In practice this criterion is softened to the minimization of a cost factor involving both the complexity of the classifier and the degree to which marginal points are misclassified, and the tradeoff between these factors is managed through a margin of error parameter (usually designated C) which is tuned through cross-validation procedures.

Although the SVM is based upon a linear discriminator, it is not restricted to making linear hypotheses. Non-linear decisions are made possible by a non-linear mapping of the data to a higher dimensional space. The phenomenon is analogous to folding a flat sheet of paper into any three-dimensional shape and then cutting it into two halves, the resultant non-linear boundary in the two-dimensional space is revealed by unfolding the pieces.

The SVM's non-parametric mathematical formulation allows these transformations to be applied efficiently and implicitly: the SVM's objective is a function of the dot product between pairs of vectors; the substitution of the original dot products with those computed in another space eliminates the need to transform the original data points explicitly to the higher space. The computation of dot products between vectors without explicitly mapping to another space is performed by a kernel function.

The nonlinear projection of the data is performed by this kernel functions. There are several common kernel functions that are used such as the linear, polynomial kernel $(K(x,y) = (< x, y >_{R^R} + 1)^d$ and the sigmoidal kernel $(K(x,y) = \tanh(< x, y >_{R^R} + a))$, where x and y are feature vectors in the input space.

The other popular kernel is the Gaussian (or "radial basis function") kernel, defined as:

$$K(x,y) = \exp\left(\frac{-|x-y|^2}{(2\sigma^2)}\right) \tag{4}$$

Where σ is a scale parameter, and x and y are feature-vectors in the input space. The Gaussian kernel has two hyper parameters to control performance C and the scale parameter σ. In this paper we used radial basis function (RBF).

3.3 AdaBoosted SVM-Based Component Classifier

We combine SVM with AdaBoost to improve its capability in classification. When applying Boosting method to strong component classifiers, these classifiers must be appropriately weakened in order to benefit from Boosting [Dietterich 2000].

Like Schapire and Singer, we used resampling to train AdaBoost, in this problem we must train weak classifiers (SVM classifier) to obtain best Gaussian width, σ and the regularization parameter, C, for optimizing strong classifier (AdaBoost classifier).

Hence, SVM with RBF kernel is used as weak learner for AdaBoost, a relatively large σ value, which corresponds to a SVM with RBF kernel with relatively weak learning ability, is preferred. Both resampling and reweighting can be used to train AdaBoost. The algorithm is shown in the following diagram.

Algorithm 2. The AdaBoost with SVM Algorithm.

1. Input: Training sample
Input: a set of training samples with labels $(y_1, x_1),...,(y_N, x_N)$,
The initial $\sigma = \sigma_{ini}$, $\sigma_{\min}$, σ_{step}

2. Initialize: the weights of training samples: $w_i^1 = 1/N$, for all $i = 1,..., N$

3. Do while $\sigma > \sigma_{\min}$
(1)Use RBFSVM to train on the weighted training sample set.
(2)Calculate the training error of h_t :

$$\varepsilon_t = \sum_{1=1}^{N} w_i^t, y_i \neq h_t(x_i).$$

(3) if $\varepsilon_t > .5$,decrease σ value by σ_{step} and goto(1)
(4)Set weight of component classifier h_t :

$$h_t : \alpha_t = \frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$$

(5)Update the weights of training samples:

$$w_i^{t+1} = \frac{w_i^t \exp\{-\alpha_t y_i h_t(x_i)\}}{C_t}$$

where C_t is a normalization constant, and

$$\sum_{i=1}^{N} w_i^{t+1} = 1$$

4. Output: $f(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$.

4 EXPERIMENTAL RESUTTS

4.1 Database

We tested our system on the MIT+CMU frontal face test set [Rowley et al. 1994] and own database. There are more than 2,500 faces in total. To train the detector, a set of face and nonface training images were used. The pairwise recognition framework is evaluated on a compound face database with 2000 face images hand labelled faces scaled and aligned to a base resolution 32 by 32 pixels by the centre point of the two eyes and the horizontal distance between the two eyes. For non-face training set, an initial 10,000 non-face samples were selected randomly from 15,000 large images which contain no face.

4.2 Face Detection System

We explain our face detection system and show how to construct a AdaBoosted SVM-based component classifier for face detection. The learning of a detector is done as follows:

1. A set of simple Haar wavelet features are used as candidate features. There are tens of thousands of such features for a 32x32 window.
2. A subset of them are selected and the corresponding weak classifiers are constructed, using AdaBoosted SVM-based component classifier learning.
3. A strong classifier is constructed as a linear combination of the weak ones.
4. A detector is composed of one or several strong classifiers in cascade.

The detector pyramid is then built upon the learned detectors [Li and Zhang 2004].

4.3 Results

The SVM-based component classifier and AdaBoost algorithm are used for the classification of each pair of individuals. We compare the detection rates to other commonly used Adaboost methods, such as Decision Trees and Neural Networks, on face database.

For showing the performance of our AdaBoosted svm-based component classifier algorithm, the results are shown in Table 1.

False detections / Detector	120	200
Adaboost with SVM	5.41	1.85
Adaboost with Decision Trees	9.81	2.42
Adaboost with Neural Networks	14.51	5.41

Table 1: Comparison of Error rate (%) for some AdaBoost methods.

A ROC curve showing the performance of our detector on this test set is shown in Figure 3 and Some results are shown in Figure 4.

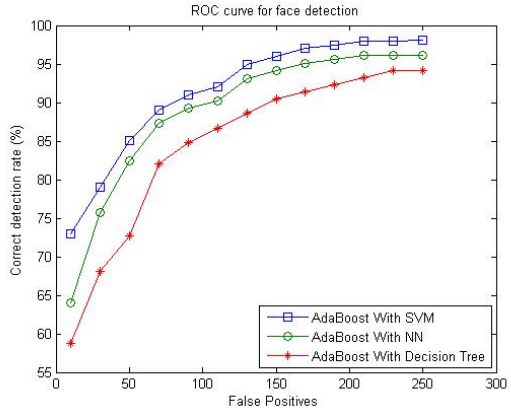


Figure 3: Comparison of ROC for frontal face detection results.

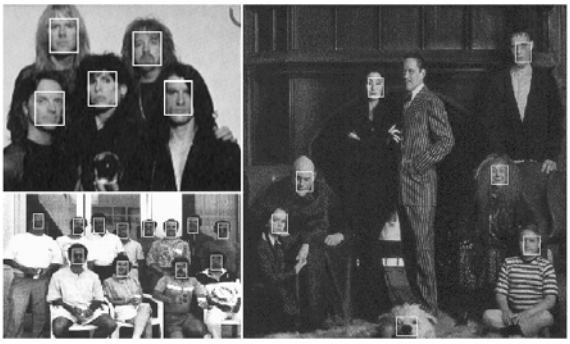


Figure 4: Some frontal face detection results.

5 CONCLUSIONS

AdaBoost with properly designed SVM-based component classifiers is proposed in this paper, which is achieved by adaptively adjusting the kernel parameter to get a set of effective component classifiers. Experimental results on CMU+MIT database for Face Detection demonstrated that proposed AdaBoostSVM algorithm performs better than other approaches of using component classifiers such as Decision Trees and Neural Networks in accuracy and speed. Besides these, it is found that proposed AdaBoostSVM algorithm demonstrated good performance on imbalanced classification problems. Based on the AdaBoostSVM, an improved version is further developed to deal with the accuracy/diversity dilemma in Boosting algorithms, giving rising to better generalization performance. Experimental results indicate that the performance of the cascaded adaboost classifier with SVM is overall superior to those obtained by the NN and Decision Tree.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the Iran Telecommunication Research Center (ITRC) for financially supporting this work.

REFERENCES

Schapire, R. E., Freund, Y., October 1998, *Boosting the margin: a new explanation for the effectiveness of voting methods. The Annals of Statistics, 26(5):1651–1686.*

Freund, Y., Schapire, R., Aug 1997 *"A decision-theoretic generalization of on-line learning and an application to boosting". Journal of Computer and System Sciences, 55(1):119–139.*

Schapire R. E., Y. Singer, Dec 1999 *"Improved boosting algorithms using confidence-rated predictions. Machine Learning, 37(3):297–336.*

Friedman, J., Hastie, T., and R. Tibshirani, July 1998. *"Additive logistic regression: a statistical view of boosting". Technical report, Department of Statistics, Sequoia Hall, Stanford Univerity*

Dietterich, T. G., Aug 2000 *"An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization," Machine Learning, vol. 40, no. 2, pp. 139–157.*

Papageorgiou, C., Oren, M., Poggio, T. , 1998, *A general framework for object detection. In International Conference on Computer Vision.*

Viola, P., Jones, M., Dec. 2001, *"Rapid Object Detection Using a Boosted Cascade of Simple Features," Proc. IEEE CS Conf. Computer Vision and Pattern Recognition.*

Rowley, H., Baluja, S., Kanade, T.,1998, *Neural network-based face detection. In IEEE Patt. Anal. Mach. Intell., volume 20, pages 22–38,.*

Li, S. Z.,EE, Zhang, Z. Q. , sept. 2004 *"FloatBoost Learning and Statistical Face Detection" In IEEE Patt. Anal. Mach. Intell., vol. 26, no. 9.*

Haykin, S., July 1998 ,*Neural networks : A comprehensive foundation. Prentice Hall.*

Lienhart, R., Kuranov, A.,and Pisarevsky , V., 2003. *"Empirical analysis of detection cascades of boosted classifiers for rapid object detection"*
.

schapire. R. E., 2002, *The boosting approach to machine learning: An overview. In MSRI Workshop on Nonlinear Estimation and Classification.*

Breiman. L. , 1996, *Bagging predictors. Machine Learning, 24:123–140.*

Opitz, D. and Maclin, R., 1999, *Popular ensemble methods: An empirical study. Journal of Artificial Intelligence Research, 11:169–198.*

Bauer ,E. and Kohavi ,R., Jul 1999, *An empirical comparison of voting classification algorithms: Bagging, boosting, and variants.Machine Learning, 36(1):105–139*

Kuncheva, L. I. and Whitaker, C. J. , 2002, *Using diversity with three variants of boosting: aggressive,conservative, and inverse.In Proceedings of the Third International Workshop on Multiple Classifier Systems.*

Schwenk , H. and Bengio. Y, 2000, *Boosting neural networks. Nueral Computation, 12:1869–1887.*

Melville P. and Mooney. R. J , Mar2005. *Creating diversity in ensembles using artificial data. Information Fusion, 6(1):99–111*

Face Recognition with Kernel Direct Discriminant Analysis and SVM Combined Method

Seyyed Majid valiollahzadeh, Abolghasem Sayadiyan, Mohammad Nazari

Electrical Engineering Department, Amirkabir University of Technology, Tehran, Iran, 15914
valiollahzadeh@yahoo.com, {mohnazari, eea35} @aut.ac.ir

Abstract: *Applications such as Face Recognition (FR) that deal with high-dimensional data need a mapping technique that introduces representation of low-dimensional features with enhanced discriminatory power and a proper classifier, able to classify those complex features .Most of traditional linear discriminant analysis (LDA) suffer from the disadvantage that their optimality criteria are not directly related to the classification ability of the obtained feature representation. Moreover, their classification accuracy is affected by the "small sample size" (SSS) problem which is often encountered in FR tasks. In this short paper, we combine nonlinear kernel based mapping of data called KDDA with Support Vector machine (SVM) classifier to deal with both of the shortcomings in an efficient and cost effective manner. The proposed here method is compared, in terms of classification accuracy, to other commonly used FR methods on UMIST face database. Results indicate that the performance of the proposed method is overall superior to those of traditional FR approaches, such as the Eigenfaces, Fisherfaces, and D-LDA methods and traditional linear classifiers.*

Keywords: Face Recognition, Kernel Direct Discriminant Analysis (KDDA), small sample size problem (SSS), Support Vector Machine (SVM).

1 Introduction

Selecting appropriate features to represent faces and proper classification of these features are two central issues to face recognition (FR) systems. For feature selection, successful solutions seem to be appearance-based approaches, (see [3], [2] for a survey), which directly operate on images or appearances of face objects and process the images as two-dimensional (2-D) holistic patterns, to avoid difficulties associated with Three-dimensional (3-D) modelling, and shape or landmark detection [2]. For the purpose of data reduction and feature extraction in the appearance-based approaches, Principle component analysis (PCA) and linear discriminant analysis (LDA) are introduced as two powerful tools. Eigenfaces [4] and Fisherfaces [5] built on the two techniques, respectively, are two state-of-the-art FR methods, proved to be very successful. It is generally believed that, LDA based algorithms outperform PCA based ones in solving problems of pattern classification, since the former optimizes the low-dimensional representation of the objects with focus on the most discriminant feature extraction while the latter achieves simply object reconstruction. However, many LDA based algorithms suffer from the so-called "small sample size problem" (SSS) which exists in high-dimensional pattern recognition tasks where the number of available samples is smaller than the dimensionality of the samples. The traditional solution to the SSS problem is to utilize PCA concepts in conjunction with LDA (PCA+LDA) as it was done for example in Fisherfaces [11]. Recently, more effective solutions, called Direct LDA (D-LDA) methods, have been presented [12], [13]. Although successful in many cases, linear methods fail to deliver good performance when face patterns are subject to large variations in

viewpoints, which results in a highly non-convex and complex distribution. The limited success of these methods should be attributed to their linear nature [14]. Kernel discriminant analysis algorithm, (KDDA) generalizes the strengths of the recently presented D-LDA [1] and the kernel techniques while at the same time overcomes many of their shortcomings and limitations.

In this work, we first nonlinearly map the original input space to an implicit high-dimensional feature space, where the distribution of face patterns is hoped to be linearized and simplified. Then, KDDA method is introduced to effectively solve the SSS problem and derive a set of optimal discriminant basis vectors in the feature space. And then SVM approach is used for classification.

The rest of the paper is organized as follows. In Section tow, we start the analysis by briefly reviewing KDDA method. Following that in section three, SVM is introduced and analyzed as a powerful classifier. In Section four, a set of experiments are presented to demonstrate the effectiveness of the KDDA algorithm together with SVM classifier on highly nonlinear, highly complex face pattern distributions. The proposed method is compared, in terms of the classification error rate performance, to KPCA, GDA and KDDA algorithm with nearest neighbour classifier on the multi-view UMIST face database. Conclusions are summarized in Section five.

2 Kernel Direct Discriminant Analysis (KDDA)

2.1. Linear Discriminant Analysis

In the statistical pattern recognition tasks, the problem of feature extraction can be stated as follows: Assume that we have a training set, $\{Z_i\}_{i=1}^{L}$ is available. Each image is defined as a vector of length $N(=I_w \times I_h)$, i.e. $Z_i \in \Re^N$ where $I_w \times I_h$ is the face image size and $\Re^N$ denotes a N-dimensional real space [1].

It is further assumed that each image belongs to one of C classes $\{Z_i\}_{i=1}^{C}$. The objective is to find a transformation φ, based on optimization of certain separability criteria, which produces a mapping $y_i = \varphi(Z_i)$, with $y_i \in \Re^N$ that leads to an enhanced separability of different face objects.

Let S_{BTW} and S_{WTH} be the between- and within-class scatter matrices in the feature space F respectively, expressed as follows:

$$S_{BTW} = \frac{1}{L}\sum_{i=1}^{C} C_i (\bar{\phi}_i - \bar{\phi})(\bar{\phi}_i - \bar{\phi})^T \quad (1)$$

$$S_{WTH} = \frac{1}{L}\sum_{i=1}^{C}\sum_{j=1}^{C_i} (\phi_{ij} - \bar{\phi}_i)(\phi_{ij} - \bar{\phi}_i)^T \quad (2)$$

Where $\phi_{ij} = \phi(Z_{ij})$, $\bar{\phi}_i = \frac{1}{C_i}\sum_{j=1}^{C_i} \phi(Z_{ij})$ is the mean of class Z_i and $\bar{\phi} = \frac{1}{L}\sum_{i=1}^{C}\sum_{j=1}^{C_i} \phi(Z_{ij})$ is the average of the ensemble.

The maximization can be achieved by solving the following eigenvalue problem:

$$\Phi = \arg\max_{\Phi} \frac{\left|\Phi^T S_{BTW}\, \Phi\right|}{\left|\Phi^T S_{WTH}\, \Phi\right|} \quad (3)$$

The feature space F could be considered as a "linearization space" [6], however, its dimensionality could be arbitrarily large, and possibly infinite. Solving this problem lead us to LDA[1].

Assuming that is S_{WTH} nonsingular and Φ the basis vectors correspond to the M first eigenvectors with the largest eigenvalues of the discriminant criterion:

$$J = tr(S_{WTH}^{-1} S_{BtW} \Phi) \quad (4)$$

The M-dimensional representation is then obtained by projecting the original face images onto the subspace spanned by the eigenvectors.

2.2. Kernel Direct Discriminant Analysis (KDDA)

The maximization process in (3) is not directly linked to the classification error which is the criterion of performance used to measure the success of the FR procedure. Modified versions of the method, such as the Direct LDA (D-LDA) approach, use a weighting function in the input space, to penalize those classes that are close and can potentially lead to misclassifications in the output space.

Most LDA based algorithms including Fisherfaces [7] and D-LDA [9] utilize the conventional Fisher's criterion denoted by (3).

The introduction of the kernel function allows us to avoid the explicit evaluation of the mapping. Any function satisfying Mercer's condition can be used as a kernel, and typical kernel functions include polynomial function, radial basis function (RBF) and multi-layer perceptrons [10].

$$\Phi = \arg\max_{\Phi} \frac{\left|\Phi^T S_{BTW}\, \Phi\right|}{\left|\left(\Phi^T S_{BTW}\, \Phi\right) + \left(\Phi^T S_{WTH}\, \Phi\right)\right|} \quad (4)$$

The KDDA method implements an improved D-LDA in a high-dimensional feature space using a kernel approach.

KDDA introduces a nonlinear mapping from the input space to an implicit high dimensional feature space, where the nonlinear and complex distribution of patterns in the input space is "linearized" and "simplified" so that conventional LDA can be applied and it effectively solves the small sample size (SSS) problem in the high-dimensional feature space by employing an improved D-LDA algorithm.

Unlike the original D-LDA method of [10] zero eigenvalues of the within-class scatter matrix are never used as divisors in the improved one. In this way, the optimal discriminant features can be exactly extracted from both of inside and outside of S_{WTH} 's null space.

In GDA, to remove the null space of SWTH, it is required to compute the pseudo inverse of the kernel matrix K, which could be extremely ill-conditioned when certain kernels or kernel parameters are used. Pseudo inversion is based on inversion of the nonzero eigenvalues.

3 SVM Based Approach for Classification

The principle of Support Vector Machine (SVM) relies on a linear separation in a high dimension feature space where the data have been previously mapped, in order to take into account the eventual non-linearities of the problem.

3.1. Support Vector Machines (SVM)

If we assume that, the training set $X=(x_i)_{i=1}^{l} \subset \mathrm{R}^{R}$ where l is the number of training vectors, R stands for the real line and R is the number of modalities, is labelled with two class targets $Y=(y_i)_{i=1}^{l}$, where :

$$y_i \in \{-1,+1\} \quad \Phi : \mathrm{R}^{R} \rightarrow F \tag{5}$$

Maps the data into a feature space F. Vapnik has proved that maximizing the minimum distance in space F between $\Phi(X)$ and the separating hyper plane $H(w,b)$ is a good means of reducing the generalization risk. Where:

$$H(w,b)=\{f \in F \mid < w, f > F + b = 0\}, \tag{6}$$

($<>$ *is* inner product)

Vapnik also proved that the optimal hyper plane can be obtained solving the convex quadratic programming (QP) problem:

$$\text{Minimize} \quad \frac{1}{2}\|w\|^2 + c\sum_{i=1}^{l}\xi \tag{7}$$

$$\text{with} \quad y_i(< w, \Phi(X) > + b) \geq 1 - \xi \quad i = 1,...,l$$

Where constant C and slack variables x are introduced to take into account the eventual non-separability of $\Phi(X)$ into F.

In practice this criterion is softened to the minimization of a cost factor involving both the complexity of the classifier and the degree to which marginal points are misclassified, and the tradeoff between these factors is managed through a margin of error parameter (usually designated C) which is tuned through cross-validation procedures.

Although the SVM is based upon a linear discriminator, it is not restricted to making linear hypotheses. Non-linear decisions are made possible by a non-linear mapping of the data to a higher dimensional space. The phenomenon is analogous to folding a flat sheet of paper into any three-dimensional shape and then cutting it into two halves, the resultant non-linear boundary in the two-dimensional space is revealed by unfolding the pieces.

The SVM's non-parametric mathematical formulation allows these transformations to be applied efficiently and implicitly: the SVM's objective is a function of the dot product between pairs of vectors; the substitution of the original dot products with those computed in another space eliminates the need to transform the original data points explicitly to the higher space. The computation of dot products between vectors without explicitly mapping to another space is performed by a kernel function.

The nonlinear projection of the data is performed by this kernel functions. There are several common kernel functions that are used such as the linear, polynomial kernel $(K(x,y)=(sx.y+a)^{d})$ and the sigmoidal kernel $(K(x,y)=\tanh(sx.y+a))$, where x and y are feature vectors in the input space.

The other popular kernel is the Gaussian (or "radial basis function") kernel, defined as:

$$K(x,y)=\exp\left(\frac{-|x-y|^2}{(2\sigma^2)}\right) \tag{8}$$

Where σ is a scale parameter, and x and y are feature-vectors in the input space. The Gaussian kernel has two hyper parameters to control performance C and the scale parameter σ. In this paper we used radial basis function (RBF).

3.2. Multi-class SVM

The standard Support Vector Machines (SVM) is designed for dichotomic classification problem (two classes only, called also binary classification). Several different schemes can be applied to the basic SVM algorithm to handle the K-class pattern classification problem. These schemes will be discussed in this section. The K-class pattern classification problem is posted as follow:

- Given l i.i.d. sample: $(x_1, y_1),...,(x_l, y_l)$ where x_i, for $i = 1,...,l$ is a feature vector of length d and $y_i = \{1,...,k\}$ is the class label for data point x_i.
- Find a classifier with the decision function, $f(x)$ such that $y = f(x)$ where y is the class label for x.

The multi-class classification problem is commonly solved by decomposition to several binary problems for which the standard SVM can be used.

For solving the multi-class problem are as listed below:

- Using K one-to-rest classifiers (one-against-all)
- Using $k(k-1)/2$ pair wise classifiers
- Extending the formulation of SVM to support the k-class problem.

3.2.1. Combination of one-to-rest classifiers

This scheme is the simplest, and it does give reasonable results. K classifiers will be constructed, one for each class. The K-th classifier will be trained to classify the training data of class k against all other training data. The decision function for each of the classifier will be combined to give the final classification decision on the K-class classification problem. In this case the classification problem to k classes is decomposed to k dichotomy decisions $f_m(x), m \in K = 1,...,k$ where the rule $f_m(x)$ separates training data of the m-th class from the other training patterns. The classification of a pattern x is performed according to maximal value of functions $f_m(x)$, $m \in K$, $K = 1,...,k$ i.e. the label of x is computed as:

$$f(x) = \arg(\max_{m \in k}(f_m(x)) \quad (9)$$

3.2.2. Pair wise Coupling classifiers

The schemes require a binary classifier for each possible pair of classes. The decision function of the SVM classifier for y_1-to-y_2 and y_2-to-y_1 has reflectional symmetry in the zero planes. Hence only one of these pairs of classifier is needed. The total number of classifiers for a K-class problem will then be $k(k-1)/2$. The training data for each classifier is a subset of the available training data, and it will only contain the data for the two involved classes. The data will be reliable accordingly, i.e. one will be labeled as +1 while the other as -1. These classifiers will now be combined with some voting scheme to give the final classification results. The voting schemes need the pair wise probability, i.e. the probability of x belong to class i given that it can be only belong to class i or j.

The output value of the decision function of an SVM is not an estimate of the p.d.f. of a class or the pair wise probability. One way to estimate the required information from the output of the SVM decision function is proposed by (Hastie and Tibshirani, 1996) The Gaussian p.d.f. of a particular class is estimated from the output values of the decision function, $f(x)$, for all x in that class. The centroid and radius of the Gaussian is the mean and standard deviation of $f(x)$ respectively.

4. EXPERIMENTS AND RESULTS

4.1 Database

In our work, we used a popular face databases (The UMIST [13]), for demonstrating the effectiveness of our combined KDDA and SVM proposed method. It is compared with KPCA, GDA and KDDA algorithm with nearest neighbor classifier.

We use a radial basis function (RBF) kernel function:

$$K(x,y) = \exp(\frac{-|x-y|^2}{(2\sigma^2)}) \quad (10)$$

Where σ is a scale parameter, and x and y are feature-vectors in the input space. The RBF function is selected for the proposed SVM method and KDDA in the experiments. The selection of scale parameter σ is empirical.

In addition, in the experiments the training set is selected randomly each time, so there exists some fluctuation among the results. In order to reduce the fluctuation, we do each experiment more than 10 times and use the average of them.

4.2 UMIST Database

The UMIST repository is a multi-view database, consisting of 575 images of 20 people, each covering a wide range of poses from profile to frontal views. Figure 1 depicts some samples contained in the two databases, where each image is scaled into (112 92), resulting in an input dimensionality of N = 10304.

For the face recognition experiments, in UMIST database is randomly partitioned into a training set and a test set with no overlap between the two set. We used ten images per person randomly chosen for training, and the other ten for testing. Thus, training set of 200 images and the remaining 375 images are used to form the test set.

Figure 1: Some sample images of four persons randomly chosen from the UMIST database.

It is worthy to mention here that both experimental setups introduce SSS conditions since the number of training samples are in both cases much smaller than the dimensionality of the input space [1].

On this database, we test the methods with different training samples and testing samples corresponding the training number k=2, 3, 4, 5,6,7,8 of each subject. Each time randomly select k samples from each subject to train and the other $10-K$ to test. The experimental results are given in the table 1.

Table 1. Recognition rate (%) on the UMIST database

K	Our method (KDDA+SVM)	KDDA +NN*	KPCA	GDA
2	81.8	81.9	75.5	71.5
3	83.5	83.4	76.2	72.8
4	87.3	85.4	77.1	74.5
5	90.4	87.9	79.8	75.1
6	94.1	89.1	83.4	79.0
7	96.0	93.9	87.1	82.1
10	96.5	95.2	89.1	83.0

* Nearest Neighbour

Figure 2 depicts the first two most discriminant features extracted by utilizing KDDA respectively and we show the decision boundary for first 6 classes for training data in Combination of one-to-rest classifier SVM.

The only kernel parameter for RBF is the scale value σ^2 for SVM classifier. Figure.4 shows the error rates as functions of σ^2, when the optimal number of feature vectors (M is optimum) is used. As such, the average error rates of our method with RBF kernel are shown in Figure 5. It shows the error rates as functions of M within the range from 2 to 19 (σ^2 is optimum).

5 Discussions and Conclusions

A new FR method has been introduced in this paper. The proposed method combines kernel-based methodologies with discriminant analysis techniques and SVM classifier. The kernel function is utilized to map the original face patterns to a high-dimensional feature space, where the highly non-convex and complex distribution of face patterns is simplified, so that linear discriminant techniques can be used for feature extraction. The small sample size problem caused by high dimensionality of mapped patterns is

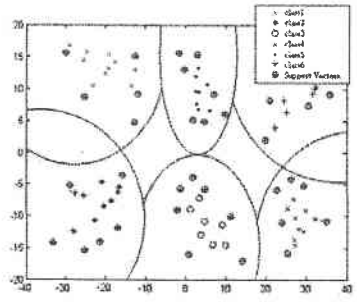


Figure 2: The decision boundary for first 6 classes for training data (Combination of one-to-rest classifier SVM)

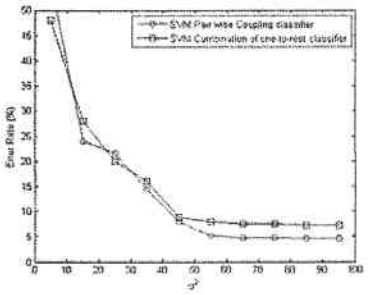


Figure 3: error rates as functions σ^2 of SVM. ($\sigma^2_{KDDA}=5\times10^6$ [1])

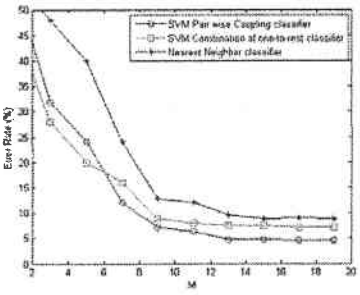


Figure 4: Comparison of error rates based on RBF kernel function.

addressed by a kernel-based D-LDA technique (KDDA) which exactly finds the optimal discriminant subspace of the feature space without any loss of significant discriminant information. Then feature space will be fed to SVM classifier. Experimental results indicate that the performance of the KDDA algorithm together with SVM is overall superior to those obtained by the KPCA or GDA approaches. In conclusion, the KDDA mapping and SVM classifier is a general pattern recognition method for nonlinearly feature extraction from high-dimensional input patterns without suffering from the SSS problem. We expect that in addition to face recognition, KDDA will provide excellent performance in applications where classification tasks are routinely performed, such as content-based image indexing and retrieval, video and audio classification.

Acknowledgment

The authors would like to acknowledge the Iran Telecommunication Research Center (ITRC) for financially supporting this work.
We would also like to thank Dr. Daniel Graham and Dr. Nigel Allinson for providing the UMIST face database.

References

[1] J. Lu, K. N. Plataniotis, A. N. Venetsanopoulos, "Face Recognition Using LDA-Based Algorithms" IEEE Trans. ON Neural Networks, vol. 14, no. 1, Jan.2003.

[2] M.Turk, "A random walk through eigenspace," *IEICE Trans. Inform.Syst.*, vol. E84-D, pp. 1586–1695, Dec. 2001.

[3] R. Chellappa, C. L.Wilson, and S. Sirohey, "Human and machine recognition of faces: A survey," *Proc. IEEE*, vol. 83, pp. 705–740, May 1995.

[4] M. Turk and A. P. Pentland, "Eigenfaces for recognition," *J. Cognitive Neurosci.*, vol. 3, no. 1, pp. 71–86, 1991.

[5] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 711–720, May 1997.

[6] M. A. Aizerman, E. M. Braverman, and L. I. Rozonoer, "Theoretical foundations of the potential function method in pattern recognition learning", Automation and Remote Control, vol. 25, pp. 821–837, 1964.

[8] L.-F.Chen, H.-Y. Mark Liao, M-.T. Ko, J.-C. Lin, and G.-J. Yu, "A new LDA-based face recognition system which can solve the small sample size problem," Pattern Recognition, vol. 33, pp. 1713–1726, 2000.

[9] H. Yu and J. Yang, "A direct LDA algorithm for high-dimensional data with application to face recognition," *Pattern Recognition*, vol. 34, pp. 2067–2070, 2001.

[10]V. N. Vapnik, "The Nature of Statistical Learning Theory", Springer-Verlag, New York, 1995.

[11] D. B. Graham and N. M. Allinson, "Characterizing virtual eigensignatures for general purpose face recognition," in Face Recognition: From Theory to Applications, H. Wechsler, P. J. Phillips, V. Bruce, F. Fogelman- Soulie, and T. S. Huang, Eds., 1998, vol. 163, NATO ASI Series F, Computer and Systems Sciences, pp. 446–456.

[12] D. L. Swets and J. Weng, "Using discriminant eigenfeatures for image retrieval," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 831–836, Aug. 1996.

[13] Q. Liu, R. Huang, H. Lu, S. Ma, "Kernel-Based Optimized Feature Vectors Selection and Discriminant Analysis for Face Recognition" 2002 IEEE

[14] C. Liu and H.Wechsler, "Evolutionary pursuit and its application to face recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 570–582, June 2000.

[15] K. Liu, Y. Q. Cheng, J. Y. Yang, and X. Liu, "An efficient algorithm for Foley–Sammon optimal set of discriminant vectors by algebraic method," *Int. J. Pattern Recog. Artificial Intell.*, vol. 6, pp. 817–829, 1992.

[16] 0. Duda, R., E. Han. P., and *G.* Stork, D. *Parrern Recognirion.* John Wiley & Sons, 2000.

[17] L. Mangasarian. 0 . and R. Musicant. D. Successive over relaxation for support vector machines, *IEEE Transacrions on Neural Nerworks,* l0(5), 1999.

SETIT 2007
4rth International Conference: **S**ciences of **E**lectronic, **T**echnologies of **I**nformation and **T**elecommunications
March 25-29, 2007 – TUNISIA

Feature Selection By KDDA For SVM-Based MultiView Face Recognition

Seyyed Majid Valiollahzadeh, Abolghasem Sayadiyan, Mohammad Nazari

Electrical Engineering Department, Amirkabir University of Technology, Tehran, Iran, 15914

valiollahzadeh@yahoo.com

eea35@aut.ac.ir

mohnazari@aut.ac.ir

Abstract: Applications such as Face Recognition (FR) that deal with high-dimensional data need a mapping technique that introduces representation of low-dimensional features with enhanced discriminatory power and a proper classifier, able to classify those complex features .Most of traditional Linear Discriminant Analysis (LDA) suffer from the disadvantage that their optimality criteria are not directly related to the classification ability of the obtained feature representation. Moreover, their classification accuracy is affected by the "small sample size" (SSS) problem which is often encountered in FR tasks. In this short paper, we combine nonlinear kernel based mapping of data called KDDA with Support Vector machine (SVM) classifier to deal with both of the shortcomings in an efficient and cost effective manner. The proposed here method is compared, in terms of classification accuracy, to other commonly used FR methods on UMIST face database. Results indicate that the performance of the proposed method is overall superior to those of traditional FR approaches, such as the Eigenfaces, Fisherfaces, and D-LDA methods and traditional linear classifiers.

Keywords: Face Recognition, Kernel Direct Discriminant Analysis (KDDA), small sample size problem (SSS), Support Vector Machine (SVM).

INTRODUCTION

Selecting appropriate features to represent faces and proper classification of these features are two central issues to face recognition (FR) systems. For feature selection, successful solutions seem to be appearance-based approaches, (see [3], [2] for a survey), which directly operate on images or appearances of face objects and process the images as two-dimensional (2-D) holistic patterns, to avoid difficulties associated with Three-dimensional (3-D) modelling, and shape or landmark detection [2]. For the purpose of data reduction and feature extraction in the appearance-based approaches, Principle component analysis (PCA) and linear discriminant analysis (LDA) are introduced as two powerful tools. Eigenfaces [4] and Fisherfaces [5] built on the two techniques, respectively, are two state-of-the-art FR methods, proved to be very successful. It is generally believed that, LDA based algorithms outperform PCA based ones in solving problems of pattern classification, since the former optimizes the low-dimensional representation of the objects with focus on the most discriminant feature extraction while the latter achieves simply object reconstruction. However, many LDA based algorithms suffer from the so-called "small sample size problem" (SSS) which exists in high-dimensional pattern recognition tasks where the number of available samples is smaller than the dimensionality of the samples. The traditional solution to the SSS problem is to utilize PCA concepts in conjunction with LDA (PCA+LDA) as it was done for example in Fisherfaces [11]. Recently, more effective solutions, called Direct LDA (D-LDA) methods, have been presented [12], [13]. Although successful in many cases, linear methods fail to deliver good performance when face patterns are subject to large variations in viewpoints, which results in a highly non-convex and complex distribution. The limited success of these methods should be attributed to their linear nature [14]. Kernel discriminant analysis algorithm, (KDDA) generalizes the strengths of the recently presented D-LDA [1] and the kernel techniques while at the same time overcomes many of their shortcomings and limitations.

In this work, we first nonlinearly map the original input space to an implicit high-dimensional feature space, where the distribution of face patterns is hoped

to be linearized and simplified. Then, KDDA method is introduced to effectively solve the SSS problem and derive a set of optimal discriminant basis vectors in the feature space. And then SVM approach is used for classification.

The rest of the paper is organized as follows. In Section tow, we start the analysis by briefly reviewing KDDA method. Following that in section three, SVM is introduced and analyzed as a powerful classifier. In Section four, a set of experiments are presented to demonstrate the effectiveness of the KDDA algorithm together with SVM classifier on highly nonlinear, highly complex face pattern distributions. The proposed method is compared, in terms of the classification error rate performance, to KPCA (kernel based PCA), GDA (Generalized Discriminant Analysis) and KDDA algorithm with nearest neighbour classifier on the multi-view UMIST face database. Conclusions are summarized in Section five.

2 Kernel Direct Discrimi-nant Analysis (KDDA)

2.1 Linear Discriminant Analysis

In the statistical pattern recognition tasks, the problem of feature extraction can be stated as follows: Assume that we have a training set, $\{Z_i\}_{i=1}^{L}$ is available. Each image is defined as a vector of length $N\left(=I_w \times I_h\right)$, i.e. $Z_i \in \Re^N$ where $I_w \times I_h$ is the face image size and $\Re^N$ denotes a N-dimensional real space [1].

It is further assumed that each image belongs to one of C classes $\{Z_i\}_{i=1}^{C}$. The objective is to find a transformation φ, based on optimization of certain separability criteria, which produces a mapping, with $y_i \in \Re^N$ that leads to an enhanced separability of different face objects.

Let S_{BTW} and S_{WTH} be the between- and within-class scatter matrices in the feature space $\mathbb{F}$ respectively, expressed as follows:

$$S_{BTW} = \frac{1}{L}\sum_{i=1}^{C} C_i \left(\overline{\phi_i} - \overline{\phi}\right)\left(\overline{\phi_i} - \overline{\phi}\right)^T \quad (1)$$

$$S_{WTH} = \frac{1}{L}\sum_{i=1}^{C}\sum_{j=1}^{C_i} \left(\overline{\phi_{ij}} - \overline{\phi_i}\right)\left(\overline{\phi_{ij}} - \overline{\phi_i}\right)^T \quad (2)$$

Where $\phi_{ij} = \phi(Z_{ij})$, $\overline{\phi_i}$ is the mean of class $Z_i j$ and $\overline{\phi}$ is the average of the ensemble.

$$\overline{\phi_i} = \frac{1}{C_i}\sum_{j=1}^{C_i} \phi\left(Z_{ij}\right) \quad (3)$$

$$\overline{\phi} = \frac{1}{L}\sum_{i=1}^{C}\sum_{j=1}^{C_i} \phi\left(Z_{ij}\right) \quad (4)$$

The maximization can be achieved by solving the following eigenvalue problem:

$$\Phi = \arg\max_{\Phi} \frac{\left|\Phi^T S_{BTW}\, \Phi\right|}{\left|\Phi^T S_{WTH}\, \Phi\right|} \quad (5)$$

The feature space F could be considered as a "linearization space" [6], however, its dimensionality could be arbitrarily large, and possibly infinite. Solving this problem lead us to LDA[1].

Assuming that is S_{WTH} nonsingular and Φ the basis vectors correspond to the M first eigenvectors with the largest eigenvalues of the discriminant criterion:

$$J = tr(S_{WTH}{}^{-1} S_{BtW} \Phi) \quad (6)$$

The M-dimensional representation is then obtained by projecting the original face images onto the subspace spanned by the eigenvectors.

2.2 Kernel Direct Discriminant Analysis (KDDA)

The maximization process in (3) is not directly linked to the classification error which is the criterion of performance used to measure the success of the FR procedure. Modified versions of the method, such as the Direct LDA (D-LDA) approach, use a weighting function in the input space, to penalize those classes that are close and can potentially lead to misclassifications in the output space.

Most LDA based algorithms including Fisherfaces [7] and D-LDA [9] utilize the conventional Fisher's criterion denoted by (3).

The introduction of the kernel function allows us to avoid the explicit evaluation of the mapping. Any function satisfying Mercer's condition can be used as a kernel, and typical kernel functions include polynomial function, radial basis function (RBF) and multi-layer perceptrons [10].

$$\Phi = \arg\max_{\Phi} \frac{\left|\Phi^T S_{BTW}\, \Phi\right|}{\left|\left(\Phi^T S_{BTW}\, \Phi\right) + \left(\Phi^T S_{WTH}\, \Phi\right)\right|} \quad (7)$$

The KDDA method implements an improved D-LDA in a high-dimensional feature space using a kernel approach.

KDDA introduces a nonlinear mapping from the input space to an implicit high dimensional feature space, where the nonlinear and complex distribution of patterns in the input space is "linearized" and "simplified" so that conventional LDA can be applied and it effectively solves the small sample size (SSS) problem in the high-dimensional feature space by employing an improved D-LDA algorithm.

Unlike the original D-LDA method of [10] zero eigenvalues of the within-class scatter matrix are never used as divisors in the improved one. In this way, the optimal discriminant features can be exactly extracted from both of inside and outside of S_{WTH} 's null space.

In GDA, to remove the null space of S_{WTH}, it is required to compute the pseudo inverse of the kernel matrix K, which could be extremely ill-conditioned when certain kernels or kernel parameters are used. Pseudo inversion is based on inversion of the nonzero eigenvalues.

3 SVM Based Approach for Classification

The principle of Support Vector Machine (SVM) relies on a linear separation in a high dimension feature space where the data have been previously mapped, in order to take into account the eventual non-linearities of the problem.

3.1 Support Vector Machines (SVM)

If we assume that, the training set $X = (x_i)_{i=1}^{l} \subset \mathrm{R}^R$ where l is the number of training vectors, R stands for the real line and R is the number of modalities, is labelled with two class targets $Y = (y_i)_{i=1}^{l}$, where :

$$y_i \in \{-1,+1\} \quad \Phi : \mathrm{R}^R \to F \tag{8}$$

Maps the data into a feature space F. Vapnik has proved that maximizing the minimum distance in space F between $\Phi(X)$ and the separating hyper plane $H(w,b)$ is a good means of reducing the generalization risk. Where:

$$H(w,b) = \{f \in F \mid < w, f >_F + b = 0\}, \quad (<> \textit{is} \text{ inner product}) \tag{9}$$

Vapnik also proved that the optimal hyper plane can be obtained solving the convex quadratic programming (QP) problem:

$$\begin{aligned} &\text{Minimize} && \frac{1}{2}\|w\|^2 + c\sum_{i=1}^{l}\xi_i \\ &\text{with} && y_i(< w, \Phi(X) > + b) \geq 1 - \xi_i \quad i = 1,...,l \end{aligned} \tag{10}$$

Where constant C and slack variables x are introduced to take into account the eventual non-separability of $\Phi(X)$ into F.

In practice this criterion is softened to the minimization of a cost factor involving both the complexity of the classifier and the degree to which marginal points are misclassified, and the tradeoff between these factors is managed through a margin of error parameter (usually designated C) which is tuned through cross-validation procedures.Although the SVM is based upon a linear discriminator, it is not restricted to making linear hypotheses. Non-linear decisions are made possible by a non-linear mapping of the data to a higher dimensional space. The phenomenon is analogous to folding a flat sheet of paper into any three-dimensional shape and then cutting it into two halves, the resultant non-linear boundary in the two-dimensional space is revealed by unfolding the pieces.

The SVM's non-parametric mathematical formulation allows these transformations to be applied efficiently and implicitly: the SVM's objective is a function of the dot product between pairs of vectors; the substitution of the original dot products with those computed in another space eliminates the need to transform the original data points explicitly to the higher space. The computation of dot products between vectors without explicitly mapping to another space is performed by a kernel function.

The nonlinear projection of the data is performed by this kernel functions. There are several common kernel functions that are used such as the linear, polynomial kernel $(K(x,y) = (< x, y >_{R^R} + 1)^d$ and the sigmoidal kernel $(K(x,y) = \tanh(< x, y >_{R^R} + a))$, where x and y are feature vectors in the input space.

The other popular kernel is the Gaussian (or "radial basis function") kernel, defined as:

$$K(x,y) = \exp\left(\frac{-|x-y|^2}{(2\sigma^2)}\right) \tag{11}$$

Where σ is a scale parameter, and x and y are feature-vectors in the input space. The Gaussian kernel has two hyper parameters to control performance C and the scale parameter σ. In this paper we used radial basis function (RBF).

3.2 Multi-class SVM

The standard Support Vector Machines (SVM) is designed for dichotomic classification problem (two classes, called also binary classification).

Several different schemes can be applied to the basic SVM algorithm to handle the K-class pattern classification problem. These schemes will be discussed in this section. The K-class pattern classification problem is posted as follow:

- Given l i.i.d. sample: $(x_1, y_1), ..., (x_l, y_l)$ where x_i, for $i=1,...,l$ is a feature vector of length d and $y_i = \{1, ..., k\}$ is the class label for data point x_i.
- Find a classifier with the decision function, $f(x)$ such that $y = f(x)$ where y is the class label for x.

The multi-class classification problem is commonly solved by decomposition to several binary problems for which the standard SVM can be used.

For solving the multi-class problem are as listed below:

- Using K one-to-rest classifiers (one-against-all)

- Using $k(k-1)/2$ pair wise classifiers
- Extending the formulation of SVM to support the k-class problem.

3.2.1. Combination of one-to-rest classifiers

This scheme is the simplest, and it does give reasonable results. K classifiers will be constructed, one for each class. The K-th classifier will be trained to classify the training data of class k against all other training data. The decision function for each of the classifier will be combined to give the final classification decision on the K-class classification problem. In this case the classification problem to k classes is decomposed to k dichotomy decisions $f_m(x)$, $m \in K = 1, ..., k$ where the rule $f_m(x)$ separates training data of the m-th class from the other training patterns. The classification of a pattern x is performed according to maximal value of functions $f_m(x)$, $m \in K$, $K = 1, ..., k$ i.e. the label of x is computed as:

$$f(x) = \arg(\max_{m \in k}(f_m(x)) \tag{12}$$

3.2.2. Pair wise Coupling classifiers

The schemes require a binary classifier for each possible pair of classes. The decision function of the SVM classifier for y_1-to-y_2 and y_2-to-y_1 has reflectional symmetry in the zero planes. Hence only one of these pairs of classifier is needed. The total number of classifiers for a K-class problem will then be $k(k-1)/2$. The training data for each classifier is a subset of the available training data, and it will only contain the data for the two involved classes. The data will be reliable accordingly, i.e. one will be labeled as +1 while the other as -1. These classifiers will now be combined with some voting scheme to give the final classification results. The voting schemes need the pair wise probability, i.e. the probability of x belong to class i given that it can be only belong to class i or j.

The output value of the decision function of an SVM is not an estimate of the p.d.f. of a class or the pair wise probability. One way to estimate the required information from the output of the SVM decision function is proposed by (Hastie and Tibshirani, 1996) The Gaussian p.d.f. of a particular class is estimated from the output values of the decision function, $f(x)$, for all x in that class. The centroid and radius of the Gaussian is the mean and standard deviation of $f(x)$ respectively.

4 EXPERIMENTS AND RESULTS

4.1 Database

In our work, we used a popular face databases (The UMIST [13]), for demonstrating the effectiveness of our combined KDDA and SVM proposed method. It is compared with KPCA, GDA and KDDA algorithm with nearest neighbor classifier.

We use a radial basis function (RBF) kernel function:

$$K(x, y) = \exp(\frac{-|x-y|^2}{(2\sigma^2)}) \tag{13}$$

Where σ is a scale parameter, and x and y are feature-vectors in the input space. The RBF function is selected for the proposed SVM method and KDDA in the experiments. The selection of scale parameter σ is empirical.

In addition, in the experiments the training set is selected randomly each time, so there exists some fluctuation among the results. In order to reduce the fluctuation, we do each experiment more than 10 times and use the average of them.

4.2 UMIST Database

The UMIST repository is a multi-view database, consisting of 575 images of 20 people, each covering a wide range of poses from profile to frontal views. Figure 1 depicts some samples contained in the two databases, where each image is scaled into (112 92), resulting in an input dimensionality of N = 10304.

For the face recognition experiments, in UMIST database is randomly partitioned into a training set and a test set with no overlap between the two set. We used ten images per person randomly chosen for training, and the other ten for testing. Thus, training set of 200 images and the remaining 375 images are used to form the test set.

It is worthy to mention here that both experimental setups introduce SSS conditions since the number of training samples are in both cases much smaller than the dimensionality of the input space [1].

Figure 1: Some sample images of four persons randomly chosen from the UMIST database.

On this database, we test the methods with different training samples and testing samples corresponding the training number k=2, 3, 4, 5,6,7,8 of each subject. Each time randomly select k samples from each subject to train and the other $10-K$ to test. The experimental results are given in the table 1.

Table 1. Recognition rate (%) on the UMIST database.

K	Our method (KDDA+SVM)	KDDA +NN *	KPCA	GDA
2	81.8	81.9	75.5	71.5
3	83.5	83.4	76.2	72.8
4	87.3	85.4	77.1	74.5
5	90.4	87.9	79.8	75.1
6	94.1	89.1	83.4	79.0
7	96.0	93.9	87.1	82.1
10	96.5	95.2	89.1	83.0

* Nearest Neighbour

Figure 2 depicts the first two most discriminant features extracted by utilizing KDDA respectively and we show the decision boundary for first 6 classes for training data in Combination of one-to-rest classifier SVM.

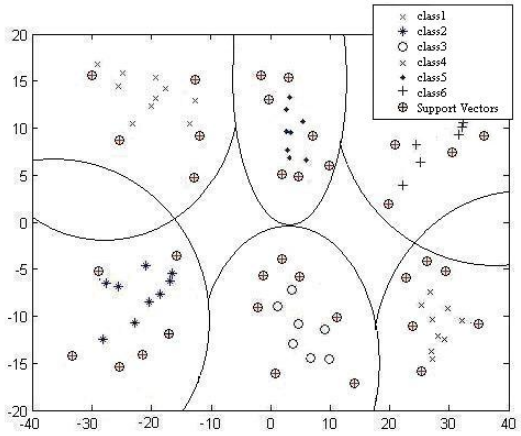


Figure 2: The decision boundary for first 6 classes for training data (Combination of one-to-rest classifier SVM)

The only kernel parameter for RBF is the scale value σ^2 for SVM classifier. Figure.4 shows the error rates as functions of σ^2, when the optimal number of feature vectors (M is optimum) is used.

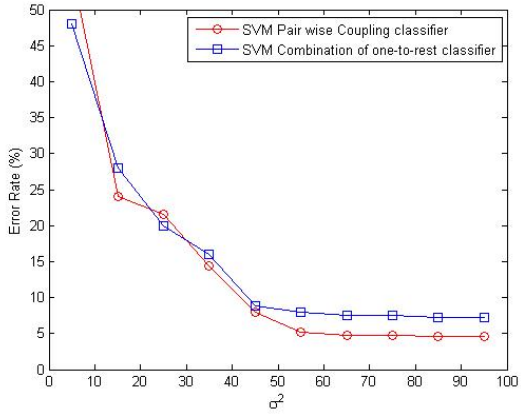


Figure 3: error rates as functions σ^2 of SVM. ($\sigma^2_{KDDA} = 5\times10^6$ [1])

As such, the average error rates of our method with RBF kernel are shown in Figure 5. It shows the error rates as functions of M within the range from 2 to 19 (σ^2 is optimum).

5 Discussions and Conclusions

A new FR method has been introduced in this paper. The proposed method combines kernel-based methodologies with discriminant analysis techniques and SVM classifier. The kernel function is utilized to map the original face patterns to a high-dimensional feature space, where the highly non-convex and complex distribution of face patterns is simplified, so that linear discriminant techniques can be used for feature extraction.

The small sample size problem caused by high dimensionality of mapped patterns is addressed by a kernel-based D-LDA technique (KDDA) which exactly finds the optimal discriminant subspace of the feature space without any loss of significant discriminant information.

Then feature space will be fed to SVM classifier. Experimental results indicate that the performance of the KDDA algorithm together with SVM is overall superior to those obtained by the KPCA or GDA approaches. In conclusion, the KDDA mapping and SVM classifier is a general pattern recognition method for nonlinearly feature extraction from high-dimensional input patterns without suffering from the SSS problem.

We expect that in addition to face recognition, KDDA will provide excellent performance in applications where classification tasks are routinely performed, such as content-based image indexing and retrieval, video and audio classification.

Acknowledgements

The authors would like to acknowledge the Iran Telecommunication Research Center (ITRC) for financially supporting this work.

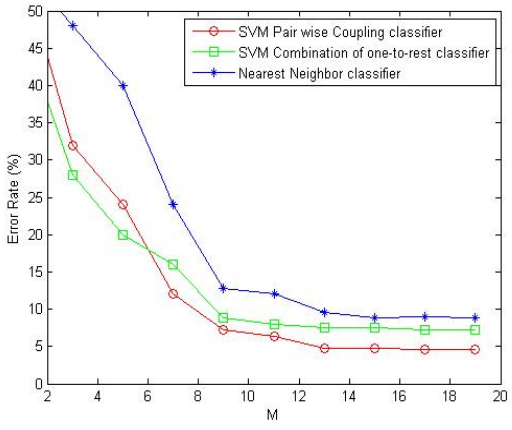


Figure 4: Comparison of error rates based on RBF kernel function.

We would also like to thank Dr. Daniel Graham and Dr. Nigel Allinson for providing the UMIST face database.

References

[1] J. Lu, K. N. Plataniotis, A. N. Venetsanopoulos, "Face Recognition Using LDA-Based Algorithms" IEEE Trans. ON Neural Networks, vol. 14, no. 1, Jan.2003.

[2] M.Turk, "A random walk through eigenspace," *IEICE Trans. Inform.Syst.*, vol. E84-D, pp. 1586–1695, Dec. 2001.

[3] R. Chellappa, C. L.Wilson, and S. Sirohey, "Human and machine recognition of faces: A survey," *Proc. IEEE*, vol. 83, pp. 705–740, May 1995.

[4] M. Turk and A. P. Pentland, "Eigenfaces for recognition," *J. Cognitive Neurosci.*, vol. 3, no. 1, pp. 71–86, 1991.

[5] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 711–720, May 1997.

[6] M. A. Aizerman, E. M. Braverman, and L. I. Rozonoer, "Theoretical foundations of the potential function method in pattern recognition learning", Automation and Remote Control, vol. 25, pp. 821–837, 1964.

[7] L.-F.Chen, H.-Y. Mark Liao, M-.T. Ko, J.-C. Lin, and G.-J. Yu, "A new LDA-based face recognition system which can solve the small sample size problem," Pattern Recognition, vol. 33, pp. 1713–1726, 2000.

[9] H. Yu and J. Yang, "A direct LDA algorithm for high-dimensional data with application to face recognition," *Pattern Recognition*, vol. 34, pp. 2067–2070, 2001.

[10] V. N. Vapnik, "The Nature of Statistical Learning Theory", Springer-Verlag, New York, 1995.

[11] D. B. Graham and N. M. Allinson, "Characterizing virtual eigensignatures for general purpose face recognition," in Face Recognition: From Theory to Applications, H. Wechsler, P. J. Phillips, V. Bruce, F. Fogelman- Soulie, and T. S. Huang, Eds., 1998, vol. 163, NATO ASI Series F, Computer and Systems Sciences, pp. 446–456.

[12] D. L. Swets and J. Weng, "Using discriminant eigenfeatures for image retrieval," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 831–836, Aug. 1996.

[13] Q. Liu, R. Huang, H. Lu, S. Ma, "Kernel-Based Optimized Feature Vectors Selection and Discriminant Analysis for Face Recognition" 2002 IEEE

[14] C. Liu and H.Wechsler, "Evolutionary pursuit and its application to face recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 570–582, June 2000.

[15] K. Liu, Y. Q. Cheng, J. Y. Yang, and X. Liu, "An efficient algorithm for Foley–Sammon optimal set of discriminant vectors by algebraic method," *Int. J. Pattern Recog. Artificial Intell.*, vol. 6, pp. 817–829, 1992.

[16] 0. Duda, R., E. Han. P., and *G.* Stork, D. *Parrern Recognirion.* John Wiley & Sons, 2000.

[17] L. Mangasarian. 0 . and R. Musicant. D. Successive over relaxation for support vector machines, *IEEE Transacrions on Neural Nerworks,* l0(5), 1999.