COMPRESSIVE CLASSIFICATION FOR FACE RECOGNITION

by

Angshul Majumdar

B. E Bengal Engineering and Science University, India, 2005

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

(Electrical and Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA
(Vancouver)

June 2009

ABSTRACT

The problem of face recognition has been studied widely in the past two decades. Even though considerable progress has been made, e.g. in achieving better recognition rates, handling difficult environmental conditions etc., there has not been any widespread implementation of this technology. Most probably, the reason lies in not giving adequate consideration to practical problems such as communication costs and computational overhead.

The thesis addresses the practical face recognition problem – e.g. a scenario that may arise in client recognition in Automated Teller Machines or employee authentication in large offices. In such scenarios the database of faces is updated regularly and the face recognition system must be updated at the same pace with minimum computational or communication costs. Such a scenario can not be handled by traditional machine learning methods as they assume the training is offline. We will develop novel methods to solve this problem from a completely new perspective.

Face recognition consists of two main parts: dimensionality reduction followed by classification. This thesis employs the fastest possible dimensionality reduction technique – random projections. Most traditional classifiers do not give good classification results when the dimensionality of the data is reduced by such method. This work proposes a new class of classifiers that are robust to data whose dimensionality has been reduced using random projections. The Group Sparse Classifier (GSC) is based on the assumption that the training samples of each class approximately form a linear basis for any new test sample belonging to the same class. At the core of the GSC is an optimization problem which although gives very good results is somewhat slow. This problem is remedied in the Fast Group Sparse Classifier where the computationally intensive optimization is replaced by a fast greedy algorithm. The Nearest Subspace Classifier is based on the assumption that the samples from a particular class lie on a subspace specific to that class. This assumption leads to an optimization problem which can be solved very fast. In this work the robustness of the said classifiers is proved theoretically and is validated by thorough experimentation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

PCA	Principal Component Analysis
LDA	Linear Discriminant Analysis
FLD	Fisher Linear Discriminant
NN	Nearest Neighbour
KNN	K Nearest Neighbour
SVM	Support Vector Machine
ANN	Artificial Neural Network
RP	Random Projections
CS	Compressive Sampling / Compressed Sensing
CC	Compressive Classification
HMM	Hidden Markov Model
OMP	Orthogonal Matching Pursuit
OLS	Orthogonal Least Squares
AAM	Active Appearance Model
RIP	Restricted Isometric Property
GRIP	Generalized Restricted Isometric Property
IID	Independently and Identically Distributed
LASSO	Least Angle Selection and Shrinkage
BP	Basis Pursuit
DCT	Discrete Cosine Transform

ACKNOWLEDGEMENTS

I would like to express my gratitude to those, without whom, it would not be possible for me to complete my thesis. The name that comes foremost in my mind is my wonderful supervisor Dr. Rabab K. Ward, who introduced me to her colleagues as ‘my son’. She introduced me to the academic world with equal sincerity, love and brilliance.

I am indebted to all the professors whose courses I took – Dr. Panos Nasiopoulos, Dr. David Lowe, Dr. Felix Herrmann, Dr. Michael Friedlander, Dr. Vikram Krishnamurthy, Dr. Arnaud Doucet and Dr. Rafeef Abugharbieh. All of them were kind enough to listen to my problems and extend their help beyond the classroom. I am especially thankful to Dr. Nasiopoulos for reviewing and correcting two of my papers in spite of being on a sabbatical. I am thankful to Dr. Abugharbieh, Dr. Lowe and Dr. Ward as committee members of my thesis defense. They gave me valuable suggestions for improving the thesis.

My sincere thanks to all the members of this wonderful Signal, Image and Multimedia Processing Lab (SIMPL) with whom I had a wonderful time both in and out of the Lab. I start with a ‘thank you’ to Colin Doutre and Angela Chuang, who took the trouble of receiving me at the Airport. Special thanks to Dr. Mehrdad Fatourechi, who was the administrator of our Lab when I first came. Even before I arrived here, he introduced me to a couple of members of this Lab – Dr. Shan Du and Zicong (Jack) Mai. Shan and I worked on the same problem (face recognition) and she always was there to help me. Jack became a pal even before I arrived in Vancouver, we used to chat online. I extend my gratitude to all the other members of the Lab, especially Tanaya Mandal who helped me a lot during my thesis. I would like to express my gratitude to Rayan from our Lab and Ewout Van Den Berg from the Scientific Computing Lab for the fruitful academic discussions.

Finally my thanks to my father without whose encouragement I wouldn’t be here today. I extend my thanks to the rest of my family and especially my grandmother who said these words when I was leaving for Vancouver in 2007, “…I do not know anything about your subject, but I wish you will do good in whatever you do…”

DEDICATION

To my Father and my Late Mother who would have been proud to see this day.

CO-AUTHORSHIP STATEMENT

This thesis presents research conducted by Angshul Majumdar, in collaboration with Dr. Rabab Ward, and Dr. Panos Nasiopoulos.

Manuscript 1: *CLASSIFICATION VIA GROUP SPARSITY PROMOTING REGULARIZATION.* This manuscript was the work of Angshul Majumdar, who received suggestions and feedback from his supervisor, Dr. Ward.

Manuscript 2: *FAST GROUP SPARSE CLASSIFICATION.* This manuscript was the work of Angshul Majumdar, who received suggestions and feedback from his supervisor, Dr. Ward.

Manuscript 3: *NEAREST SUBSPACE CLASSIFIER.* This manuscript was the work of Angshul Majumdar, who received suggestions and feedback from his supervisor, Dr. Ward.

Manuscript 4: *COMPRESSIVE CLASSIFICATION.* This manuscript was the work of Angshul Majumdar, who received suggestions and feedback from his supervisor, Dr. Ward. The language was checked by Craig Wilson.

Manuscript 5: *FACE RECOGNITION FROM VIDEO: RANDOM PROJECTIONS AND HYBRID CLASSIFICATION.* This manuscript was the work of Angshul Majumdar, who received suggestions and feedback from his supervisor Dr. Ward and from Dr. Nasiopoulos.

The first and last chapters of the thesis were written by Angshul Majumdar, with editing assistance and consultation from Dr. Rabab Ward.

CHAPTER 1: INTRODUCTION AND OVERVIEW

1.1 Introduction

Face recognition is a typical problem in classification. One or more training images of each person under study are available to the classifier along with their identities. The problem is to identify a person from a new (test) image. There are two slight variations to the basic problem. The first one is face authentication – this is a binary classification problem, and the classifier has to decide whether or not the new image belongs to the training set. The second one is face recognition – this is a more difficult multiclass problem, where the identity of the new image must be decided. Face recognition can also be carried out with videos, but in this work the focus is on image based face recognition.

Face recognition had been an active field of research over the last two decades. A detailed review of face recognition research can be found in [1]. Any face recognition system consists of three interconnected modules as shown in Figure 1-1. The pre-processing module comprises of tasks like face detection, illumination normalization, pose estimation etc. The feature extraction block, extracts features that are relevant for representing a face for recognition problems. The classification block is either designed to carry out authentication or recognition, or sometimes both.

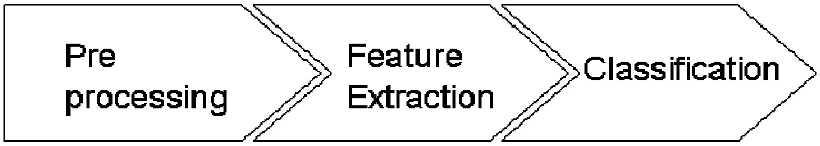

Figure 1-1: Face Recognition Modules

The main task of the preprocessing module is face detection. This is a problem in its own class where the challenge is to detect and separate faces from images with varying scales, backgrounds and poses. There are several methods to address the problem [2]; perhaps the most popular one (owing to its reasonable accuracy and high speed) is the Viola Jones algorithm

[3]. This is the algorithm used nowadays in digital cameras for detecting faces in real-time. The preprocessing block also handles other problems like illumination normalization, pose estimation etc. There have been previous studies [4] in low complexity algorithms to handle such problems at the preprocessing level.

Feature extraction is the most well researched problem in face recognition. There are three broad approaches towards feature extraction (Figure 1-2) – holistic, local and hybrid. In the holistic recognition approach, features are extracted from the whole face. Eigenfaces, Fisherfaces are examples of this approach. These extracted features are used by a classifier for recognition. In local methods, the entire face is divided into several patches. These patches may be regular grids (as shown in Figure 1-2) or may depend on interesting regions of the face like eyes, nose, mouth etc. Features are extracted from these patches. Corresponding to each patch there is one classifier. These classifiers are weak and try to distinguish the face from the patch information. Finally, the output of the weak classifiers is fused to arrive at the final decision. Hybrid methods believe that global and local methods carry complementary information. So, an approach that hauls both types of information may provide better recognition accuracy. In hybrid methods, first, the local features are extracted. Discriminative information is extracted from all the local features taken together. These discriminative features are fed into the classifier for recognition.

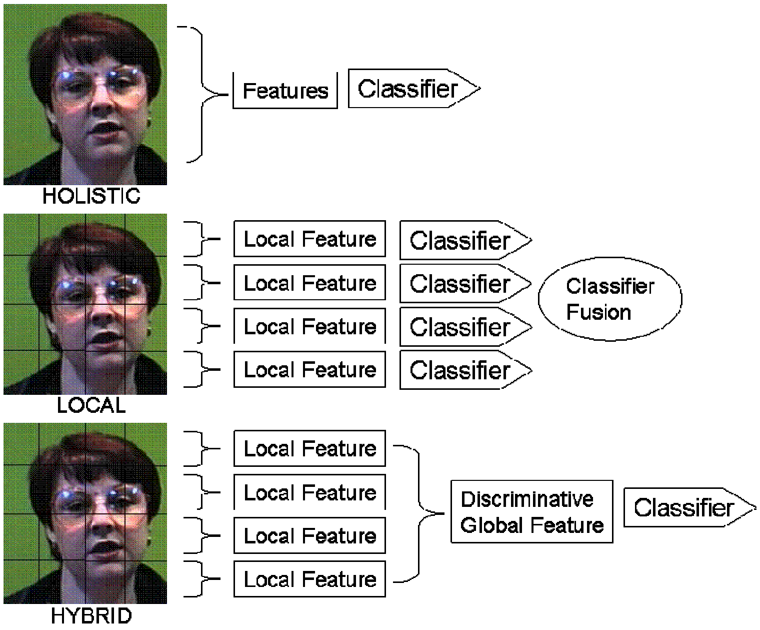

Figure 1-2: Face Recognition Approaches

In the classification module, the extracted features form the input to the classifier for recognition. Generally, standard classification tools like Nearest Neighbor (NN), Support Vector Machine (SVM) and Artificial Neural Networks (ANN) are employed for face recognition. The NN is the most widely used classifier. This owes to the fact that face recognition suffers from the problem of insufficient training samples, and under such conditions, parametric classifiers like SVM and ANN overfit the training data thus losing their generalization ability due to over-fitting. This leads to poor recognition accuracy for new test samples. NN is a non-parametric classifier and does not suffer from the problem of over-fitting and hence is preferred over the others.

1.2 Face Recognition Challenges

From the perspective of a machine, faces are specific objects that look similar, but subtle features make one different from the other. Human beings are easily able to recognize faces but automatic recognition of faces by machine is a challenging job. From the point of view of computer vision, face recognition poses the following challenges:

- Head pose: Rotation or tilt of the head; even if the appearance is frontal, it significantly affects the performance of the recognition system significantly.

- Illumination change: The direction of light illuminating the faces greatly affects the recognition accuracy. It has been noticed that illuminating a face image bottom up reduces the accuracy of the system.
- Facial expression change: Minor expression changes like smiling to extreme expression variations, like shouting or crying or making grimaces, has some effect on recognition.
- Aging: Images taken at long interval or even at different days may seriously affect the correct recognition rate of a system.
- View: Profile images can be difficult to recognize, when mostly frontal faces are available for matching.
- Occlusion: Even partial occlusion of faces due to objects or accessories like sunglass or scarf, makes identification a difficult task.
- Miscellaneous: Other issues like variations in background, changes in hairstyle also add to the problem of face recognition to a certain extent.

All the aforementioned problems have been at the forefront of face recognition research for the better part of the last two decades. While the challenges are still pertinent, the current work does not address them. The interested reader is directed to [5], where all the classic works in face recognition are archived. The main purpose of this work is to address a practical recognition problem which has been overlooked previously.

1.3 Purpose of Thesis

Face images (with column/row concatenation) form very high dimensional vectors, e.g. a standard webcam takes images of size 320x240 pixels, which leads to a vector of length 76,800. All the pixel values of the face image may be used as features for classification. The computational complexity of most classifiers is dependent on the dimensionality of the input features. If the dimensionality is large, the classifier takes a long time to operate. This prohibits direct usage of pixel values as features for face recognition.

To overcome this problem, different dimensionality reduction techniques (i.e. the feature extraction block of Figure 1-1) has been proposed over the last two decades – starting from Principal Component Analysis (Eigenface [6]) and Fisher Linear Discriminant (Fisherface [7]). A comprehensive study in different dimensionality reduction techniques can be found in [8]. All such dimensionality reduction techniques have a basic problem – they are data-dependent adaptive techniques, i.e. the projection function from the higher to lower dimension can not be computed unless al the training samples are available. Thus the system cannot be updated efficiently when new data should be added.

Data dependency is the major computational bottleneck of such dimensionality reduction methods. Consider a situation where a bank intends to authenticate a person at the ATM, based on face recognition. So, when a new client is added to its customer base, a training image of the person is acquired. When that person goes to an ATM, another image is acquired by a camera at the ATM and the new image is compared against the old one for identification. Suppose that at a certain time the bank has 200 customers, and is employing a data-dependent dimensionality reduction method. At that point of time it has computed the projection function from higher to lower dimension for the current set of images. Suppose that at a later time, the bank has 10 more clients, then with the data-dependent dimensionality reduction technique the projection function for all the 210 samples must be recomputed from scratch; in general there is no way the previous projection function can be updated with results of the 10 new samples only. This is a major computational bottleneck for the practical application of current face recognition research.

For an organization such as a bank, where new customers are added regularly, it means that the projection function from higher to lower dimension will have to be updated regularly. The cost of computing the projection function is intensive and is dependent on the number of samples. As the number of samples keeps on increasing, the computational cost keeps on increasing as well (as every time new customers are added to the training dataset, the projection function has to be recalculated from scratch). This is a major issue for any practical face recognition system.

One way to work around this problem is to skip the dimensionality reduction step. But as mentioned earlier this increases the classification time. With the ATM scenario there is another problem as well. This is from the perspective of communication cost. There are two possible scenarios in terms of transmission of information – 1) the ATM sends the image to some central station where dimensionality reduction and classification are carried out or 2) the dimensionality reduction is carried out at the ATM so that the dimensionality reduced feature vector is sent instead. The latter reduces the volume of data to be sent over the internet but requires that the dimensionality reduction function is available at the ATM. With the first scenario, the communication cost arises from sending the whole image over the communication channel. In the second scenario, the dimensionality reduction function is available at the ATM. As this function is data-dependent it needs to be updated every time new samples are added. Periodically updating the function increases the communication cost as well.

In this work we propose a dimensionality reduction method that is independent of the data. Practically this implies that the dimensionality reduction function is computed once and for all and is available at all the ATMs. There is no need to update it, and the ATM can send the dimensionality reduced features of the image. Thus both the computational cost of calculating the projection function and the communication cost of updating it are reduced simultaneously.

Our dimensionality reduction is based on Random Projection (RP). In the past, RP has been proposed as a non-adaptive (data-independent) alternative to adaptive data-dependent dimensionality reduction techniques in machine learning problems [9-15]. Two of these studies [14, 15] were aimed at face recognition. In [14] RP was employed for dimensionality reduction followed by NN classification. A new classifier was proposed in [15] for use with RP dimensionality reduction. It was called the Sparse Classifier (SC). SC was found to give better recognition results than NN or SVM. In this work, a class of novel classifiers is proposed which are robust to RP. ‘Robustness’ implies that the classification accuracy in the original dimension (before dimensionality reduction) and in the randomly projected reduced dimension is nearly the same.

Another way to generate a data-independent dimensionality reduction matrix is to generate a PCA (or LDA) projection matrix for a large number of selected samples and use this projection matrix for reducing the dimensionality of any new data without updating it. Such a scheme may give good results, but it will not be possible to prove the robustness of such an ad hoc dimensionality reduction scheme without making strong assumptions on the distribution of the data.

1.4 Contribution of the Thesis

This thesis aims at proposing a class of new classifiers that are robust to dimensionality reduction via Random Projections (RP). By robust, it is meant that the classification accuracy does not vary much when the RP dimensionality reduced samples are used in classification instead of the original samples (without dimensionality reduction). The traditional NN classifier is robust to RP. Other classifiers like SVM has been shown to be somewhat robust to RP as well, but there is a problem with using classifiers like SVM and ANN in the current scenario. Both SVM and the ANN have a data-driven training phase, where all the training data must be available to the classifier in order to learn how to classify (e.g. in the ATM scenario, SVM and ANN need to be retrained whenever new samples are available). The computational cost of training these classifiers increases when the number of samples increases. This is a computational bottleneck that needs to be avoided for practical recognition systems. Moreover, with limited number of samples in each class (as in face recognition) SVM and ANN do not show good recognition accuracy, since both of them are parametric classifiers which suffer from the problem of over-fitting when the training samples are scarce, consequently they lose their generalization ability leading to poor accuracy while testing.

The NN is robust to RP dimensionality reduction; it does not require training so new data can be added anytime. Unfortunately the recognition accuracy of NN is not very high. The Sparse Classifier (SC), [15] is shown to have better recognition accuracy (17% more compared to NN). But there are certain shortcomings with the SC implementation. In Chapter 2, the shortcomings of the SC are addressed and a new classifier called the Group Sparse Classifier (GSC), is proposed. The results from GSC are better than SC by about 3%.

The GSC algorithm is based on optimization techniques which are slow. To overcome the limitations in speed, fast approximate algorithms are proposed in Chapter 3 as an alternate to optimization. These classifiers based on the greedy approximate algorithms are called the Fast Group Sparse Classifiers (FGSC). FGSC has a marginally less (0.5%) classification accuracy compared to GSC but is about 2 orders of magnitude faster.

The Nearest Subspace Classifier (NSC) is proposed in Chapter 4. It can be looked upon as a generalization of the NN algorithm or a special case of the GSC. The operational speed of NSC is much faster compared to the sparse classifiers (SC, GSC and FGSC).

All the novel classification algorithms (including SC) are analyzed in detail in Chapter 5. Theoretical proofs regarding the robustness of these classifiers under random projections are also provided in this chapter.

Chapter 6 shows a practical application of the techniques developed so far for the problem of video based face recognition in ATMs. It shows how the non-adaptive techniques developed in this thesis can be extended to the problem of video based face recognition. Our proposed method leads to significant reduction in recognition error (65%) when compared with other recent methods in video based face recognition. The conclusions of this work are discussed in the final chapter 7.

1.5 References

[1] W. Zhao, R. Chellappa, A. Rosenfeld, and P.J. Phillips, “Face Recognition: A Literature Survey”, ACM Computing Surveys, pp. 399-458, 2003.

[2] http://www.facedetection.com/

[3] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” Conference on Computer Vision and Pattern Recognition, 2001, pp. 511-518.

[4] Shan Du, "Image-based face recognition under varying pose and illuminations conditions," PhD thesis, UBC, Nov. 28, 2008.

[5] http://face-rec.org/

[6] M. Turk, A. Pentland, “Eigenfaces for Recognition”, Journal of Cognitive Neuroscience, Vol. 3 (1), pp. 71-86, 1991.

[7] P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman, Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19 (7), pp. 711-720, 1997.

[8] I. K. Fodor, “A survey of dimension reduction techniques”, Technical Report, UCRL-ID-148494., 2002.

[9] D. Achlioptas, “Database-friendly random projections”. ACM Symposium on the Principles of Database Systems, pp. 274–281, 2001.

[10] X. Z. Fern, and C. E. Brodley, “Random projection for high dimensional data clustering: A cluster ensemble approach”, International Conference on Machine Learning, pp. 186-193., 2003.

[11] D. Fradkin and D. Madigan “Experiments with random projection for machine learning”, ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp517-522, 2003.

[12] S. Kaski, “Dimensionality reduction by random mapping: Fast similarity computation for clustering”, IEEE International Joint Conference on Neural Networks, pp. 413-418, 1998.

[13] A. Magen, “Dimensionality reductions that preserve volumes and distance to affine spaces, and their algorithmic applications”, International Workshop on Randomization and Approximation Techniques in Computer Science, pp. 239-253, 2002.

[14] N. Goel, G. M. Bebis and A. V. Nefian, “Face recognition experiments with random projections”, SPIE Conference on Biometric Technology for Human Identification, pp. 426-437, 2005.

[15] Y. Yang, J. Wright, Y. Ma and S. S. Sastry, “Feature Selection in Face Recognition: A Sparse Representation Perspective”, IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 31 (2), pp. 210-227, 2009.

CHAPTER 2: CLASSIFICATION VIA GROUP SPARSITY PROMOTING REGULARIZATION[1]

2. 1 Introduction

Recently a new classifier was proposed in [1]. The work makes a novel classification assumption. It assumes that the training samples of a particular class approximately form a linear basis for a new test sample belonging to the same class. The classification algorithm built upon this assumption gave good recognition results on the Extended Yale B face recognition database [2].

The assumption made in [1] is novel and departs largely from the assumptions of conventional machine learning. Although this assumption led to good results, the intuition behind this assumption is lacking in the previous work [1]. We propose a logical interpretation of this assumption. Geometrically, this assumption means that training samples for each class can lie on different subspaces, and the test sample can be expressed as a union of these subspaces. For faces, it is likely that different head poses constitute different subspaces, e.g. frontal view may be one subspace, left profile view may be another subspace while the right profile view may be the third one. It is not difficult to assume that any new head pose may be constructed as a linear combination of these three fundamental views. The mathematical form of this interpretation leads to the assumption in [1]. If $v_{k,test}$ is the test sample belonging to the k^{th} class then,

$$v_{k,test} = \alpha_{k,1}v_{k,1} + \alpha_{k,2}v_{k,2} + ... + \alpha_{k,n_k}v_{k,n_k} + \varepsilon \qquad (1)$$

where $v_{k,i}$ are the training samples and ε is the approximation error.

In a classification problem, the training samples and their class labels are provided. The task is to assign the given test sample with the correct class label. This requires finding the coefficients $\alpha_{k,i}$ in equation (1). In [1] the solution is framed as a sparse optimization problem. In this work we

[1] A version of this chapter has been published. A. Majumdar, and R. K. Ward, "Classification via Group Sparsity Promoting Regularization", IEEE International Conference on Acoustics, Speech, and Signal Processing, Taipei, Taiwan, Apr. 2009, pp. 873-876.

propose alternate solutions and show that our solutions give better results compared to the previous one [1].

Equation (1) expresses the assumption in terms of the training samples of a single class. Alternately, it can be expressed in terms of all the training samples (of C classes) so that

$$v_{k,test} = V\alpha + \varepsilon \quad (2)$$

where $V = [v_{1,1} | \ldots | v_{n,1} | \ldots | v_{k,1} | \ldots | v_{k,n_k} \ | \ldots v_{C,1} | \ldots | v_{C,n_C}]$ and $\alpha = [\alpha_{1,1} \ldots \alpha_{1,n_1} \ldots \alpha_{k,1} \ldots \alpha_{k,n_k} \ldots \alpha_{C,1} \ldots \alpha_{C,n_C}]'$.

There are two implications that follow from the assumption expressed in equation (2):

1. The vector α should be sparse.
2. All (or most of) the training samples corresponding to the correct class should have non-zero values in α.

The above implications demand that α should be ‘group sparse’ - meaning that the solution of the inverse problem (2) should have non-zero coefficients corresponding to a particular group of correlated training samples and zero elsewhere. The solution in [1] is based on the first implication only. It imposes Lasso regularization on equation (2). Lasso promotes a sparse solution of α but does not favor grouping of correlated samples. Consequently the non-zero values in α do not necessarily correspond to training samples belonging to the same group. Our work proposes two alternate regularizations that promote group sparsity in α. Experimental evaluation shows that our method provides better recognition results compared to [1].

The rest of the paper will be organized into several sections. In Section 6.2.2, we will discuss the background of the problem. Section 6.2.3 will detail our proposed methods. In Section 6.2.4 we will show the experimental results. Finally in Section 6.2.5, conclusions and future scope of work will be discussed.

2.2 Review of Previous Work

The novel classification assumption was first proposed in [1]. The first step towards classification is to solve for the coefficient vector α in equation (2). The simplest solution to equation (2) involves the pseudo-inverse of V and is expressed as $\hat{\alpha} = (V'V)^{-1}V'v_{k,test}$. However, in most cases the matrix *V* is ill-conditioned or ill-posed. So the simple solution involving the pseudo-inverse is not stable.

To obtain a stable solution, one requires regularizing equation (2) in order to find an approximate stable solution. Since the solution $\hat{\alpha}$ should be sparse, an *l₀*-norm regularizer is required and the following optimization problem needs to be solved

$$\min_{\alpha} \|\alpha\|_0 \text{ such that } \|v_{k,test} - V\alpha\|_2 < \varepsilon \qquad (3)$$

In [1], it is argued that solving the l0-norm is an NP hard problem and there is no tractable algorithm to solve it. Citing studies in Compressive Sampling [3], they argued that for large systems the l_0-norm can be replaced by the l_1-norm (Lasso regularization) which also leads to a sparse solution.

$$\min_{\alpha} \|\alpha\|_1 \text{ such that } \|v_{k,test} - V\alpha\|_2 < \varepsilon \qquad (4)$$

The optimization problem in equation (4) can be solved by quadratic programming methods.

Once a sparse solution of α is obtained, the following classification algorithm was proposed to determine the class of the test sample.

Algorithm 1

1. Solve the optimization problem expressed in (4).
2. For each class i repeat the following two steps:

a. Reconstruct a sample for each class by a linear combination of the training samples belonging to that class by the equation $v_{recon}(i) = \sum_{j=1}^{n_i} \alpha_{i,j} v_{i,j}$.

b. Find the error between the reconstructed sample and the given test sample by $error(v_{test}, i) = \| v_{k,test} - v_{recon(i)} \|_2$.

3. Once the error for every class is obtained, choose the class having the minimum error as the class of the given test sample.

l_1-norm minimization leads to a sparse solution, but there are spurious coefficients in the vector α associated with the samples that do not belong to the class of the test sample. Step 2 is required to eliminate the effects of these coefficients. The coefficients in $\alpha_{i,j}$ corresponding to each class i are used to reconstruct a sample. For each class the error between the reconstructed sample and the given test sample is calculated. The assumption in equation (1) says that the error between the reconstruction and the test sample will be the least for the correct class. Based on this assumption, the identity of the test sample is decided by the minimum error.

2.3 Proposed Classification Methods

We mentioned in Section 6.2.1 that the assumption in [1] leads to two implications. The previous work [1] based their solution on the first implication, i.e. on the sparsity of the solution. It did not account for the group sparsity of α. In this section we will introduce regularizations that make α group sparse, i.e. it has non-zero coefficients corresponding to a particular group of correlated training samples and zero elsewhere.

2.3.1. Disadvantage of Lasso Regularization

There is a limitation to the Lasso (l_1-norm) regularization. If there is a group of samples whose pair-wise correlations are very high, then Lasso tends to select one sample only from the group [5].

In a classification problem, training samples belonging to the same class are correlated with each other. In such a situation the Lasso regularization proposed in [1] tends to select only a single training sample from the entire class. Thus, in the extreme case, the classifier in [1] becomes a scaled version of the Nearest Neighbour (NN) classifier.

For explaining this effect of the Lasso regularization we rewrite the assumption expressed in equation (1):

$$v_{k,test} = \alpha_{k,1}v_{k,1} + \alpha_{k,2}v_{k,2} + ... + \alpha_{k,n_k}v_{k,n_k} + \varepsilon$$

where the v_k's belong to the same class and are correlated with each other. If algorithm 1 is employed for classifying the test sample, then (in the extreme case) we find that

1. The Lasso regularization tends to select only one of the training samples from the group. We call it $v_{k,best}$.
2. Step 2 is repeated for each class.
    a. The reconstructed vector becomes a scaled version of the selected sample, i.e. $v_{recon}(i) = \alpha_{i,best}v_{i,best}$.
    b. The error from the reconstructed vector is calculated $error(v_{test}, i) = || v_{k,test} - \alpha_{i,best}v_{i,best} ||_2$.
3. The class with the minimum error is assumed to be the class of the test sample.

The minimum Lasso error in step 2.b is $|| v_{k,test} - \alpha_{k,best}^{Lasso}v_{k,best}^{Lasso} ||_2$. In NN classification the criterion for choosing the class of the test sample is $|| v_{k,test} - v_{i,j} ||_2 \;\; \forall$ j $\in$ class i . This error is minimized when $v_{i,j} = v_{k,best}^{NN}$ and is given by $|| v_{k,test} - v_{k,best}^{NN} ||_2$. The Lasso error and the NN error are the same except for the scaling factor ($\alpha_{k,best}^{Lasso}$).

When the training samples are highly correlated (which generally is the case in classification), employing Lasso regularization forms a serious limitation to the sparse classification problem. Decision regarding the correct class of the test sample should depend on all the training samples belonging to a class. Lasso however, favors selecting a single training sample from the entire class. We look for alternate regularization methods to overcome this problem.

2.3.2 Elastic Net Regularization

The problem of selecting a sparse group is studied in [5, 6] where an alternate regularization called 'Elastic Net' that promotes selection of sparse groups is proposed. We apply this regularization to the classification problem.

We repeat the optimization problem used in [1]

$$\min_{\alpha} \| \alpha \|_1 \text{ such that } \| v_{k,test} - V\alpha \|_2 < \varepsilon$$

This has the equivalent (Lasso) expression

$$\min_{\alpha} \| v_{k,test} - V\alpha \|_2 \text{ such that } \| \alpha \|_1 < \tau$$

The unconstrained form of Lasso regularization is

$$\min_{\alpha} \| v_{k,test} - V\alpha \|_2 + \lambda \| \alpha \|_1 \qquad (5)$$

To promote group sparsity, Elastic Net regularization, proposes the following optimization problem

$$\min_{\alpha} \| v_{k,test} - V\alpha \|_2 + \lambda_1 \| \alpha \|_2^2 + \lambda_2 \| \alpha \|_1 \qquad (6)$$

The l_1 penalty in the above expression promotes sparsity of the coefficient vector α, while the quadratic l_2 penalty encourages grouping effect, i.e. selection of a group of correlated training samples. The combined effect of the mixed penalty term is that it enforces group-sparsity, i.e. the recovery of one or very few groups of correlated samples.

The classification is performed by algorithm 1, but instead of solving the optimization problem in equation (4) we need to solve the problem in equation (6). The Elastic Net regularization problem was solved using the 'elasticnet' package [7].

2.3.3 Sum-Over-l_2-norm Regularization

In Section 6.2.1, we mentioned two implications of the assumption expressed in equation (1). The Lasso exploits only the first implication while Elastic Net exploits both. The Elastic Net regularization is better than the Lasso in the sense that it promotes the selection of one or very few groups of samples. Elastic Net regularization however, does not exploit the labels of the training samples (columns of V). When the labels are known a stronger group sparsity constraint than the Elastic Net can be imposed.

When the column labels of the matrix V is known, a stronger group sparsity promoting regularization [8, 9] can be employed

$$\begin{aligned}&\min_{\alpha} \|\mathrm{A}_1\|_2 + \|\mathrm{A}_2\|_2 + \ldots + \|\mathrm{A}_C\|_2 \\ &\text{such that } \|v_{k,test} - V\alpha\|_2 < \varepsilon \\ &\text{where } \mathrm{A}_i = [\alpha_{i,1}, \alpha_{i,2}, \ldots, \alpha_{i,n_i}], \text{ for i = 1,2,...,C}\end{aligned} \qquad (7)$$

The formulation is similar to the Elastic Net regularization. The l2-norm over the group of correlated variables (A_i's) enforces the selection of the entire group of samples whereas the summation over the l_2-norm ($\sum A_i$) enforces group sparsity, i.e. the selection of one or very few classes.

The optimization problem (7) requires the label of each column in the matrix V, i.e. the class the column belongs to. In classification tasks, the labels of the training samples are always available, and hence we can use the Sum-Over-l_2-norm regularization (7) for our problem. We propose a slightly modified version of algorithm 1 in this case.

Algorithm 2

1. Solve the optimization problem expressed in (7).
2. Find those i's for which $||A_i||_2 > 0$.
3. For those classes i satisfying the condition in step 2, repeat the following two steps:

a. Reconstruct a sample for each class by a linear combination of the training samples in that class via the equation $v_{recon}(i)=\sum_{j=1}^{n_i}\alpha_{i,j}v_{i,j}$.

b. Find the error between the reconstructed sample and the given test sample by $error(v_{test},i)=\| v_{k,test}-v_{recon(i)}\|_2$.

4. Once the $error(v_{test},i)$ for every class i is obtained, choose the class having the minimum error as the class of the given test sample.

The computational cost of algorithm 2 is less than algorithm 1, because step 3 is not repeated for all the classes. Instead we evaluate only those classes for which there are non-zero entries in the coefficient vector α (step 2).

2.4 Experimental Results

We performed two sets of experiments. In the first set we apply the sparse classification algorithms on some benchmark databases from the University of California Irvine Machine Learning (UCI ML) repository [8]. Databases that do not have missing values in feature vectors or unlabeled training data were chosen.

In the second set of experiments, we compared the recognition accuracy on the different sparse classifiers for the face recognition task on the Extended Yale B face database. The sparse classification algorithm [1] was originally proposed to address the face recognition problem.

Table 2-1, shows the classification results on the UCI ML databases. The results are obtained by Leave-One-Out validation. We compare the classification algorithm in [1] against ours. We use the Nearest Neighbour (NN) classifier as a benchmark.

Table 2-1: Recognition Accuracy of Different Methods

Name of Dataset	Recognition Accuracy (%)			
	Lasso [1]	Elastic Net	Sum-Over-l_2-norm	NN
Page Block	94.78	95.32	**95.66**	93.34
Abalone	**27.17**	**27.17**	**27.17**	26.67
Segmentation	**96.31**	94.09	94.09	**96.31**
Yeast	57.75	58.23	**58.94**	57.71
German Credit	69.32	72.67	**74.54**	**74.54**
Tic-Tac-Toe	78.89	**84.41**	**84.41**	83.28
Vehicle	65.58	72.34	**73.86**	**73.86**
Australian Cr.	85.94	85.94	**86.66**	**86.66**
Balance Scale	93.33	94.57	**95.08**	93.33
Ionosphere	86.94	**90.32**	**90.32**	**90.32**
Liver	66.68	69.04	**70.21**	69.04
Ecoli	81.53	82.06	**82.88**	80.98
Glass	68.43	69.11	**70.19**	68.43
Wine	**85.62**	**85.62**	**85.62**	82.21
Iris	**96.00**	**96.00**	**96.00**	**96.00**
Lymphography	85.81	86.04	**86.42**	85.32
Hayes Roth	40.23	**41.01**	**41.01**	33.33
Satellite	80.30	80.30	**82.37**	77.00
Haberman	40.52	43.28	43.28	**57.40**

In Table 2-1, the best results for each dataset are highlighted in bold. Experiments were run on 19 datasets. Our proposed Sum-Over-l_2-norm regularization gave the best results 17 times. Results from our Elastic Net regularization closely followed our Sum-Over-l_2-norm regularization. The recognition results from the Lasso regularization [1] were better than our methods for one case (Segmentation).

For the face recognition experiments, we repeat the experimental set-up in [1]. The experiments are carried on the Extended Yale B Face Database. For each subject, we randomly select half of the images for training and the other half for testing. Table 2-2 contains the results for face recognition. The features are selected using the Eigenface method. To compare our results with [1], we select the same number of Eigenfaces as proposed in [1].

Table 2-2: Recognition Accuracies on Extended Yale B

Method	Number of Eigenfaces			
	30	56	120	504
Lasso [1]	86.49	91.71	93.87	96.77
Elastic Net	86.96	92.05	94.26	97.13
Sum-Over-l_2-norm	**89.40**	**93.37**	**95.14**	**97.79**
NN	74.48	81.85	86.08	89.47

The best recognition results are highlighted in bold. It is seen from Table 2-2 that our proposed Sum-Over-l_2-norm regularization gives the best recognition results for any number of Eigenfaces selected. The Elastic Net regularization is little lower than the Sum-Over-l_2-norm regularization but better than the Lasso.

2.5 Conclusion

A novel classification assumption: states that the training samples of a class approximately form a linear basis for any new test sample" was proposed in [1]. Based on this assumption, a classifier using LASSO regularization was built. We argued that the Lasso regularization is not an ideal choice for the classifier based on the aforesaid assumption as it selects one training sample only to form the basis. To select a basis with many training samples, we proposed two alternate regularizations techniques, Elastic Net and Sum-Over-l_2-norm for selecting a group of samples. Results on 20 different datasets (19 from the UCI ML repository and Yale Face Database) show that the sparse classifier based on our alternate regularizations yield better recognition results.

The previous work [1] used the sparse classifier for face recognition only. We however, show that the sparse classifier can be used for general purpose classification tasks including face recognition. Using many benchmark datasets from the UCI ML repository, our proposed sparse classifiers is shown to consistently outperform the classifier in [1] and also the NN classifier (see Table 2-1). For face recognition tasks, our proposed methods, on average, yield around 2% and 11% better recognition accuracy than the classifier in [1] and NN respectively.

2.6 References

[1] Y. Yang, J. Wright, Y. Ma and S. S. Sastry, “Feature Selection in Face Recognition: A Sparse Representation Perspective”, IEEE Trans. PAMI, (to appear).

[2] http://cvc.yale.edu/projects/yalefacesB/yalefacesB.html

[3] D. Donoho, “For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution,” Comm. on Pure and Applied Math, Vol. 59 (6), pp. 797–829, 2006.

[4] http://www.cs.ubc.ca/labs/scl/spgl1/

[5] H. Zou and T. Hastie, “Regularization and variable selection via the elastic net”, Journal of Royal Statistical Society B., Vol. 67 (2), pp. 301-320.

[6] C. De Mol, E. De Vito, and L. Rosasco, “Elastic-Net Regularization in Learning Theory”, eprint arXiv:0807.3423

[7] http://cran.r-project.org/web/packages/elasticnet/index.html

[8] M. Stojnic, F. Parvaresh and B. Hassibi, “On the reconstruction of block-sparse signals with an optimal number of measurements”, eprint arXiv:0804.0041v1

[9] E. van Den Berg, M. Schmidt, M. P. Friedlander and K. Murphy, “Group Sparsity via Linear-Time Projection”, Technical Report TR-2008-09, Department of Computer Science, University of British Columbia.

[10] http://archive.ics.uci.edu/ml/

CHAPTER 3: FAST GROUP SPARSE CLASSIFICATION[2]

3. 1 Introduction

Recently a new classifier was proposed [1]. It assumes that the training samples of a particular class approximately form a linear basis for a new test sample belonging to the same class. The logic behind this assumption is already mentioned in Chapter 1. This assumption can be represented formally. If $v_{k,test}$ is the test sample belonging to the kth class then,

$$v_{k,test} = \alpha_{k,1}v_{k,1} + \alpha_{k,2}v_{k,2} + ... + \alpha_{k,n_k}v_{k,n_k} + \varepsilon \qquad (1)$$

where $v_{k,i}$ are the training samples of the kth class and ε is the approximation error.

Equation (1) expresses the assumption in terms of the training samples of a single class. Thus can be expressed in terms of all the classes

$$v_{k,test} = V\alpha + \varepsilon \qquad (2)$$

where $V = [v_{1,1} | ... | v_{1,n_1} | ... | v_{k,1} | ... | v_{k,n_k} | ... v_{C,1} | ... | v_{C,n_C}]$ and

$$\alpha = [\underbrace{\alpha_{1,1},...,\alpha_{1,n_1}}_{\alpha_1}, \underbrace{\alpha_{2,1},...,\alpha_{2,n_2}}_{\alpha_2}, ... \underbrace{\alpha_{C,1},...,\alpha_{C,n_C}}_{\alpha_C}]^T .$$

where $v_{k,i}$ is the ith training sample of the kth class.

The above assumption demands that α should be 'group sparse' - meaning that the solution of the inverse problem (2) should have non-zero coefficients corresponding to a particular group of training samples

and zero elsewhere (i.e. $\alpha_i \neq 0$ for only one of the α_i's, i=1,…,C). This requires the solution of

$$\min_{\alpha} \| \alpha \|_{2,0} \text{ such that } \|v_{test} - V\alpha\|_2 < \varepsilon \qquad (3)$$

[2] A version of this chapter has been accepted for publication. A. Majumdar and R. K. Ward, "Fast Group Sparse Classification", IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, Victoria, B.C., Canada, August 2009.

The mixed norm $||\bullet||_{2,0}$ is defined for $\alpha = [\underbrace{\alpha_{1,1},...,\alpha_{1,n_1}}_{\alpha_1},\underbrace{\alpha_{2,1},...,\alpha_{2,n_2}}_{\alpha_2},...\underbrace{\alpha_{C,1},...,\alpha_{C,n_C}}_{\alpha_k}]^T$.

as $||\alpha||_{2,0}=\sum_{l=1}^{C} I(||\alpha_l||_2>0)$, where $I(||\alpha_l||_{2,0}>0)=1$ if $||\alpha_l||_{2,0}>0$. Solving (3) is an NP hard problem; to overcome this problem [1] proposes a convex relaxation of the above problem by solving the following problem

$$\min_{\alpha} ||\alpha||_{2,1} \text{ such that } ||\mathrm{v}_{test} - V\alpha||_2 < \varepsilon \qquad (4)$$

where $||\alpha||_{2,1}=||\alpha_1||_2 + ||\alpha_2||_2 +...+ ||\alpha_C||_2$.

The conditional equivalence of the $||\bullet||_{2,0}$ and $||\bullet||_{2,1}$ minimization has been studied previously [2]. Even though in many cases (4) is a good approximation of (3), the computational time required for solving (4) is large. Consequently the time required for classification is also large. To overcome this problem, in this paper we propose greedy approximate solutions to (3).

We propose 12 greedy algorithms, with various degrees of accuracy and computational complexity that will approximate the solution to (3). They are based on the Orthogonal Matching Pursuit method. All of them are faster than the GSC [1]. Among these, eight algorithms are as accurate as the previous GSC. Five of these algorithms are significantly faster (2 orders of magnitude) than the GSC.

The idea of approximating a test sample by a linear combination of the training samples of a class was originally proposed in [3]. However the optimization problem in [3] did not account for group-sparsity (more discussion on this topic can be found in [1]). Therefore the recognition results from [3] were a little worse than [1].

The rest of the paper consists of several sections. The following section discusses the background of the problem. The proposed algorithms are described in Section 3.3. In Section 3.4 the experimental results are shown. Finally in Section 3.5, conclusions and future scope of work are discussed.

3. 2 Background

3.2.1 Group Sparse Classification

Group Sparse Classification (GSC) [1] assumes that a test sample can be represented as a linear combination of the training samples belonging to its correct class. We repeat equation (2)

$$v_{test} = V\alpha + \varepsilon$$

where $V = [v_{1,1} | \ldots | v_{n,1} | \ldots | v_{k,1} | \ldots | v_{k,n_k} | \ldots v_{C,1} | \ldots | v_{C,n_C}]$ and

$$\alpha = [\underbrace{\alpha_{1,1}, \ldots, \alpha_{1,n_1}}_{\alpha_1}, \underbrace{\alpha_{2,1}, \ldots, \alpha_{2,n_2}}_{\alpha_2}, \ldots \underbrace{\alpha_{C,1}, \ldots, \alpha_{C,n_C}}_{\alpha_C}]^T .$$

As discussed above, a group sparse solution α, can be obtained by

$$\min_{\alpha} \| \alpha \|_{2,0} \text{ such that } \| \mathrm{v}_{test} - V\alpha \|_2 < \varepsilon$$

Since this is an NP hard problem, a convex relaxation of the above optimization is employed. This is represented in equation (4)

$$\min_{\alpha} \| \alpha \|_{2,1} \text{ such that } \| \mathrm{v}_{test} - V\alpha \|_2 < \varepsilon$$

where $\| \alpha \|_{2,1} = \| \alpha_1 \|_2 + \| \alpha_2 \|_2 + \ldots + \| \alpha_C \|_2$

The $\| \bullet \|_{2,1}$ minimization is the actual work-horse behind the GSC [1]. We repeat the GSC algorithm [1] here for the sake of completeness:

1. Solve the optimization problem expressed in (4).

2. Find the i's for which $\|\alpha_i\|_2 > 0$.

3. For all classes satisfying the condition in step 2, repeat the following two steps:

    a. Reconstruct a sample for each class by a linear combination of the training samples in that class via the equation $v_{recon}(i) = \sum_{j=1}^{n_i} \alpha_{i,j} v_{i,j}$.

b. Find the error between the reconstructed sample and the given test sample by $error(v_{test}, i) = \| v_{test} - v_{recon(i)} \|_2$

4. Once the $error(v_{test}, i)$ for every class i is obtained, choose the class having the minimum error as the class of the given test sample.

The convex relaxation of the NP hard problem is still slow for practical classification purposes. In this paper, we propose 12 greedy algorithms (based on the Orthogonal Matching Pursuit method) to approximate the $\|\bullet\|_{2,0}$ minimization problem. But before proposing our greedy group sparse algorithms, it will be appropriate to briefly discuss about the sparse optimization – the precursor of group sparse optimization.

3.2.2 Sparse Optimization

Group sparse optimization is a successor of the problem of sparse optimization. Ideally, sparse optimization takes the form

$$\min_x \| x \|_0 \text{ such that } \| b - Ax \|_2 < \varepsilon \qquad (5)$$

Where $\| x \|_0$ is defined as the number of non-zero terms in the vector 'x'.

Minimizing the l_0 norm finds widespread applications in signal processing (compressed sensing) [4] and machine learning (regression) [5].

Solving the l_0-norm is an NP hard problem. There are two approaches to approximate (5) – convex optimization and greedy algorithms. In the former, the l_0-norm is replaced by its nearest convex surrogate the l_1-norm and the resulting form is solved by quadratic programming techniques. Greedy (sub-optimal) algorithms try to find an approximate solution for (5).

Researchers in signal processing and machine learning use two slightly varied versions of the l_1-norm minimization problem. In signal processing the following form, called basis pursuit denoising (BPDN) is commonly employed

$$\min_x \| x \|_1 \text{ such that } \| b - Ax \|_2 < \varepsilon \qquad (6)$$

Where as in machine learning, the Least Angle Shrinkage and Selection Operator (LASSO) form is more common

$$\min_x \| b - Ax \|_2 \text{ such that } \| x \|_1 < \tau \quad (7)$$

For particular choices of ε and τ, equations (6) and (7) are equivalent and can be solved by quadratic programming in the form

$$\min_x \| b - Ax \|_2^2 + \lambda \| x \|_1 \quad (8)$$

A seminal study [6] studies the conditions under which the l_1-norm minimization is equivalent to the l_0-norm minimization. Solving the l_1 norm minimization problem is computationally expensive. Where as greedy approximate algorithms for solving (5) are computationally fast. They do not have the same theoretical guarantees as of convex optimization, but practically they have been found to be quite accurate.

There are two classes of greedy approaches for solving (5) – Matching Pursuit [7-10] and Thresholding [11]. Both are iterative schemes. The former starts with a zero vector and identifies the position of the sparse coefficients at each iteration and their values thereof. The latter starts with a full vector and at each iteration thresholds the coefficients to arrive at a sparse solution.

The matching pursuit method is suited when selecting grouped variables. Our algorithms are all modifications of the Orthogonal Matching Pursuit [9] method. We will discuss our proposed methods in Section 3.3. But before that we will briefly discuss the convex group sparse optimization methods that have been developed so far.

3.2.3 Group Sparse Classification

The problem of group sparsity is relatively new. Researchers in machine learning became interested in the problem mainly for applications in regression [12]. The signal processing community is more interested in the theoretical aspects of the problem [2]; mainly the conditions under which the convex optimization yields reasonable results.

The group sparsity optimization is an extension of the LASSO method [8] popularly known as Group LASSO. For the following inverse problem

$$b = Ax + \eta,\ \eta \to normal\ N(0,\sigma) \tag{9}$$

and $x = [\underbrace{x_{1,1},...,x_{1,n_1}}_{x_1},\underbrace{x_{2,1},...,x_{2,n_2}}_{x_2},...\underbrace{x_{k,1},...,x_{k,n_k}}_{x_k}]^T$

then Group LASSO takes the form

$$\min_x \|b - Ax\|_2 \text{ such that } \sum_{j=1}^{k} \|x_j\|_2 < \tau \tag{10}$$

In [12], three algorithms to solve (10) – Group LASSO, Group LARS and Group Non-negative Garrote are discussed in detail. Recently a faster implementation to the above problem has been proposed [13].

3. 3 Greedy Group Sparsity

Previous studies indicate that greedy sparsity promoting algorithms like Orthogonal Matching Pursuit (OMP) and its variants are much faster than the solutions obtained via convex optimization. Until now there are only a few scattered greedy group sparse algorithms like Block Orthogonal Matching Pursuit [14] and Group Matching Pursuit [15]. In this paper we propose 12 variants of greedy group sparsity promoting algorithms within the OMP framework. Therefore before proceeding into our algorithms we analyze the OMP briefly. The OMP is a greedy solution to (5).

OMP Algorithm

Given – Matrix A and vector b.

Required – estimate sparse vector x

Initialize – t = 1; $r_0 = y$; $L_0 = []$;

Repeat until a stopping criterion met

```
1. $c_t = A^T r_{t-1}$
2. $l_t = \{j: \max|c_t(j)|\}$
3. $L_t = [L_{t-1}\ l_t]$
4. $x_t = \arg\min_x ||y-A_{Lt}x_t||_2$
5. $r_t = y - Ax_t$
```

The OMP algorithm finds a greedy solution to the sparse optimization problem (6). Initially (i.e. at t=1) the residual (r_0) is set to the vector b and the index set L is empty. Steps 1 to 5 are repeated until some stopping criterion is met. Generally the stopping criterion is a minimum bound on the residual or the maximum number of steps.

In step 1 of the t^th iteration a vector (c) is formed by multiplying the matrix A^T with the current residual. This step basically finds the correlation between the current residual and each column of A^T. The index with the highest correlation is selected in step 2. This index is added to the current index set L in step 3. A_L represents the columns of A indexed in L. The current signal estimate is obtained via least squares as shown in step 4. In step 5 of the OMP algorithm, the residual for the next iteration is obtained.

There are three major stages in each OMP iteration – the selection stage (steps 1-2); the minimization stage (step 4) and the update (stage 5). There are several variants of the OMP algorithm based on the variations in these stages. Greedier algorithms like Stagewise OMP (StOMP) [10] and Regularized OMP (ROMP) [9] selects more than one index in the selection stage. In Matching Pursuit (MP) [7] the minimization stage is skipped altogether. Therefore MP is really fast but not quite as accurate as OMP. In a recent study [16] approximate fast algorithms based on gradient updates have been proposed to reduce the complexity of OMP.

3.3.1. Greedy Algorithms for Group Sparsity

The main contribution of this paper is the approximate solution of equation (3).

$$\min_x ||\, x\, ||_{2,0} \text{ such that } ||y - Ax\, ||_2 < \varepsilon$$

where $x = [\underbrace{x_{1,1},...,x_{1,n_1}}_{x_1},\underbrace{x_{2,1},...,x_{2,n_2}}_{x_2},...\underbrace{x_{k,1},...,x_{k,n_k}}_{x_k}]^T$

In addition to the vector y and the matrix A, the group label, for each column in A are also needed.

We will now discuss different greedy solutions to the above group sparsity problem (equation (3)). The first proposed greedy algorithm to solve (3) is the Block Orthogonal Matching Pursuit (BOMP) [14]. It was first proposed by the name Group Matching Pursuit (GMP) in [15]. In this work the name GMP refers to a different algorithm.

BOMP Algorithm

Given – Matrix A, vector y and group labels of each column in A.

Required – estimate a block sparse vector x

Initialize – t = 1; r_0 = y; L_0 = [];

Repeat until a stopping criterion is met

1. $c_t = A^T r_{t-1}$
2. l_t = {j: group(max|c_t(j)|)}
3. $L_t = [L_{t-1}\ l_t]$
4. $x_t = \arg\min_x ||y-A_{Lt}x_t||_2$
5. $r_t = y – Ax_t$

The BOMP algorithm is similar to the OMP algorithm except for step 2. In this step, instead of selecting only the index having the highest correlation (as in OMP) the indices of the entire group containing the highest correlation is selected.

We propose an alternate algorithm called the Group Orthogonal Matching Pursuit (GOMP). The group selection criterion in GOMP is slightly different from that of BOMP.

GOMP Algorithm

```
Given – Matrix A, vector y and group labels of each column in A.
Required – estimate a block sparse vector x
Initialize – $t = 1$; $r_0 = y$; $L_0 = []$;
Repeat until a stopping criterion is met
1. $c_t = A^T r_{t-1}$
2. $I_t = \{j: group(max(group\text{-}mean|(c_t(j)|))\}$
3. $L_t = [L_{t-1}\ I_t]$
4. $x_t = \arg\min_x ||y-A_{Lt}x_t||_2$
5. $r_t = y – Ax_t$
```

GOMP differs from the previous algorithm only in step 2. In the Group version of the algorithm, at each iteration the group average of the correlations are calculated and the group with the highest average is selected. In the rest of the paper whenever an algorithm's name starts with 'Block' it will mean that selection strategy is similar to BOMP, i.e. the groups are selected based on the highest individual correlation. Also when some algorithm's name starts with 'Group' we mean that the groups are selected based on the highest average group-wise correlation.

The OMP based algorithms require the solution of a least squares problem (step 4). Depending upon the size of the problems this may be expensive. Now we will discuss some algorithms that reduce the computational cost of the algorithm by either skipping that step (matching pursuit) or approximately solving that step (gradient pursuit).

First we discuss the Block Matching Pursuit (BMP); this is based on the Matching Pursuit (MP) algorithm [7]. It is the cheapest (least computational effort) possible group sparse estimation algorithm.

BMP Algorithm

```
Given – Matrix A, vector y and group labels of each column in A.
```

```
Required – estimate a block sparse vector x
Initialize – $t = 1$; $r_0 = y$;
Repeat until a stopping criterion is met
    1. $c_t = A^T r_{t-1}$
    2. $I_t = \{j: group(\max|c_t(j)|)\}$
    3. $x_t(I_t) = x_{t-1}(I_t) + c_t(I_t)$
    4. $r_t = r_{t-1} – A(I_t)x_t(I_t)$
```

The first step is similar to the OMP based algorithms. It calculates the correlations between the current residual and the columns of A^T. In the next step it selects the group indices having the highest correlation according to the 'Block' selection criterion. In step 3, it only updates the sparse estimate at the group indices selected in the previous step. Step 4 updates the residual by subtracting the product of the matrix A and the sparse estimate x indexed in step 2 from the previous residual.

As before, we can also have a Group Matching Pursuit (GMP) in which the indices are selected in step 2 according to the 'Group' selected method. Note that our proposed GMP algorithm bears no semblance with the so called Group MP in [15]. As mentioned earlier, their algorithm is actually BOMP.

Although the MP based algorithms are very fast, they are relatively inaccurate compared to those based on OMP. There is always a trade-off between computational expense and accuracy. A recent work [16] proposes gradient pursuit algorithms to 'approximately' solve the least squares problem in OMP to keep the computational cost low (but at the same time does not compromise much on the accuracy).

There are different ways to approximate the least squares problem in the gradient pursuit framework. We give the Block version of the general gradient pursuit algorithm called Block Gradient Pursuit.

BGP Algorithm

```
Given – Matrix A, vector y and group labels of each column in A.

Required – estimate a block sparse vector x

Initialize – t = 1; $r_0 = y$; $L_0 = []$;

Repeat until a stopping criterion is met

1.  $c_t = A^T r_{t-1}$

2.  $I_t = \{j: group(max|c_t(j)|)\}$

3.  $L_t = [L_{t-1}\ I_t]$

4.  Calculate update direction $d_{Lt}$

5.  $u_t = A_{Lt} d_{Lt}$

6.  $a_t = u_t' r_t / ||u_t||_2$

7.  $x_{Lt} = x_{Lt-1} + a_t d_{Lt}$

8.  $r_t = r_{t-1} – a_t u_t$
```

The gradient pursuit algorithm has update steps similar to MP (steps 7 and 8). But the update is more accurate compared to MP. All the steps apart from 4, in the BGP algorithm are self explanatory. There can be three different ways to update the gradient direction in step 4; this gives rise to three different algorithms [16] – gradient pursuit (GP), partial conjugate gradient pursuit (PCGP) and approximate conjugate gradient pursuit (ACGP) or nearly orthogonal matching pursuit (NOMP). Due to limitations in space, we can not go into the details of each algorithm. We can have separate ‘Block’ and ‘Group’ versions of each algorithm.

We have been discussing how to reduce the complexity of the least squares step in the OMP based algorithms. There is another way to reduce the overall complexity of the algorithm, i.e. in each step the new algorithms may be more expensive than OMP based ones but will terminate faster. This is achieved by selecting multiple groups in each iteration instead of selecting only one. We propose two such algorithms – Stagewise Block Orthogonal Matching Pursuit (StBOMP) and Regularized Group Orthogonal Matching Pursuit (ReGOMP).

StBOMP Algorithm

```
Given – Matrix A, vector y and group labels of each column in A.

Required – estimate a block sparse vector x

Initialize – $t = 1$; $r_0 = y$; $L_0 = []$;

Repeat until a stopping criterion is met

1.  $c_t = A^T r_{t-1}$

2.  $I_t = \{j: group(|c_t(j)| > k_t \alpha_t)\}$

3.  $L_t = [L_{t-1}\ I_t]$

4.  $x_t = \arg\min_x ||y - A_{Lt} x_t||_2$

5.  $r_t = y – Ax_t$

$\sigma_t = ||r_s||_2 / n^{½}$ and $2<k<3$
```

StBOMP is a greedier algorithm than BOMP. It is based on the concepts of StOMP [10]. At the selection step 2, it greedily selects more than one group at a time. It selects all the groups that have correlations greater than a threshold. At each step the StBOMP algorithm selects more than one group and consequently terminates faster than the previous ones.

Our last algorithm is the Regularized Group Orthogonal Matching Pursuit (ReGOMP). This is also a greedy algorithm like StOMP – it selects more than one group at each iteration. But this algorithm is more cautious, it selects the groups based on a regularization test. Consequently the algorithm is more accurate than StBOMP.

ReGOMP Algorithm

Given – Matrix A, vector y, group labels of each column in A and s = estimate of the # sparse groups.

Required – estimate a block sparse vector x

```
Initialize – $t = 1$; $r_0 = y$; $L_0 = []$;

Repeat until a stopping criterion is met

1. $c_t = A^T r_{t-1}$
2. $g_t(k) = \text{sort}(\text{group-mean}(c_t(j)))$; $k = 1$ to # groups
3. $J = \{j: \text{group}(g_t(1:s))\}$
4. Among all subsets $J_0$ belonging to J perform regularization: $|g_t(m)| < 2|g_t(n)|$ for all m, n in $J_0$.
   Among all the $J_0$'s satisfying the regularization, choose the one with maximum energy.
5. $L_t = [L_{t-1}\ J0]$
6. $x_t = \arg\min_x \|y - A_{Lt} x_t\|_2$
7. $r_t = y – Ax_t$
```

The ReGOMP's regularization criterion has been discussed in the ROMP algorithm [9]. Step 1 is similar to all other algorithms discussed so far. In step 2, the group wise mean is stored in a vector after sorting. The top 's' groups are selected in J (step 3). Step 4 performs the regularization. Step 5, selects the indices of the regularized groups. Step 6 solves the least squares problem for the selected groups. Finally in step 7 the residual is updated based on the current signal estimate.

We have implemented all the algorithms discussed here in Matlab. It is named GroupSparseBox and the latest version can be downloaded from [26].

3. 4 Experimental Evaluation

In Section 3.2.1 the (GSC) classification algorithm proposed in [1] is discussed. Step 1 of GSC requires the solution of the NP hard problem (3). In [1] a convex relaxation of (3) is employed. In this work, we replace the convex optimization by our greedy group sparse algorithms. We call our classifiers as Fast Group Classification (FGSC) algorithms. The FGSC algorithms are compared with GSC [1] is terms of recognition accuracy and speed.

This study uses some benchmark classification datasets from the UCI Machine Learning repository. We use the same datasets and experimental procedure as in [1].

For each of the algorithms, we perform a simple non-parametric t-test to test the hypothesis that the mean classification accuracy from the FGSC is significantly different from GSC. The results are in Table 3-1.

Table 3-1: Results of Hypothesis Testing

FGSC Algorithm	95% Significance	99% Significance
BMP	Yes	No
GMP	Yes	No
BOMP	Yes	Yes
GOMP	Yes	Yes
BGP	Yes	No
GGP	Yes	No
BNOMP	Yes	Yes
GNOMP	Yes	Yes
BPCGP	Yes	Yes
GPCGP	Yes	Yes
StBOMP	Yes	Yes
ReGOMP	Yes	Yes

The results of Hypothesis Testing (Table 3-1) shows that at 95% significance level, the results from all the FGSC algorithms are the same as GSC. As the significance level is increased (made stricter) – at 99%, differences between the FGSC and the GSC algorithms are observed.

As the name suggests – FGSC is significantly faster than GSC. The values in Table 3-2 reflect the run times required for one classification by the different FGSC algorithms relative to GSC. In

Table 3-2 the datasets are arranged such that the size of the dataset decreases from top to bottom.

Tables 3-1 and 3-2 indicate that 5 algorithms – BOMP, GOMP, BNOMP, GNOMP and ReGOMP are the fastest and most accurate (in terms of recognition accuracy compared with GSC) algorithms. The computational advantage of the FGSC algorithms become more evident as the size of the dataset increases.

Table 3-2: FGSC Run-Times as Percentage of GSC Run-Time

Dataset	BMP	GMP	BOMP	GOMP	BGP	GGP	BNOMP	GNOMP	BPCGP	GPCGP	StBOMP	ReGOMP
Satellite	4.97	5.63	0.72	0.95	11.07	8.55	0.19	0.19	9.91	9.17	17.78	0.47
Page Block	8.71	10.99	0.94	1.03	3.86	3.16	0.22	0.23	8.55	7.38	5.43	0.69
Abalone	2.34	3.93	1.05	1.20	4.51	3.68	0.78	0.78	23.82	12.59	16.44	1.01
Segment.	3.66	3.80	1.54	1.57	17.69	4.28	0.25	0.25	17.80	4.29	20.11	0.87
Yeast	17.46	20.10	8.34	9.03	79.17	23.92	4.08	4.80	81.89	23.13	71.25	3.95
German Cr.	2.79	3.94	1.17	1.31	6.23	4.75	0.23	0.33	6.57	4.79	3.66	0.73
Tic-Tac-Toe	12.59	18.37	1.86	2.49	29.07	21.43	2.22	3.29	31.20	21.10	13.82	1.49
Vehicle	14.51	21.73	1.76	2.27	32.56	24.75	5.72	9.45	32.56	25.38	15.49	1.53
Aus. Cr.	5.37	31.29	2.14	4.05	18.97	37.62	7.62	18.30	20.63	30.11	13.29	2.39
Bal. Scale	5.86	30.82	1.98	4.64	10.27	39.51	7.14	20.81	11.64	33.27	11.86	3.06
Ionosph.	9.27	42.56	2.33	9.15	13.56	52.67	9.53	33.76	21.81	47.91	10.64	5.78
Liver	9.83	40.11	2.94	10.71	16.85	51.74	9.72	39.64	19.62	51.79	7.51	4.62
Ecoli	8.08	45.69	2.58	8.54	14.63	49.88	10.90	35.57	17.57	50.32	8.96	4.81
Haberman	8.66	44.13	3.11	9.62	15.83	51.10	11.65	38.47	9.76	52.17	8.62	5.08
Glass	9.24	42.57	3.64	10.70	17.03	52.31	12.40	41.36	1.94	54.01	8.27	5.35
Wine	11.60	52.76	2.76	9.66	15.74	66.85	14.91	45.30	20.44	68.50	11.87	5.81
Iris	13.02	66.28	1.53	6.13	11.11	52.10	12.26	38.69	16.09	54.78	6.51	6.89
Lymphogr.	9.61	51.28	1.92	7.37	12.50	58.97	13.14	43.91	17.94	63.14	6.70	6.08
Hayes Roth	10.43	58.58	2.35	10.43	13.47	61.61	13.13	44.78	18.18	62.62	6.39	7.74

Table 3-2 shows that the BOMP, GOMP, BNOMP, GNOMP and the ReGOMP are the fastest algorithms. Computational speed from the aforementioned greedy algorithms is two orders of magnitude higher than convex optimization.

3. 5 Conclusion

This work proposes novel algorithms for greedy group sparsity promoting optimization. These algorithms are applied to the problem of group sparsity promoting classification. Experimental results indicate that our greedy algorithms are as accurate as the previous one [1] but are significantly faster. Apart from the Matching Pursuit (GMP, BMP) and the Gradient Pursuit (GGP, BGP) methods, all other algorithms show the same accuracy as GSC even at 99% confidence interval. The GNOMP and the BNOMP algorithms are the fastest of all (closely followed by GOMP, BOMP and the ReGOMP). For large databases they are faster by more than 2 orders of magnitude.

We have applied the group sparse algorithms for the classification problem. Such algorithms however also finds applications in other areas – signal processing [2, 14], communications [15] and regression [12]. We encourage the interested reader to use our algorithms [17] to these problems.

3. 6 References

[1] A. Majumdar and R. K. Ward, “Classification via Group Sparsity Promoting Regularization”, IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 873-876, 2009

[2] Y. C. Eldar and M. Mishali, “Robust recovery of signals from a union of subspaces,” IEEE Transactions on Information Theory (submitted.).

[3] Y. Yang, J. Wright, Y. Ma and S. S. Sastry, “Feature Selection in Face Recognition: A Sparse Representation Perspective”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 31 (2), pp. 210-227, 2009.

[4] S. S. Chen, D. L. Donoho and M. A. Saunders, “Atomic Decomposition by Basis Pursuit”, SIAM J. Sci. Comp., Vol. 20 (1), pp. 33-61, 1998.

[5] R. Tibshirani, “Regression shrinkage and selection via the lasso”. J. Royal. Statist. Soc B., Vol. 58 (1), pp. 267-288, 1996.

[6] D. L. Donoho, “For most large underdetermined systems of linear equations the minimal l1-normsolution is also the sparsest solution”, Comm. Pure Appl. Math., 59, pp. 797-829, 2006.

[7] S. G. Mallat and Z. Zhang, Matching Pursuits with Time-Frequency Dictionaries , IEEE Trans. Sig. Proc. pp. 3397-3415, 1993.

[8] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad. “Orthogonal Matching Pursuit: Recursive function approximation with applications to wavelet decomposition”, Asilomar Conf. Sig., Sys., and Comp., Nov. 1993.

[9] D. Needell and R. Vershynin, “Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit,” Submitted, 2007.

[10] D.L. Donoho, Y. Tsaig, I. Drori, J.-L. Starck, “Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit”, preprint http://www-stat.stanford.edu/~idrori/StOMP.pdf

[11] M. Fornasier and H. Rauhut, “Iterative thresholding algorithms”, (Preprint, 2007).

[12] M. Yuan and Y. Lin, “Model selection and estimation in regression with grouped variables”, J. R. Statist. Soc. B, 68, pp. 49-67, 2006.

[13] E. van den Berg, M. Schmidt. M. Friedlander, K. Murphy, “Group sparsity via linear-time projection”, Technical Report TR-2008-09, Department of Computer Science, University of British Columbia, June 2008.

[14] Y. C. Eldar and H. Bolcskei, "Block Sparsity: Uncertainty Relations and Efficient Recovery," (accepted) ICASSP09.

[15] C. J. Wu and D. W. Lin, "A Group Matching Pursuit Algorithm for Sparse Channel Estimation for OFDM Transmission," IEEE International Conference on Acoustics, Speech, and Signal Processing , pp. 429-432, 2006.

[16] T. Blumensath and M. E. Davies, "Gradient Pursuits," IEEE Transactions on Signal Processing Vol.56 (6), pp.2370-2382, 2008.

[17] http://www.mathworks.com/matlabcentral/fileexchange/22835

CHAPTER 4: NEAREST SUBSPACE CLASSIFIER[3]

4. 1 Introduction

In a classification problem, the task is to find the class of the test sample given the classes of the training samples. This work makes a novel classification assumption – samples from each class lie on a hyper-plane specific to that class. According to this assumption, the training samples of a particular class span a subspace. When a sample becomes available, the problem is to determine, to which subspace it belongs.

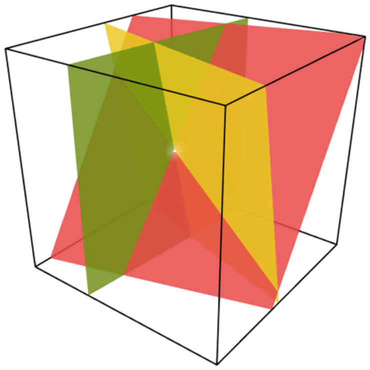

Figure 4-1: Subspaces

Let us consider a three-class classification problem, where each input is represented by a 2 dimensional vector. Figure 4-1. shows the three two dimensional subspaces (corresponding to three classes) embedded in three dimensional Euclidean space. According to our assumption, samples from each class will lie in one of these 3 subspaces.

Generally, the term ‘subspace’ is associated with ‘dimensionality reduction’ in machine learning where the problem is to project the original high dimensional data to a lower dimension (subspace) in a smart fashion so that certain properties of the original data are maintained. Our work is completely different and is in no way related to dimensionality reduction. In our work the subspaces participate in the classification.

[3] A version of this chapter has been submitted for publication. A. Majumdar and R. K. Ward, “Nearest Subspace Classifier”.

The assumption for the Nearest Subspace Classifier (NSC) is a simplification of the assumption made in Chapters 1 and 2. According to the previous assumption, the training samples can lie on different subspaces and the test sample can be expressed as a union of these subspaces. The assumption for NSC is more restrictive, it assumes that all the training samples lie on the same subspace. We will see in Section 4.2, the slight simplification in assumption leads to huge reduction in computational time.

The proposed classification assumption appears restrictive, but it has a distinct advantage over previous classifiers in the 'small number of training samples' scenario. Most previous classifiers (NN, SVM, ANN) work well when the number of training samples is much larger than the dimensionality of the inputs. But our proposed classifier works well when the number of training samples is less than the dimensionality of the inputs. The 'small number of training sample' problem arises in many real world classification tasks, especially in image classification. In this work, we will show that, with this assumption the proposed Nearest Subspace Classifier (NSC) can yield better recognition results than traditional classifiers like Nearest Neighbour (NN), Support Vector Machine (SVM) and new classifier like Sparse Classifier [1].

Section 4.2 will be on our proposed classification algorithm. Section 4.3 will have the results. Finally in Section 4.4, we will discuss conclusions of this work.

4. 2 Proposed Classifier

We assume that the samples of a particular class lie in a subspace. Additionally, we assume that the training samples of each class are sufficient to span its subspace. Therefore any new test sample belonging to that class can be represented as a linear combination of the test samples, i.e.

$$v_{k,test} = \sum_{i=1}^{n_k} \alpha_{k,i} \cdot v_{k,i} + \varepsilon_k \qquad (1)$$

where $v_{k,test}$ is the test sample (i.e. the vector of features) for the kth class, $v_{k,i}$ is the ith test sample for the kth class, and ε_k is the approximation error for the kth class.

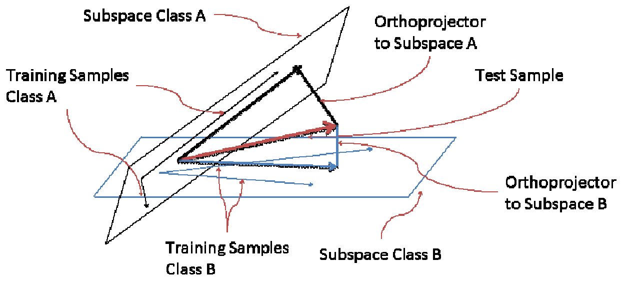

Figure 4-2: Nearest Subspace Classifier

Owing to the error term in equation (1), the relation holds for all the classes k=1:C. In such a situation it is reasonable to assume that for the correct class the test sample has the minimum error ε_k .

To find the class that has the minimum error in equation (1), the coefficients $\alpha_{k,i}$ k=1:C must be estimated first. This can be performed by rewriting (1) in matrix-vector notation

$$v_{k,test} = V_k \alpha_k + \varepsilon_k \tag{2}$$

where $V_k = [v_{k,1} \mid v_{k,2} \mid \ldots \mid v_{k,n_k}]$ and $\alpha_k = [\alpha_{k,1}, \alpha_{k,2} \ldots \alpha_{k,n_k}]^T$.

In our problem the matrix V_k is tall since we are interested in the 'small sample' problem where the dimensionality of the feature vector is typically greater than the number of training samples in each class. Equation (2) expresses an over-determined system of equations. Therefore the column-space of V_k is not rank deficient and a solution to (2) can be obtained by minimizing

$$\hat{\alpha}_k = \underset{\alpha}{\arg\min} \| v_{k,test} - V_k \alpha \|_2^2 \tag{3}$$

The analytical solution of (3) is

$$\hat{\alpha}_k = (V_k^T V_k)^{-1} V_k^T v_{k,test} \tag{4}$$

Plugging this expression in (2), and solving for the error term, we get

$$\varepsilon_k = (V_k (V_k^T V_k)^{-1} V_k^T - I) v_{k,test} \tag{5}$$

The advantage of (5) is that the expression within the brackets (the orthoprojector) can be computed before hand (for all classes) since it does not depend on the test sample.

Based on these simple derivations, we devise a very efficient algorithm for classification

Training

1. For each class ‘k’, compute the orthoprojector (the term in brackets in equation (5)).

Testing

2. Calculate the error for each class ‘k’ computing matrix vector product between the orthoprojector and $v_{k,test}$.
3. Classify the test sample as the class having the minimum error ($\| \varepsilon_k \|$).

4.2.1. Relationship with Previous Classifiers

Conceptually our classification assumption is somewhere between the Nearest Neighbor (NN) and the Sparse Classifier (SC) [1]. The NN classification algorithm is based on distances from individual samples. According to the assumption in SC [1] is based on the distance from a union of subspaces. The proposed classifier is based on distances from a subspace and can classify correctly if all the samples (for a class) lie on a single subspace. Our classifier is less generalized than SC but more generalized than NN.

The operational aspects of our algorithm are better than both NN and SC. Both of these classifiers are ‘lazy learning’ algorithms – they do not have a training phase thus all the computation is performed during run-time, where as in our case, the bulk of the work (finding the orthoprojectors for each class) is performed before the testing-task. Consequently the proposed method has a distinct operational advantage compared to NN and SC.

4.2.2. Properties of Nearest Subspace Classifier

Our classification algorithm has two important properties – i) invariance to orthogonal and tight-frame projections and ii) approximate invariance to restricted isometric projections. These invariance properties are analytically proven in the following two sub-sections.

4.2.2.1. Orthogonal and Tight-Frame Projections

For an orthogonal projection matrix, say *W* (e.g. wavelets, DCT, Fourier Transform) the following relation holds

$$WW^T = W^T W = I \tag{6}$$

The forward transform followed by the backward transform produces the same effect as obtained by applying these transforms in the reverse order.

For tight-frame projections, say *C* (e.g. curvelets, stationary wavelets)

$$C^T C = I \neq CC^T \tag{7}$$

The forward transform followed by the backward transforms preserves the original data, but the operations in reverse do not.

For orthogonal and tight-frame projections, the classification results from the proposed method will be invariant. i.e. the same as those from the original data. Let Φ be the projection matrix (orthogonal or tight-frame), the projected samples are used instead of the original ones then

$$x = \Phi v \tag{8}$$

Denote the vector of coefficients related to x's by $\beta_{k.}$ First we show that the coefficient remains the same ($\alpha_k = \beta_k$) before and after orthogonal or tight-frame projections (when x's are used instead of v's). When x's are used instead of v's, the solution of the least square error problem turns out to be

$$\hat{\beta}_k = \underset{\beta}{\arg\min} \| x_{k,test} - X_k \beta \|_2^2 \tag{9}$$

The orthoprojector will remain the same if the solution to (3) and (9) are the same.

$$\begin{aligned}\hat{\beta}_k &= (X_k^T X_k)^{-1} X_k^T x_{k,test} \\ &= (V_k^T \Phi^T \Phi V_k) V_k^T \Phi^T \Phi v_{k,test} \\ &= (V_k^T V_k) V_k^T v_{k,test} \quad \because \ \Phi^T \Phi = I \\ &= \hat{\alpha}_k\end{aligned}$$

The class-wise error is of the form

$$\begin{aligned}\delta_k &= x_{k,test} - X_k\beta_k \\ &= \Phi v_{k,test} - \Phi V_k\alpha_k \\ &= \Phi(v_{k,test} - V_k\alpha_k)\end{aligned} \tag{10}$$

The norm of the error becomes

$$\begin{aligned}\|\delta_k\|_2 &= \|\Phi(v_{k,test} - V_k\alpha_k)\|_2 \\ &= \|\Phi(v_{k,test} - V_k\alpha_k)\|_2 \\ &= \|\Phi\|_2\|(v_{k,test} - V_k\alpha_k)\|_2 \\ &= \|(v_{k,test} - V_k\alpha_k)\|_2 \quad \because \quad \|\Phi\|_2 = 1\end{aligned}$$

The error for each class is preserved, therefore the classification results too will remain as before.

4.2.2.2. Restricted Isometric Projections

A matrix *A* is supposed to be a Restricted Isometric Projection (RIP) matrix if for a vector *I*, the following holds

$$(1-\delta)\|I\|_2 \le \|AI\| \le (1+\delta)\|I\|_2$$

where, δ is small. The RIP property is defined in the Compressed Sensing [2] literature.

If RIP holds, then we can show that the classification results will be approximately preserved before and after RIP projection. The proof is similar to the previous one. First we show that the solution to the least squared error problem is approximately preserved.

Instead of using the original samples, let the RIP projected vectors $r = Av$ be used in classification. Then

$$\hat{\lambda}_k = \underset{\lambda}{\arg\min}\|r_{k,test} - R_k\lambda\|_2^2 \tag{11}$$

$$\begin{aligned}&= \underset{\lambda}{\arg\min}\|Av_{k,test} - AV_k\lambda\|_2^2 \\ &= \underset{\lambda}{\arg\min}\|A(v_{k,test} - V_k\lambda)\|_2^2 \\ &\approx (1\pm\delta)\underset{\lambda}{\arg\min}\|(v_{k,test} - V_k\lambda)\|_2^2 \approx (1\pm\delta)\hat{\alpha}_k \text{ by RIP}\end{aligned}$$

The solution to the least squared error problem is approximately the same, before and after the RIP projection. We will show that the norm of the error is also approximately preserved after RIP projection; the class-wise error now takes the form

$$\sigma_k = ((1\pm\delta)A(V_k^T V_k)^{-1}V_k^T - I)\delta_{k,test} \qquad (12)$$

and the norm of the error is

$$\begin{aligned} \| \sigma_k \|_2 &= \| ((1\pm\delta)AV_k(V_k^T V_k)^{-1}V_k^T - I)r_{k,test} \|_2 \\ &= (1\pm\delta) \| A \|_2 \| (V_k(V_k^T V_k)^{-1}V_k^T - I)v_{k,test} \|_2 \\ &= (1\pm\delta)^2 \| (V_k(V_k^T V_k)^{-1}V_k^T - I)v_{k,test} \|_2 \\ &= (1\pm 2\delta) \| \varepsilon_k \|_2 \end{aligned}$$

Since the norm of the error is approximately preserved the classification results will be approximately preserved as well.

The Compressed Sensing literature shows that dimensionality reduction by Random Projection (RP) matrices follow the RIP property for images. RP is a cheap and data-independent dimensionality reduction method scarcely used in machine learning. There has been only a few previous work in face recognition that used RP with NN [3] and SC [1] classification. Since, RP satisfies the RIP property for images, it can be used as dimensionality reduction technique prior to Nearest Subspace Classification.

4. 3 Experimental Evaluation

The proposed Nearest Subspace Classifier works well in the ‘small sample’ problem – i.e. when the number of training samples per class is less than the dimensionality of the feature vector. This scenario is typically faced in image recognition problems. In this work, we test the proposed classifier on an optical character recognition (OCR) problem.

Standard OCR datasets like USPS and MNIST have more samples than the dimensionality of the features. Moreover previous studies in OCR have reached near perfect recognition accuracy on these datasets; therefore there is not much room for improvement. In this work, we use the dataset of [4] which addresses the problem of Bengali OCR. It is a challenging dataset, and the number of samples in each class is much less than the dimensionality of the input features. The printed character recognition dataset (corpus) consists of 12 different fonts (f) and each font has 5 different sizes (s). During experimentation, the images are normalized to 16X16 pixels. The fonts were downloaded from [5].

There may be two different scenarios with printed character recognition i) Pessimistic and ii) Optimistic (realistic). In the pessimistic scenario, the classifier is trained with a subset of all the fonts while in the optimistic scenario it is trained with all the fonts. In the pessimistic scenario, the training set comprises p (<f) fonts and each font has all the s sizes; the test set has the remaining (f-p) fonts with all the s sizes. In the optimistic case the training set comprises of all the f fonts but with only q (<s) sizes and the test set consists of all the f fonts but with the remaining (s-q) font sizes.

For our experiments f = 12, s = 5, p = 9 and q = 3. For the pessimistic scenario, we created 5 different datasets. Each training set is formed by randomly selecting 9 of the 12 fonts from the corpus and the corresponding testing set consists of the remaining 3 fonts. For the optimistic scenario we created 3 different datasets. Each dataset is formed by randomly selecting 3 font sizes from the corpus for training and the remaining 2 font sizes for testing.

We compare our classification results with three other classifiers – Nearest Neighbor, Sparse Classifier [1] and Support Vector Machine. In Table 4-1 and 4-2, the results for the pessimistic and the optimistic scenarios are provided respectively. The inputs for these experiments were the raw pixel values. The best results are shown in bold.

Table 4-1: Recognition Error for Pessimistic Scenario

Set No.	KNN	Proposed	SC	SVM
1	0.1529	**0.0667**	0.1085	0.1200
2	0.3660	**0.2157**	0.2967	0.3507
3	0.4065	**0.2549**	0.3098	0.3720
4	0.4157	**0.3150**	0.3699	0.3853
5	0.2706	**0.2065**	0.2314	0.2507

Table 4-2: Recognition Error for Optimistic Scenario

Set No.	KNN	Proposed	SC	SVM
1	0.0253	**0.0074**	0.0114	0.0204
2	0.0376	**0.0082**	0.0188	0.0302
3	0.0596	**0.0245**	0.0384	0.0474

Tables 4-1 and 4-2 show that our classifier outperforms all classifier for every set by a large margin. The reason SVM fails to yield good results is probably due to the scarcity of samples. Ideally, SC should have given the same accuracy as the proposed classifiers, since the former is based on a more powerful assumption – classification from a union of subspaces. However, in practice SC is only little better than NN [6]; therefore it falls behind the Nearest Subspace Classifier (NSC) in practice.

The previous experiment used pixel values as features for recognition. In Section 4.2 we proved that the results from an orthogonal transformation will remain the same as in pixel domain. We carried out the recognition process on the different datasets but this time with wavelet coefficients as input features. The results from KNN, SC and our proposed NSC were exactly the same. The results from SVM were considerably worse. Due to limitations in space, we do not show the results from wavelets since they will be largely repetitive of the values in Tables 4-1 and 4-2.

In Tables 4-3 and 4-4, we tabulate the results of RIP projection. The projection matrix was created by normalizing the columns of a matrix whose columns comes from i.i.d Normal distribution. The original data is a feature vector of length 256. We make RIP projections to 120 dimensions. Since a single RP is not stable, we follow the methodology proposed in [1, 2].

Table 4-3: Recognition Error for Pessimistic Scenario after RP

Set No.	KNN	Proposed	SC	SVM
1	0.1582	**0.0928**	0.1098	0.1895
2	0.3516	**0.2667**	0.3033	0.3908
3	0.4183	**0.2941**	0.3255	0.4300
4	0.4275	**0.3477**	0.3673	0.4405
5	0.2863	**0.2222**	0.2392	0.3000

Table 4-4: Recognition Error for Optimistic Scenario after RP

Set No.	KNN	Proposed	SC	SVM
1	0.0253	**0.0074**	0.0114	0.0278
2	0.0368	**0.0082**	0.0172	0.0408
3	0.0596	**0.0278**	0.0392	0.0621

Tables 4.3 and 4.4 empirically corroborate our results from Section 4.3. The results after random projection are approximately the same as before projection. The superiority of our classifier is maintained. Due to limitations in space we have only proved the approximate invariance of NSC for RIP projections. It is possible to show that both NN and SC also enjoy the same properties, however SVM is not robust to such projections.

4. 4 Conclusion

In this paper we propose a Nearest Subspace Classifier. The classifier is based on a simple assumption and is quite easy to implement. Our proposed classifier works well in the 'small sample' scenario (training samples in each class being less than the number of input features). The NSC is invariant to orthogonal (or tight frame) projection of the inputs and is also approximately invariant to random projections following the Restricted Isometric Property.

We applied this classifier for the problem of Bengali isolated basic character recognition. Our classifier outperforms both traditional (NN and SVM) and state-of-the-art classifiers (SC) by a significant margin.

4. 5 References

[1] Y. Yang, J. Wright, Y. Ma and S. S. Sastry, “Feature Selection in Face Recognition: A Sparse Representation Perspective”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 31 (2), pp. 210-227, 2009.

[2] IEEE Signal Processing Magazine, Vol. 25(2), March, 2008.

[3] N. Goel, G. Bebis and A. V. Nefian, “Face recognition experiments with random projections”, SPIE Conference on Biometric Technology for Human Identification, 426-437, 2005.

[4] A. Majumdar, “Bangla Basic Character Recognition using Digital Curvelet Transform”, Journal of Pattern Recognition Research, Vol. 2, (1), 2007.

[5] http://www.omicronlab.com/bangla-fonts.html

[6] A. Majumdar and R. K. Ward, “Classification via Group Sparsity Promoting Regularization”, IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 873-876, 2009.

CHAPTER 5: COMPRESSIVE CLASSIFICATION[4]

5. 1 Introduction

The term ‘Compressive Classification’ (CC) was first coined in [1]. It originated with a new paradigm in signal processing called ‘Compressive Sampling’ or ‘Compressed Sensing’ (CS) [2, 3]. CS combines dimensionality reduction with data acquisition by collecting a (random) lower dimensional projection of the original data instead of sampling it. Compressive Classification (CC) refers to a new class of classification methods that are robust to data acquired using CS. Only a few properties are preserved by CS data acquisition, and Compressive Classifiers are designed to exploit these properties so that the recognition accuracy on data acquired by CS is approximately the same as that on data acquired by traditional sampling.

There is a basic difference that separates CC from conventional classification methods. In conventional classification, the data are acquired by traditional (Nyquist) sampling. Once all the data are obtained, a data-dependent dimensionality reduction technique is employed; the data acquisition and dimensionality reduction are disjoint activities. CC operates on data acquired by CS technique where the dimensionality reduction occurs simultaneously with data acquisition. Thus CC works with a dimensionality reduction method that is data-independent, whereas the dimensionality reduction techniques in traditional classification are data-dependent (e.g., Principal Component Analysis, Linear Discriminant Analysis, etc.).

For some practical situations, data-dependent dimensionality reduction methods are not efficient. Consider a practical scenario of face authentication in a bank or an office. In a bank, new clients are added daily to the database, and in some offices, employees are also added on a regular basis. Suppose at a certain given time, face images of 200 people are available, and following conventional face recognition methods (e.g., Eigenface, Fisherface) a data-dependent dimensionality reduction is employed, resulting in a high-to-low dimensional projection matrix.

[4] A version of this chapter has been submitted for publication. A. Majumdar and R. K. Ward, “Compressive Classification”.

When images of 10 more people are added (e.g., the next day), the projection matrix from the high-to-low dimension must be recalculated for all 210 people. Unfortunately, there is no way for the old projection matrix to be updated by the new data. For such cases, a data-independent dimensionality reduction method is desirable. Such a scenario can be easily handled by CC. CC uses a random projection matrix for dimensionality reduction. The projection matrix is data independent (it can be a Gaussian or Bernoulli type random matrix or a partial Fourier matrix). Compressive Classifiers are data-independent in the sense that they do not require re-training (like Support Vector Machines or Artificial Neural Networks) whenever new data are added.

Dimensionality reduction by random projection (i.e. CS data acquisition) gives good results only if the classifier is based on a distance-based measure (e.g. Euclidean or cosine). Consequently, the Nearest Neighbour (NN) classifier is robust to such randomly projected data and can be used as a Compressive Classifier. Other studies have shown empirically that random projection can also be used in conjunction with certain Artificial Neural Networks (ANN) [4] and Support Vector Machines (SVM) [5]. However, both ANN and SVM have a data dependent training phase, i.e., they need to be retrained whenever new data are added. As a result, ANN or SVM are not computationally efficient solutions to the aforementioned problem. Hence, we will not consider these classifiers in this work. The idea behind Compressive Classification is to provide data-independent solutions for dimensionality reduction and classification problems.

Traditionally it is assumed that the training phase is offline, so constraint on the time/computation during training is weak. In this case, the existing sophisticated methods related to dimensionality reduction [6-12] and classification [13, 14] can be employed. It should be mentioned that some effort in online-training is discernible in current face recognition research. Traditionally it was assumed that the training phase is offline, and the training samples are fixed, i.e. do not change with time. However practical scenarios dictate updating of the recognition system on a regular basis. Some studies aim at modifying the existing dimensionality reduction [15, 16] and classification methods [17] to accommodate incremental training. In this paper we do not try to modify the established machine learning techniques and make them amenable for online training. We address the problem from a new perspective altogether.

Compressive Sampling is a new area of signal processing and Compressive Classification is even newer. There are only a handful of studies that classify data acquired by CS techniques and thus fall under the CC framework. The first work on Compressive Classification aims at target recognition, i.e., identifying objects from translated and/or rotated versions [1]. It applies the Generalized Maximum Likelihood Ratio Test (GMLRT) to compressed samples for the purpose of target recognition. This approach shows good results on some toy examples. Another work that collects data by CS techniques but does not really fall under the purview of CC is [18]. In [18], a CS type data acquisition is employed, but instead of working on the compressed samples, the original high dimensional data is reconstructed before classification, therefore the computational advantage of working on low dimensional data is lost. This work also uses synthesized toy examples for experimental verification. A third work addresses face recognition [19]. It is based on the assumption that the test sample can be approximately represented as a linear combination of the test samples belonging to the correct class. This study shows that robust recognition results can be obtained by taking random lower dimensional projection (CS type projection) of face images followed by a new compressive classification algorithm (the details of this work will be discussed in Section 5.3). This was the first work to propose a CC algorithm (Sparse Classifier) that parallels traditional classifiers like Artificial Neural Networks, Support Vector Machines or Nearest Neighbor. In [19], it was shown that the new compressive classifier yields better results than NN or SVM on real world face recognition datasets. The other works in this field are by the authors of the current paper [20-23]. The work in [20] is a generalization of the sparse classifier (SC) developed in [19], with the original assumption extended to include cases where test samples can be represented by a non-linear combination of test samples. The work in [21] has the same assumption as [19], but proposes a more refined group sparse optimization (instead of ordinary sparse optimization [19]) technique for increasing classification accuracy; this method is called Group Sparse Classifier (GSC). Recently, we have proposed a faster version of the GSC based on approximate greedy algorithms rather than group sparse optimization; these are called the Fast Group Sparse Classifiers (FGSC) [22]. In a separate but related work [23] we proposed the Nearest Subspace Classifier (NSC), which is based on the assumption that training samples for a particular class span a sub-space for representing any new test sample belonging to that class. The NSC is not based on any optimization approach and is operationally faster than the rest. The previous studies [19-23] were proposed as independent classifiers. In this paper, we will (in Section 5.4) prove that why all these methods fall under the category of Compressive Classifiers.

The Fast Group Sparse Classifier [22] increases the operation speed of the Group Sparse Classifier [21], by replacing an optimization problem by a greedy approximate algorithm. In a similar manner, it is possible to speed up the Sparse Classifier [19] by replacing its sparse optimization step by greedy approximate algorithms. We propose this variation of the SC in this paper and call it the Fast Sparse Classifier (FSC). In our previous work on FGSC [22], the greedy group algorithms were based on Orthogonal Matching Pursuit; in this paper we will show that similar algorithms can be based on Orthogonal Least Squares as well.

The aim of Compressed Sensing is signal reconstruction from compressed samples (random projections of the original signal). Compressive Classification, on the other hand, aims at directly classifying such compressed samples, i.e., without the need to reconstruct the original signals. The following section provides a very brief introduction to the basic tenets of CS. Section 5.3 discusses the recently proposed Compressive Classification algorithms. As mentioned above, of the traditional classification methods, Nearest Neighbour can be employed as a CC. In Section 5.4, we prove that NN and the other algorithms perform Compressive Classification. The experimental results are shown in Section 5.5. The first set of results obtained using general purpose classification databases shows that the proposed classification algorithms are good for these tasks. The second set of results experimentally validates the robustness of these classifiers to CS data acquisition. Finally, in Section 5.6 we conclude the work.

5. 2 Compressed Sensing: A Brief Overview

Compressed Sensing, or Compressive Sampling (CS), is a new paradigm in signal processing. It studies the reconstruction of a sparse signal from its under-sampled projections (compressed samples). There are many natural signals where CS is a desirable alternative to conventional

Nyquist sampling, as the signals can be made sparse by representations in another domain such as DCT or Wavelets.

CS data acquisition is very simple and involves obtaining under-sampled projections of the signal. The challenge is in the signal reconstruction. Let *x* be the underlying sparse signal, which has been under-sampled by a projection matrix *A* to collect the compressive sample *y*:

$$y = Ax, \text{ where A is m} \times \text{n (n>m), y is m} \times \text{1 and x is n} \times \text{1} \qquad (1)$$

During reconstruction, the problem is to solve an under-determined system (1); the system is under-determined because the projection matrix A has more columns than rows. This system of equations in general does not have a unique solution. But if the solution is known to be sparse, studies in CS literature say that the sparse solution is unique as well, i.e., an under-determined system of equations does not have more than one sparse solution. If it is assumed that the signal *x* is *k* sparse, then there are *2k* unknowns (k-positions and k-non zero values). With enough computing resources, it would be possible to solve this problem by an exhaustive search of all possible combinations. This would require *m*=*O*(*k*) observations for solving (1).

Even though in principle it is possible to solve (1) from O(*k*) equations, (1) is known to be an NP hard problem requiring an exhaustive search of all the possible combinations. Mathematically, this problem is expressed as

$$\begin{array}{l} \min ||x||_0 \text{ subject to } y = Ax \\ ||x||_0 = \text{ number of non-zeroes in x} \end{array} \qquad (2)$$

Solving the l_0 *minimization* problem although possible theoretically, is impractical since it requires an exhaustive search. CS literature indicates that, when *x* is sparse and the matrix *A* follows certain properties, it is possible to solve (1) via a convex optimization (l_1 minimization) problem of the form

$$\min ||x||_1 \text{ subject to } y = Ax \qquad (3)$$

This is a tractable solution which can be easily solved by linear programming. But the number of observations needed to solve (1) via an optimization of the form (3) requires about *m*=*O*(*k* log*n*) and is larger than that required by (2). This theoretical estimate is pessimistic; practically it has been found that (1) can be solved via l_1 minimization with fewer observations. There are other greedy (suboptimal) solutions to the sparse inversion problem. They are faster, but the solution requires more samples (equations).

The condition on matrix A that guarantees a unique solution of (1) via l_1 minimization is called the Restricted Isometric Property (RIP). This is written as

$$(1-\delta)\,||\,x\,||_2 \le ||\,Ax\,||_2 \le (1+\delta)\,||\,x\,||_2, \text{ where } \delta \text{ is a small constant} \qquad 4)$$

In general, it is not possible to find deterministic matrices following RIP. But matrices with i.i.d Gaussian columns, Bernoulli matrices and partial Fourier matrices have been proven to follow RIP with a very high probability.

Many naturally occurring signals are not sparse in the physical domain but can be made sparse by some linear transform, e.g. images are sparse in the Discrete Cosine Transform or Wavelet domain, Distributed Sensor Network (DSN) data are sparse in network wavelets and Orthogonal Frequency Division Multiplexing (OFDM) channels are sparse in the time domain. Recent studies have applied CS techniques to such signal estimation problems.

Let *t* be a naturally (not sparse) occurring signal. Orthogonal transforms (Φ) can sparsify *t* to obtain sparse coefficients (α) i.e.

$$\begin{aligned} &\alpha = \Phi^T t - \text{ analysis equation} \\ &t = \Phi\alpha - \text{ synthesis equation} \end{aligned} \qquad (5)$$

Let us assume that a Gaussian RIP matrix *A* is used for sampling the signal *t* in physical domain, i.e.

$$s = At = A\Phi\alpha \qquad (6)$$

where *s* is the vector of collected samples (compressed samples) and *A* is a RIP matrix.

As Φ is an orthogonal transform, the matrix AΦ (formed by a linear combination of i.i.d Gaussian columns) also satisfies RIP. This implies that it is possible to sample the naturally occurring signal *t* via a RIP matrix to obtain the observation vector *s*, and then solve for the sparse coefficients in *α*:

$$\min \|\alpha\|_1 \text{ subject to } s = A\Phi\alpha$$

Once α is obtained, the signal t can be reconstructed by the synthesis equation $t = \Phi\alpha$.

The fact that RIP holds for linearly sparsifiable signals (i.e. signals that are not sparse in the original domain but are sparse in an orthogonal domain) makes CS a practical tool for signal processing.

5. 3 Classification Algorithms

The classification problem involves finding the identity of an unknown test sample given a set of training samples and their class labels. Compressive Classification addresses the case where compressive samples (random projections) of the original signals are available instead of the original signal. Compressive Classifiers have two challenges to meet:

1. The classification accuracy of CC on the original signals should be at par with classification accuracy from traditional classifiers (SVM or ANN or KNN).
2. The classification accuracy from CC should not degrade much when compressed samples are used instead of the original signals.

Recently some classifiers have been proposed that can be employed as compressive classifiers. We discuss those classification algorithms in this section.

5.3.1 The Sparse Classifier

The Sparse Classifier (SC) is proposed in [19]. It is based on the assumption that the training samples of a particular class approximately form a linear basis for a new test sample belonging to the same class. If $v_{k,test}$ is the test sample belonging to the k^th class, then

$$v_{k,test} = \alpha_{k,1}v_{k,1} + \alpha_{k,2}v_{k,2} + \ldots + \alpha_{k,n_k}v_{k,n_k} + \varepsilon_k = \sum_{i=1}^{n_k}\alpha_{k,i}v_{k,i} + \varepsilon_k \qquad 7)$$

where $v_{k,i}$ *'s* are the training samples of the kth class and ε is the approximation error (assumed to be Normally distributed).

Equation (7) expresses the assumption in terms of the training samples of a single class. Alternatively, it can be expressed in terms of all the training samples such that

$$\begin{aligned} v_{k,test} &= \alpha_{1,1} + \ldots \alpha_{k,1} v_{k,1} + \ldots + \alpha_{k,n_k} v_{k,n_k} + \ldots + \alpha_{C,n_C} v_{C,n_C} + \varepsilon \\ &= \sum_{i=1}^{n_1} \alpha_{1,i} v_{1,i} + \ldots \sum_{i=k}^{n_k} \alpha_{k,i} v_{k,i} + \ldots + \sum_{i=1}^{n_C} \alpha_{C,i} v_{C,i} + \varepsilon \end{aligned} \qquad (8)$$

where C is the total number of classes.

In matrix vector notation, equation (8) can be expressed as

$$v_{k,test} = V\alpha + \varepsilon \qquad (9)$$

where $V = [v_{1,1} \,|\, \ldots \,|\, v_{k,1} \,|\, \ldots \,|\, v_{k,n_k} \,|\, \ldots \,|\, v_{C,n_C}]$ and $\alpha = [\alpha_{1,1} \ldots \alpha_{k,1} \ldots \alpha_{k,n_k} \ldots \alpha_{C,n_C}]'$.

The linearity assumption in [19] coupled with formulation (9) implies that the coefficients vector α should be non-zero only when it corresponds to the correct class of the test sample.

Based on this assumption, the following sparse optimization problem was proposed in [19]

$$\min \|\alpha\|_0 \text{ subject to } \|v_{k,test} - V\alpha\|_2 \leq \eta,\ \eta \text{ is related to } \varepsilon \qquad (10)$$

As previously mentioned, (10) is an NP hard problem. Consequently in [19], a convex relaxation to the NP hard problem was made and the following problem was solved instead

$$\min \|\alpha\|_1 \text{ subject to } \|v_{k,test} - V\alpha\|_2 \leq \eta \qquad (11)$$

The formulation of the sparse optimization problem as in (11) is not ideal for this scenario, as it does not impose sparsity on the entire class as the assumption implies. The proponents of the Sparse Classifier [19] 'hope' that the *l_1*-norm minimization will find the correct solution even though it is not explicitly imposed in the optimization problem. We will speak more about group sparse classification in the subsection 5.3.3.

The sparse classification (SC) algorithm proposed in [19] is as follows:

Sparse Classifier Algorithm

1. Solve the optimization problem expressed in (11).

2. For each class (i) repeat the following two steps:

   2.1 Reconstruct a sample for each class by a linear combination of the training samples belonging to that class using $v_{recon}(i) = \sum_{j=1}^{n_i} \alpha_{i,j} v_{i,j}$.

   2.2 Find the error between the reconstructed sample and the given test sample by $error(v_{test}, i) = \| v_{k,test} - v_{recon(i)} \|_2$.

3. Once the error for every class is obtained, choose the class having the minimum error as the class of the given test sample.

5.3.2 Fast Sparse Classifiers

The above sparse classification (SC) algorithm yields good classification results, but it is slow. This is because of convex optimization (*l_1* minimization). It is possible to create faster versions of the SC by replacing the optimization step (step 1 of the above algorithm) by a fast greedy (suboptimal) alternative that approximates the original *l_0* minimization problem (10). Such greedy algorithms serve as a fast alternative to convex-optimization for sparse signal estimation problems. In this work, we apply these algorithms in a new perspective (classification).

We will discuss two basic greedy algorithms that were employed to speed-up the SC in [19]: the Orthogonal Matching Pursuit (OMP) and the Orthogonal Least Squares (OLS) [24]. We repeat these algorithms here for the sake of completeness. Both of these algorithms approximate the NP hard problem, $\min \| x \|_0$ subject to $\| y - Ax \|_2 \leq \eta$.

<u>OMP Algorithm</u>

Inputs: measurement vector *y* (mX1), measurement matrix *A* (mXn) and error tolerance η.

Output: estimated sparse signal *x*.

Initialize: residual $r_0 = y$, the index set $\Lambda_0 = \varnothing$, the matrix of chosen atoms $\Phi_0 = \varnothing$, and the iteration counter t = 1.

1. At the iteration = t, find $\lambda_t = \underset{j=1\ldots n}{\arg\max} |< r_{t-1}, \varphi_j >|$
2. Augment the index set $\Lambda_t = \Lambda_{t-1} \cup \lambda_t$ and the matrix of chosen atoms $\Phi_t = [\Phi_{t-1} \; A_{\lambda_t}]$.
3. Get the new signal estimate $\min_x || x_t - \Phi_t y ||_2^2$.
4. Calculate the new approximation and the residual $a_t = \Phi_t x_t$ and $r = y - a_t$.

Increment t and return to step 1 if $|| r || \geq \varepsilon$.

<u>OLS Algorithm</u>

Inputs: measurement vector y (mX1), measurement matrix A (mXn) and error tolerance η.

Output: estimated sparse signal x.

Initialize: residual $r_0 = y$, the index set $\Lambda_0 = \varnothing$, the matrix of chosen atoms $\Phi_0 = \varnothing$, and the iteration counter t = 1.

1. At the iteration = t, find $\lambda_t = \underset{j=1\ldots n \notin \Lambda_{t-1}}{\arg\min} || x - \Phi_{\Lambda_t^j} \Phi^{\dagger}_{\Lambda_t^j} ||_2$.
2. Augment the index set $\Lambda_t = \Lambda_{t-1} \cup \lambda_t$ and the matrix of chosen atoms $\Phi_t = [\Phi_{t-1} \; A_{\lambda_t}]$.
3. Get the new signal estimate $\min_x || x_t - \Phi_t y ||_2^2$.
4. Calculate the new approximation and the residual $a_t = \Phi_t x_t$ and $r = y - a_t$.

Increment t and return to step 1 if $|| r || \geq \varepsilon$.

The inputs, output and the initialization process of the OLS and the OMP algorithms are the same. The problem is to estimate the sparse signal. The residual is initialized to the measurement vector. The index set and the matrix of chosen atoms (columns from the measurement matrix) are empty. In both algorithms, the first step of each iteration is to select a non-zero index of the sparse signal. The OLS and the OMP algorithms differ from each other only in the selection strategy (step 2). In OMP, the current residual is correlated with the measurement matrix and the index of the highest correlation is selected. In OLS, the index is selected such that it leads to the minimum residual error after orthogonalization. In the third step of the iteration, the selected index is added to the current index set and the set of selected atoms (columns from the measurement matrix) is also updated from the current index set. In the fourth step, the estimates of the signal at the given indices are obtained via least squares. In step 4, the residual is updated. Once all the steps are performed for the iteration, a check is done to see if the norm of the residual falls below the error estimate. If it does, the algorithm terminates; otherwise it repeats steps 2 to 4.

The FSC algorithm differs from the SC algorithm only in step 1. Instead of solving the *l_1 minimization* problem, FSC uses either OMP or OLS for a greedy approximation of the original *l_0 minimization* problem. There may be variants of the FSC algorithm depending on the variations of the basic OMP or OLS algorithms.

5.3.3 Group Sparse Classifier

As mentioned in subsection 5.3.1, the optimization algorithm formulated in [19] does not exactly address the desired objective. A sparse optimization problem was formulated in the hope of selecting training samples of a particular (correct) class. It has been shown in [25] that *l_1 minimization* cannot select a sparse group of correlated samples (in the limiting case it selects only a single sample from all the correlated samples). In classification problems, the training samples from each class are highly correlated, therefore *l_1 minimization* is not an ideal choice for ensuring selection of all the training samples from a group. To overcome this problem of [19] the Group Sparse Classifier was proposed in [21]. It has the same basic assumption as [19] but the optimization criterion is formulated so that it promotes selection of the entire class of training samples.

The basic assumption of expressing the test sample as a linear combination of training samples is formulated in (9) as $v_{k,test} = V\alpha + \varepsilon$

where $V = [v_{1,1} | \dots | v_{1,n_1} | \dots | v_{k,1} | \dots | v_{k,n_k} | \dots v_{C,1} | \dots | v_{C,n_C}]$ and

$$\alpha = [\underbrace{\alpha_{1,1},\dots,\alpha_{1,n_1}}_{\alpha_1},\underbrace{\alpha_{2,1},\dots,\alpha_{2,n_2}}_{\alpha_2},\dots\underbrace{\alpha_{C,1},\dots,\alpha_{C,n_C}}_{\alpha_C}]^T .$$

The above formulation demands that ***α*** should be 'group sparse', meaning that the solution of the inverse problem (2) should have non-zero coefficients corresponding to a particular group of training samples, and zero coefficients elsewhere (i.e. $\alpha_i \neq 0$ for only one of the α_i's, i=1,…,C). This requires the solution of

$$\min_{\alpha} || \alpha ||_{2,0} \text{ such that } ||\text{v}_{test} - V\alpha ||_2 < \varepsilon \quad (12)$$

The mixed norm $||\bullet||_{2,0}$ is defined for $\alpha = [\underbrace{\alpha_{1,1},\dots,\alpha_{1,n_1}}_{\alpha_1},\underbrace{\alpha_{2,1},\dots,\alpha_{2,n_2}}_{\alpha_2},\dots\underbrace{\alpha_{k,1},\dots,\alpha_{k,n_k}}_{\alpha_k}]^T$

as $|| \alpha ||_{2,0} = \sum_{l=1}^{k} I(|| \alpha_l ||_2 > 0)$, where $I(|| \alpha_l ||_2 > 0) = 1$ if $|| \alpha_l ||_2 > 0$.

Solving the *l*$l_{2,0}$ *minimization* problem is NP hard. We proposed a convex relaxation in [23], so that the optimization takes the form

$$\min_{\alpha} || \alpha ||_{2,1} \text{ such that } ||\text{v}_{test} - V\alpha ||_2 < \varepsilon \quad (13)$$

where $|| \alpha ||_{2,1} = || \alpha_1 ||_2 + || \alpha_2 ||_2 + \dots + || \alpha_k ||_2$.

Solving the $l_{2,1}$ *minimization* problem is at the core of the GSC. The classification algorithm is as follows:

Group Sparse Algorithm

1. Solve the optimization problem expressed in (13).
2. Find those i's for which $||\alpha_i||_2 > 0$.
3. For those classes (i) satisfying the condition in step 2, repeat the following two steps:

3.1 Reconstruct a sample for each class by a linear combination of the training samples in that class via the equation $v_{recon}(i)=\sum_{j=1}^{n_i}\alpha_{i,j}v_{i,j}$.

3.2 Find the error between the reconstructed sample and the given test sample by $error(v_{test},i)=\|v_{k,test}-v_{recon(i)}\|_2$.

4. Once the error for every class is obtained, choose the class having the minimum error as the class of the given test sample.

5.3.4 Fast Group Sparse Classification

The Group Sparse Classifier [21] gives better results than the Sparse Classifier [19] but is slower. In a very recent work [23], we proposed alternate greedy algorithms for group sparse classification and were able to increase the operating speed by two orders of magnitude. These classifiers were named Fast Group Sparse Classifiers (FGSC).

FSC is built upon greedy approximation algorithms of the NP hard sparse optimization problem (10). Such greedy algorithms are a well studied topic in signal processing. It was therefore straightforward to apply some known greedy algorithms (OMP and OLS) to the sparse classification problem. Group sparsity-promoting optimization, however is not a vastly researched topic, unlike sparse optimization. Since previous work in group sparsity solely rely on convex optimization [26-31]. We had to develop a number of greedy algorithms as (fast and accurate) alternatives to convex group sparse optimization [21].

Twelve greedy algorithms were proposed in [21], based on the Orthogonal Matching Pursuit (OMP) algorithm [32]. Alternatively, similar algorithms can be based on the Orthogonal Least Squares (OLS) [24] as well. In this work, we propose two new group-sparse optimization algorithms based on OLS. It is not possible to discuss each of the OMP and the OLS based algorithms in detail. Instead, we describe the basic framework for these algorithms and discuss two specific algorithms (one from OMP and the other from OLS) in detail.

All greedy group sparse algorithms approximate the problem $\min \| x \|_{2,0}$ subject to $\| y - Ax \|_2 \leq \eta$. They work in a very intuitive way – first they try to identify the group which has non-zero coefficients. Once the group is identified, the coefficients for the group indices are estimated by some simple means.

Framework for Greedy Group Sparse Approximation Algorithm

Inputs: measurement vector y mX1, measurement matrix A (mXn), group labels and the error tolerance η.

Output: estimated sparse signal x.

Initialization: As the estimated signal (vector) is sparse, therefore it is initialized to all zeros. At each iteration, one group of indices is selected and updated to have non-zero coefficients. The residual is the measure of misfit between the sparse signal and its estimate. But since the sparse signal is not available, the algorithm calculates the residual between the projection of the current estimate of the signal and the measurement vector, i.e. $r = y - A\hat{x}_{current}$. The residual is initialized to the current measurement vector $r_0 = y$ .The index set Λ contains the indices of all the selected groups that have been chosen so far, i.e., up to the current iteration. Initially the set is empty, $\Lambda_0 = \varnothing$. The matrix Φ contains the group of selected columns (atoms) from A that are indexed in Λ; initially the matrix of selected atoms $\Phi_0 = \varnothing$.

Iteration Steps –

1. The first step in any iteration *t* is the selection criterion for choosing the sparse group. Two selection strategies can be employed here – OMP based or OLS based. In the OMP based methods, the correlations between the current residual and the measurement matrix A are stored in λ_t. In the OLS based methods the mismatch between the vector y and the selected columns of A are stored as λ_t.

2. The second step chooses the groups. There can be two strategies for selecting the group – Block selection and Group selection. Block selection for OMP was proposed in [31]. We

proposed the group selection criterion in [23]. In block selection, the entire group is selected based on the highest correlation or lowest mismatch (between the current residual with the measurement matrix) of a single index, whereas in group selection the group is selected based on the average correlation or mismatch of the group. Whichever selection strategy is employed, at the end of step 2, the class of selected samples is stored as $class(\lambda_t)$.

3. The third step updates the index set and the matrix of selected atoms. The index set is updated as $\Lambda_t = \Lambda_{t-1} \cup class(\lambda_t)$ and the matrix of chosen atoms is then $\Phi_t = [\Phi_{t-1}\ A_{class(\lambda_t)}]$.

4. The fourth step calculates the current estimate of the signal. It is most common to use least squared estimation for the current signal $\hat{x}$ yielded by $\min_x \| y - \Phi_t x_t \|_2^2$.

5. Finally the residual is updated as $r = y - \Phi_t x_t$.

The stopping criterion: after each iteration, the algorithms checks whether a stopping criterion is met. The stopping criterion may be a bound on the norm of the residual or on the sparsity of the estimated signal.

It is not possible to discuss all the proposed greedy algorithms here. In this paper, we will use the Block/Group OMP and the Block/Group OLS algorithms. We discuss one algorithm of each type. The Block OMP was proposed in a different context in [33], therefore we do not discuss it here. We describe our Group OMP (GOMP) algorithm (for other OMP based algorithms please refer to [23]). The GOMP algorithm explains the Group selection strategy using the OMP based method. Next we discuss the Block OLS since it explains both the OLS selection strategy at the same time.

GOMP Algorithm

Inputs: measurement vector y (mX1), measurement matrix A (mXn), group labels and the error tolerance η.

Output: estimated sparse signal x.

Initialize: residual $r_0 = y$, index set $\Lambda_0 = \varnothing$, matrix of chosen atoms $\Phi_0 = \varnothing$, and iteration counter t = 1.

1. At iteration t, compute $\lambda(j) = |< r_{t-1}, \varphi_j >|, \forall j = 1...n$

2. Group selection – select the class with the maximum average correlation $\tau_t = \underset{i=1...C}{\arg\max}(\frac{1}{n_i}\sum_{j=1}^{n_i}\lambda(j))$, denote it by $class(\tau_t)$.

3. Augment the index set $\Lambda_t = \Lambda_{t-1} \cup class(\tau_t)$ and the matrix of the chosen atoms $\Phi_t = [\Phi_{t-1}\ A_{class(\tau_t)}]$.

4. Get the new signal estimate using $\min_x \| y - \Phi_t x_t \|_2^2$.

5. Calculate the new approximation and the residual $a_t = \Phi_t x_t$ and $r = y - a_t$.

Increment t and return to step 1 if $\| r_t \| \geq \varepsilon$.

As mentioned earlier, one algorithm will be discussed from each of the OMP and the OLS based methods. Since we have already described the GOMP, we discuss the Block OLS (BOLS) next since it will explain both the OLS selection criterion and the block strategy.

<u>BOLS Algorithm</u>

Inputs: measurement vector y (mX1), measurement matrix A (mXn), group labels and the error tolerance η.

Output: estimated sparse signal x.

Initialize: residual $r_0 = y$, index set $\Lambda_0 = \varnothing$, matrix of chosen atoms $\Phi_0 = \varnothing$, and iteration counter t = 1.

1. At iteration t, find $\lambda_t = \underset{j=1...n \notin \Lambda_{t-1}}{\arg\min} \| x - \Phi_{\Lambda_t^j} \Phi_{\Lambda_t^j}^{\dagger} \|_2$.

2. Block selection: select the class which includes the minimum residual, call it $class(\lambda_t)$.

3. Augment the index set $\Lambda_t = \Lambda_{t-1} \cup class(\lambda_t)$ and the matrix of chosen atoms $\Phi_t = [\Phi_{t-1} \; A_{class(\lambda_t)}]$.

4. Get the new signal estimate using $\min_x \| y - \Phi_t x_t \|_2^2$.

5. Calculate the new approximation and the residual $a_t = \Phi_t x_t$ and $r_t = y - a_t$.

Increment t and return to step 1 if $\| r_t \| \geq \varepsilon$.

5.3.5 Nearest Subspace Classifier

The Nearest Subspace Classifier (NSC) [20] makes a novel classification assumption – samples from each class lie on a hyper-plane specific to that class. According to this assumption, the training samples of a particular class span a subspace. When a sample becomes available, the problem is to determine to which subspace it belongs. Any new test sample belonging to that class can thus be represented as a linear combination of the test samples, i.e.,

$$v_{k,test} = \sum_{i=1}^{n_k} \alpha_{k,i} \cdot v_{k,i} + \varepsilon_k \qquad (14)$$

where $v_{k,test}$ is the test sample (i.e., the vector of features) assumed to belong to the kth class, $v_{k,i}$ is the ith training sample of the kth class, and ε_k is the approximation error for the kth class.

Owing to the error term in equation (14), the relation holds for all the classes *k=1…C*. In such a situation, it is reasonable to assume that for the correct class the test sample has the minimum error ε_k.

To find the class that has the minimum error in equation (14), the coefficients $\alpha_{k,i}$ *k=1…C* must be estimated first. This can be performed by rewriting (14) in matrix-vector notation

$$v_{k,test} = V_k \alpha_k + \varepsilon_k \qquad (15)$$

where $V_k = [v_{k,1} \,|\, v_{k,2} \,|\, ... \,|\, v_{k,n_k}]$ and $\alpha_k = [\alpha_{k,1}, \alpha_{k,2} ... \alpha_{k,n_k}]^T$.

The solution to (15) can be obtained by minimizing

$$\hat{\alpha}_k = \underset{\alpha}{\arg\min} \| v_{k,test} - V_k \alpha \|_2^2 \tag{16}$$

The previous work on NSC [20] directly solves (16). However, the matrix V_k may be under-determined, i.e., the number of samples may be greater than the dimensionality of the inputs. In such a case, instead of solving (16), Tikhonov regularization is employed, so that the following is minimized:

$$\hat{\alpha}_k = \underset{\alpha}{\arg\min} \| v_{k,test} - V_k \alpha \|_2^2 + \lambda \| \alpha \|_2^2 \tag{17}$$

The analytical solution of (17) is

$$\hat{\alpha}_k = (V_k^T V_k + \lambda I)^{-1} V_k^T v_{k,test} \tag{18}$$

Plugging this expression into (15) and solving for the error term, we get

$$\varepsilon_k = (V_k (V_k^T V_k + \lambda I)^{-1} V_k^T - I) v_{k,test} \tag{19}$$

Based on equations (15-19), the Nearest Subspace Classifier algorithm has the following steps.

<u>Training:</u>

1. For each class ‘k’, by computing the orthoprojector (the term in brackets in equation (19)).

<u>Testing:</u>

2. Calculate the error for each class ‘k’ by computing the matrix vector product between the orthoprojector and $v_{k,test}$.
3. Classify the test sample as the class having the minimum error ($\| \varepsilon_k \|$).

5. 4 Classification Robustness to Data Acquired by CS

The idea of using random projection for dimensionality reduction of face images was proposed in [19]. It was experimentally shown that the Sparse Classifier is robust to such dimensionality reduction. However the theoretical understanding of why the SC is robust to such dimensionality reduction was lacking in [19]. In this section, we will prove why the SC and all the other classifiers discussed in the previous section can be categorized as Compressive Classifiers. The two conditions that guarantee the robustness of CC under random projection are the following:

Restricted Isometric Property (RIP) [3] – The l_2-norm of a sparse vector is approximately preserved under a random lower dimensional projection, i.e., when a sparse vector *x* is projected by a random projection matrix *A*, then $(1-\delta)\|x\|_2 \leq \|Ax\|_2 \ (1+\delta)\|x\|_2$. The constant *δ* is a RIP constant whose value depends on the type of matrix *A* and the number of rows and columns of *A* and the nature of *x*. An approximate form (without upper and lower bounds) of RIP states $\|Ax\|_2 \approx \|x\|_2$.

Generalized Restricted Isometric Property (GRIP) [34] – For a matrix *A* which satisfies RIP for inputs x_i, the inner product of two vectors $(< w,v >= \|w\|_2 \cdot \|v\|_2 \cos\theta)$ is approximately maintained under the random projection *A*, i.e. for two vectors x_1 and x_2(which satisfies RIP with matrix *A*), the following inequality is satisfied: $(1-\delta)\|x_1\|_2 \cdot \|x_2\|_2 \cos[(1+\sqrt{3}\delta_m)\theta] \leq \langle Ax_1, Ax_2 \rangle \leq (1+\delta)\|x_1\|_2 \cdot \|x_2\|_2 \cos[(1-\sqrt{3}\delta_m)\theta]$.

The constants *δ* and δ_m depend on the dimensionality and the type of matrix A and also on the nature of the vectors. Even though the expression seems overwhelming, it can be stated simply as: the angle between two sparse vectors (θ) is approximately preserved under random projections. An approximate form of GRIP is $\langle Ax_1, Ax_2 \rangle \approx \langle x_1, x_2 \rangle$.

RIP and the GRIP were originally proven for sparse vectors. For sparse vectors, the random dimensionality reduction matrix (*A*) is formed by normalizing the columns of an i.i.d Gaussian matrix with zero mean and unit covariance. Such matrices also follow RIP and GRIP for linearly sparsifiable vectors, i.e. vectors that are not sparse but can be sparsified by an orthogonal transform (such as Wavelet or DCT for images or by STFT for speech). Therefore if a vector *y* is not sparse but can be sparsifibale by a linear orthogonal transform such that $y = \Phi x$, where *x* is sparse, RIP and GRIP holds for *y*. This fact (RIP holds for sparsifiable signals) is the main cornerstone of all compressed sensing imaging applications [26].

5.4.1 The Nearest Neighbor Classifier

The Nearest Neighbor (NN) is a compressive classifier. The criterion for NN classification depends on the magnitude of the distance between the test sample and each training sample. There are two popular distance measures:

Euclidean distance ($\| v_{test} - v_{i,j} \|_2, i = 1...C$ and $j = 1...n_i$)

Cosine distance ($\langle v_{test}, v_{i,j} \rangle, i = 1...C$ and $j = 1...n_i$)

It is easy to show that both these distance measures are approximately preserved under random dimensionality reduction, assuming that the random dimensionality reduction matrix *A* follows RIP with samples *v*. Then following the RIP approximation, the Euclidean distance between samples is approximately preserved, i.e.

$$\| Av_{test} - Av_{i,j} \|_2 = \| A(v_{test} - v_{i,j}) \|_2 \approx \| (v_{test} - v_{i,j}) \|_2$$

The fact that the Cosine distance is approximately preserved follows directly from the GRIP assumption

$$\langle Av_{test}, Av_{i,j} \rangle \approx \langle v_{test}, v_{i,j} \rangle$$

5.4.2 The Sparse Classifier and the Group Sparse Classifier

In this subsection it will be shown why the Sparse Classifier and the Group Sparse Classifier can act as compressive classifiers. At the core of the SC and GSC classifiers are the *l_1 minimization* and the *$l_{2,1}$ minimization* optimization problems, respectively:

$$\begin{aligned} &\text{SC-}\min \|\alpha\|_1 \text{ subject to } \|v_{k,test} - V\alpha\|_2 \le \eta \\ &\text{GSC-}\min \|\alpha\|_{2,1} \text{ subject to } \|v_{k,test} - V\alpha\|_2 \le \eta \end{aligned} \qquad (20)$$

In compressive classification, all the samples are projected from a higher to a lower dimension by a random matrix *A*. Therefore, the optimization is as follows:

$$\begin{array}{l}\text{SC-}\min \|\beta\|_1 \text{ subject to } \|Av_{k,test} - AV\beta\|_2 \leq \eta \\ \text{GSC-}\min \|\beta\|_{2,1} \text{ subject to } \|Av_{k,test} - AV\beta\|_2 \leq \eta\end{array} \qquad (21)$$

The objective function does not change before and after projection, but the constraints do. We will show that the constraints of (21) and (20) are approximately the same; therefore the optimization problems are the same as well. The constraint in (21) is:

$$\begin{array}{l}\|Av_{k,test} - AV\beta\|_2 \leq \eta \\ = \|A(v_{k,test} - V\beta)\|_2 \leq \eta \\ \approx \|(v_{k,test} - V\beta)\|_2 \leq \eta, \text{ following RIP}\end{array}$$

Since the constraints are approximately preserved and the objective function remains the same, the solution to the two optimization problems (20) and (21) will be approximately the same, i.e., $\beta \approx \alpha$.

In the classification algorithm for SC and GSC (and for FSC, FGSC and NSC), the deciding factor behind the class of the test sample is the class-wise error, $error(v_{test}, i) = \| v_{k,test} - \sum_{j=1}^{n_i} \alpha_{i,j} v_{i,j} \|_2, i = 1 \ldots C$. This class-wise error is approximately preserved after random projection.

$$\begin{aligned} error(Av_{test}, i) &= \|Av_{k,test} - A\sum_{j=1}^{n_i} \alpha_{i,j} v_{i,j}\|_2 \\ &= \|A(v_{k,test} - \sum_{j=1}^{n_i} \alpha_{i,j} v_{i,j})\|_2 \\ &\approx \|(v_{k,test} - \sum_{j=1}^{n_i} \alpha_{i,j} v_{i,j})\|_2, \text{ due to RIP}\end{aligned}$$

As the class-wise error is approximately preserved under random projections, the recognition results will also be approximately the same.

5.4.3 Fast Sparse and Fast Group Sparse Classifiers

In the FSC and FGSC classifiers, the following NP hard optimization problem is solved greedily:

$$\text{SC-}\min \|\alpha\|_0 \text{ subject to } \|v_{k,test} - V\alpha\|_2 \le \eta$$
$$\text{GSC-}\min \|\alpha\|_{2,0} \text{ subject to } \|v_{k,test} - V\alpha\|_2 \le \eta$$

Problem (22) pertains to the case of original data. When the samples are randomly projected, the problem has the following form:

$$\text{SC-}\min \|\beta\|_0 \text{ subject to } \|Av_{k,test} - AV\beta\|_2 \le \eta$$
$$\text{GSC-}\min \|\beta\|_{2,0} \text{ subject to } \|Av_{k,test} - AV\beta\|_2 \le \eta$$

We need to show that the results of greedy approximation for the above problems yields $\beta \approx \alpha$.

The OMP and OLS algorithms vary from their block/group versions only in the number of indices chosen at each iteration. The criterion for choosing the indices (step 1 of all algorithms) is the same for both the basic and the block/group versions. Therefore, it is sufficient to show that the selection step in the OMP and OLS algorithms are approximately invariant to random projections. The other important step in these greedy algorithms is the least squares signal estimation step (before the last step in all algorithms), which is the same for both OLS and OMP. We need to show that the least squares estimate is also robust to random projections.

As the least squares signal estimation step is common to both the OLS and the OMP based algorithms, we will first show its robustness to random projections. The least squares estimation is performed as:

$$\min \|y - \Phi x\|_2 \qquad (22)$$

The problem is to estimate the signal *x*, from measurements *y* given the matrix *Φ*.

Both *y* and *Φ* are randomly sub-sampled by a random projection matrix *A* which satisfies RIP. Therefore, the least squares problem in the sub-sampled case takes the form

$$\begin{aligned}&\min \| Ay - A\Phi x \|_2 \\ &= \min \| A(y - \Phi x) \|_2 \\ &\approx \min \| y - \Phi x \|_2, \text{ since RIP holds}\end{aligned}$$

Thus the signal estimate x, obtained by solving the original least squares problem (22), and the randomly sub-sampled problem are approximately the same.

Therefore the signal estimation step for the OMP and OLS based algorithms remains approximately invariant under random projections. Now we will show how the selection step in these algorithms remains invariant to such projections as well. We first show it for OMP based algorithms.

In OMP, the selection is based on the correlation between the measurement matrix *Φ* and the observations *y*, i.e. $\Phi^T y$. If we have $\Phi_{m\times n}$ and $y_{m\times 1}$, then the correlation can be written as inner products between the columns of *Φ* and the vector *y,* i.e., $\langle \phi_i, y \rangle, i = 1...n$. After random projection, both columns of Φ and the measurement y are randomly sub-sampled by a random projection matrix A. The correlation can be calculated as $\langle A\phi_i, Ay \rangle, i = 1...n$, which by GRIP can be approximated as $\langle \phi_i, y \rangle, i = 1...n$ .n Since the correlations are approximately preserved before and after the random projection, the OMP selection is also robust under such random sub-sampling.

OLS differs from OMP in its selection criterion. In OLS, the selection is based on the following expression

$$\| y - \Phi(\Phi^T \Phi)^{-1} \Phi^T y \|_2 \tag{23}$$

After random sub-sampling, it takes the form

$$\| Ay - A\Phi((\mathrm{A}\Phi)^T (A\Phi))^{-1} (A\Phi)^T Ay \|_2 \tag{24}$$

In expression (24) the term $((\mathrm{A}\Phi)^T (A\Phi))$ is the inner product between the columns of Φ, therefore according to GRIP $((\mathrm{A}\Phi)^T (A\Phi)) \approx (\Phi^T \Phi)$. Similarly, $(A\Phi)^T Ay \approx \Phi^T y$. With these approximations, (24) can be approximated as

$$\begin{aligned}&\| Ay - A\Phi(\Phi^T\Phi)^{-1}\Phi^T y \|_2 \\ &= \| A(y - \Phi(\Phi^T\Phi)^{-1}\Phi^T y) \|_2 \\ &\approx \| y - \Phi(\Phi^T\Phi)^{-1}\Phi^T y \|_2 \text{ , since RIP holds}\end{aligned}$$

Therefore, the selection criterion for OLS is also approximately preserved under random projections. The main steps of the OLS algorithm are the selection and the signal estimation steps. It has already been shown that the least squares estimation step is robust to random sub-sampling. As the selection step is also approximately robust, we can claim that the OLS algorithm is also approximately robust to random projections.

The block/group versions of the OMP and OLS algorithms have the same selection criterion but instead of choosing one index a group of indices are chosen. As the selection criterion does not change for the group/block versions, these algorithms are thus robust to random dimensionality reduction as well.

The main criterion of the FSC and the FGSC classification algorithms is the class-wise error. It has already been shown that the class-wise error is approximately preserved after random projection. Therefore, the classification results before and after projection will remain approximately the same.

5.4.4 Nearest Subspace Classifier

The classification criterion for the NSC is the norm of the class-wise error expressed as

$$\| \varepsilon_k \|_2 = \| (V_k(V_k^T V_k + \lambda I)^{-1} V_k^T - I) v_{k,test} \|_2$$

We need to show that the class-wise error is approximately preserved after a random dimensionality reduction. When both the training and the test samples are randomly projected by a matrix A, the class-wise error takes the form

$$
\begin{aligned}
&\| (AV_k((AV_k)^T(AV_k)+\lambda I)^{-1}(AV_k)^T - I)Av_{k,test} \|_2 \\
&= \| AV_k((AV_k)^T(AV_k)+\lambda I)^{-1}(AV_k)^T Av_{k,test} - Av_{k,test} \|_2 \\
&\approx \| AV_k(V_k^T V_k + \lambda I)^{-1} V_k^T v_{k,test} - Av_{k,test} \|_2, \text{ since GRIP holds} \\
&= \| A(V_k(V_k^T V_k + \lambda I)^{-1} V_k^T v_{k,test} - v_{k,test}) \|_2 \\
&\approx \| V_k(V_k^T V_k + \lambda I)^{-1} V_k^T v_{k,test} - v_{k,test} \|_2, \text{ since RIP holds}
\end{aligned}
$$

Since the norm of the class-wise error is approximately preserved under random dimensionality reduction, the classification results will also remain approximately the same.

5. 5 Experimental Results

In subsection 5.5.3, we mentioned the two challenges that Compressive Classifiers should meet. First and foremost, they should have classification accuracy comparable to traditional classifiers. Experiments for general purpose classification were carried out on some benchmark databases from the University of California Irvine Machine Learning (UCI ML) repository [36] to compare the new classifiers (SC, FSC, GSC, FGSC and NSC) with the well-known NN. We chose those databases that do not have missing values in feature vectors or unlabeled training data. The results are tabulated in Table 5-1. The results show that the classification accuracy from the new classifiers is better than that of NN.

Table 5-1: Recognition Accuracy

Dataset	SC	FSC-OMP	FSC-OLS	GSC	FGSC-BOMP	FGSC-GOMP	FGSC-BOLS	FGSC-GOLS	NSC	NN-Euclid	NN-Cosine
Page Block	94.78	94.64	94.48	**95.66**	**95.66**	**95.66**	**95.66**	**95.66**	95.01	93.34	93.27
Abalone	**27.17**	27.29	27.05	**27.17**	26.98	26.98	26.98	26.98	27.05	26.67	25.99
Segmentation	**96.31**	96.10	95.58	94.09	94.09	94.09	94.09	94.09	94.85	**96.31**	95.58
Yeast	57.75	57.54	57.00	58.94	58.36	58.36	58.01	58.01	**59.57**	57.71	57.54
German Credit	69.30	70.00	70.06	**74.50**	**74.50**	**74.50**	**74.50**	**74.50**	72.6	**74.50**	**74.50**
Tic-Tac-Toe	78.89	78.28	78.89	**84.41**	**84.41**	**84.41**	**84.41**	**84.41**	81.00	83.28	82.98
Vehicle	65.58	66.49	66.80	73.86	71.98	71.98	72.19	72.19	**74.84**	73.86	71.98
Australian Cr.	85.94	85.94	85.94	**86.66**	**86.66**	**86.66**	**86.66**	**86.66**	**86.66**	**86.66**	**86.66**
Balance Scale	93.33	93.33	93.33	**95.08**	**95.08**	**95.08**	**95.08**	**95.08**	**95.08**	93.33	93.33
Ionosphere	86.94	86.94	86.94	**90.32**	**90.32**	**90.32**	**90.32**	**90.32**	**90.32**	**90.32**	**90.32**
Liver	66.68	65.79	67.24	70.21	70.21	70.21	70.21	70.21	70.21	69.04	69.04
Ecoli	81.53	81.53	81.53	**82.88**	**82.88**	**82.88**	**82.88**	**82.88**	**82.88**	80.98	81.54
Glass	68.43	69.62	68.43	70.19	**71.02**	**71.02**	**71.02**	**71.02**	69.62	68.43	69.62
Wine	85.62	85.62	84.83	85.62	**85.95**	**85.95**	84.83	84.83	82.58	82.21	82.21
Iris	**96.00**	**96.00**	**96.00**	**96.00**	**96.00**	**96.00**	**96.00**	**96.00**	**96.00**	**96.00**	**96.00**
Lymphography	85.81	85.81	85.46	**86.42**	**86.42**	**86.42**	**86.42**	**86.42**	86.42	85.32	85.81
Hayes Roth	40.23	43.12	43.12	41.01	**43.12**	**43.12**	**43.12**	**43.12**	43.12	33.33	33.33
Satellite	80.30	80.30	80.99	82.37	**82.37**	**82.37**	**82.37**	**82.37**	80.30	77.00	77.08
Haberman	40.52	40.85	40.85	**43.28**	**43.28**	**43.28**	43.12	43.12	46.07	57.40	56.20

The best recognition result for each database is shown in bold. Table 5-1 shows that the classification accuracy from the proposed classifiers is better than NN almost always (except yeast and vehicle). The GSC and the FGSC classifiers give the best results in most number of cases.

The second challenge the Compressive Classifiers should meet is that their classification accuracy should approximately be the same, when sparsifiable data is randomly sub-sampled by RIP matrices. In Section 5.4, we proved the robustness of these classifiers. The experimental verification of this claim is shown in Tables 5-2 and 5-3. It has already been mentioned (Section 5.4) that images follow RIP with random matrices having i.i.d Gaussian columns normalized to unity. In this paper, we validate our work using two well known image based recognition problems – face and handwriting recognition, with compressive classification. The face recognition experiments were carried out on the Yale B face database. The images stored as 192X168 pixel grayscale images. We followed the same methodology as in [5]. Only the frontal faces were chosen for recognition. One half of the images (for each individual) were selected for training and the other half for testing. The experiments were repeated 5 times with 5 sets of random splits.

The average results of 5 sets of experiments are shown in Table 5-2. The first column of the following table indicates the number of lower dimensional projections (1/32, 1/24, 1/16 and 1/8 of original dimension).

Table 5-2: Recognition Results on Yale B

Dimensionality	SC	FSC-OMP	FSC-OLS	GSC	FGSC-BOMP	FGSC-GOMP	FGSC-BOLS	FGSC-GOLS	NSC	NN-Euclid	NN-Cosine
30	82.73	82.08	82.08	85.57	83.18	83.18	82.73	82.73	**87.68**	70.39	70.16
56	92.60	92.34	92.34	**92.60**	91.83	91.83	91.44	91.44	91.83	75.45	75.09
120	95.29	95.04	95.04	**95.68**	95.06	95.06	93.22	93.22	93.74	78.62	78.37
504	**98.09**	97.57	97.19	**98.09**	97.21	97.21	95.06	95.06	94.42	79.13	78.51
Full	**98.09**	**98.09**	**98.09**	**98.09**	**98.09**	**98.09**	95.29	95.29	95.05	82.08	82.08

Table 5-2 shows that the new compressive classifiers are far better than the NN classifiers in terms of recognition accuracy. The Group Sparse Classifier gives by far the best results (at a relatively higher computational cost). All the classifiers are relatively robust to random sub-sampling. The results are at par with the ones obtained from the previous study on Sparse Classification (SC) [19]. We do not compare these classifiers with the Support Vector Machine (SVM), since it is already shown in [19] that the SC outperforms SVM (in terms of recognition accuracy) with random projections.

Finally, the experiments on handwritten character recognition were carried out on the well-known United States Postal Service (USPS) database of handwritten numerals. The images are of size 16X16 pixels. In the following table, dimensionality reduction is carried from the original dimension to 40, 80, 120 and full.

Table 5-3: Recognition Results on USPS

Dimensionality	SC	FSC-OMP	FSC-OLS	GSC	FGSC-BOMP	FGSC-GOMP	FGSC-BOLS	FGSC-GOLS	NSC	NN-Euclid	NN-Cosine
40	**94.47**	94.41	94.17	92.02	92.02	92.02	92.02	92.02	90.14	94.02	93.33
80	**94.77**	**94.77**	94.46	94.66	94.66	94.66	93.97	93.97	93.33	94.17	94.17
120	94.77	94.77	94.77	94.77	94.66	94.66	94.66	94.66	**95.91**	95.01	95.01
Full	94.77	94.77	94.77	94.77	94.77	94.77	94.77	94.77	**95.91**	95.01	95.01

Table 5-3 shows that the NSC gives the best recognition results. The results are very robust to random dimensionality reduction, even for a very small number of projections (40).

Results from Table 5-1 indicate that these classifiers can be used for general purpose classification. Tables 5-2 and 5-3 experimentally corroborate the fact that the proposed classifiers are robust to dimensionality by random projection for inputs that follow the RIP (and consequently) the GRIP conditions.

5. 6 Conclusion

This paper proposes alternatives to data-dependent dimensionality reduction and classification methods. Data-dependency poses certain practical problems as mentioned in the Introduction. To overcome these problems, we advocate employing Random Projections (RP) as a means of data-independent dimensionality reduction. Previous studies have employed RP dimensionality reduction but only in conjunction with Nearest Neighbour (NN) classification. Recently, some classifiers [5, 19-21] have been proposed that perform better than NN when used with RP dimensionality reduction. These classifiers are analyzed in this paper, and a new one is proposed which we term as Fast Sparse Classifier. We formally prove the robustness of these classifiers to RP dimensionality reduction.

In traditional classification, data are first collected and dimensionality reduction is then applied on these data. Classification is then carried out on the dimensionally reduced data. Compressive Classification can be applied in this fashion, but to save resources, it is more effective to incorporate it within the data acquisition process. This procedure precludes unnecessary collection of high-dimensional samples; the classification can thus be applied directly on the compressed (random lower dimension projected) samples. The Compressive Sampling Camera [37] is one such device that can directly collect compressed samples of natural scenes.

A seminal study revealed that face recognition by humans is a holistic process [38], i.e. we remember the entire face image simultaneously instead of remembering in parts. Moreover, the human visual system employs random projections to reduce aliasing [39]. Motivated by this work, random projection was employed in computer vision tasks to reduce aliasing as well [40, 41]. In this work, the random projection mimics the data acquisition step of the human visual system.

This study shows that compressive classifiers can be effectively employed for general classification tasks, but emphasizes that their main advantage over other classifiers is their robustness to data that are dimensionally reduced by random projections. Compressive classifiers are here shown to yield robust recognition results for different image classification tasks such as face and character recognition. Among the different classifiers discussed, there is no clear winner. The recognition accuracy of a classifier depends on the problem it is applied to. For example, the Group Sparse Classifier gives the best results for face recognition problems, whereas the Nearest Subspace Classifier gives the best results for the character recognition task. What we can conclude is that all the classifiers discussed are robust to RP dimensionality reduction.

5. 6 References

[1] M. Davenport, M. Duarte, M. Wakin, J. Laska, D. Takhar, K. Kelly, and R. Baraniuk, “The smashed filter for compressive classification and target recognition”, Computational Imaging V at SPIE Electronic Imaging, pp. 326–330, 2007.

[2] D. L. Donoho, “Compressed sensing,” IEEE Transactions on Information Theory, Vol. 52 (4), pp. 1289–1306, 2006.

[3] E. J. Cand`es and T. Tao, “Near optimal signal recovery from random projections: Universal encoding strategies?”, IEEE Transactions on Information Theory, Vol. 52 (12), pp. 5406–5425, 2006.

[4] E. Skubalska-Rafajłowicz, “Random projection RBF nets for multidimensional density estimation”, Int. J. Appl. Math. Comput. Sci., 2008, Vol. 18 (4), pp. 455–464, 2008.

[5] D. Fradkin, and D. Madigan, D” Experiments with random projection for machine learning”, ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 517-522, 2003.

[6] X. Y. Jing, D. Zhang, and Y. Y. Tang, “An Improved LDA Approach”, IEEE Transactions on Systems, Man, And Cybernetics—Part B: Cybernetics, Vol. 34 (5), pp. 1942-1951, 2004.

[7] J. Peng, P. Zhang, and N. Riedel, “Discriminant Learning Analysis”, IEEE Trans. on Systems, Man, And Cybernetics—Part B: Cybernetics, Vol. 38 (6), pp.1614-1625, 2008.

[8] D. Q. Dai and P. C. Yuen, “Face Recognition by Regularized Discriminant Analysis”, IEEE Transactions on Systems, Man, And Cybernetics—Part B: Cybernetics, Vol. 37 (4), pp.1080-1085, 2007.

[9] X. Jiang, B. Mandal, and A. Kot, “Eigenfeature Regularization and Extraction in Face Recognition”, IEEE Transactions on Pattern Analysis And Machine Intelligence, Vol. 30 (3) pp. 686-694, 2008.

[10] N. Kwak, “Principal Component Analysis Based on L1-Norm Maximization”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 30 (9) pp. 1672-1680, 2008.

[11] H. Cevikalp, M. Neamtu, and A. Barkana, “The Kernel Common Vector Method: A Novel Nonlinear Subspace Classifier for Pattern Recognition”, IEEE Transactions on Systems, Man, And Cybernetics—Part B: Cybernetics, Vol. 37 (4), pp937-951, 2007.

[12] H. L. Wei and S. A. Billings, “Feature Subset Selection and Ranking for Data Dimensionality Reduction”, IEEE Transactions on Pattern Analysis And Machine Intelligence, Vol. 29 (1) pp. 162-166, 2007.

[13] M. Doumpos, C. Zopounidis, and V. Golfinopoulou, “Additive Support Vector Machines for Pattern Classification”, IEEE Transactions on Systems, Man, And Cybernetics—Part B: Cybernetics, Vol. 37 (3), pp.540-550, 2007.

[14] M. Skurichina and R. P. W. Duin, “Bagging, Boosting and the Random Subspace Method for Linear Classifiers”, Pattern Analysis & Applications Vol. 5, pp.121–135, 2002.

[15] T. J. Chin and D. Suter, “Incremental Kernel Principal Component Analysis”, IEEE Transactions on Image Processing, Vol. 16, (6), pp. 1662-1674, 2007.

[16] H. Zhao and P. C. Yuen, “Incremental Linear Discriminant Analysis for Face Recognition”, IEEE Transactions on Systems, Man, And Cybernetics—Part B: Cybernetics, Vol. 38 (1), pp.210-221, 2008.

[17] D. Masip, À. Lapedriza, and Jordi Vitrià, “Boosted Online Learning for Face Recognition”, ”, IEEE Transactions on Systems, Man, And Cybernetics—Part B: Cybernetics, Vol. 39 (2), pp.530-538, 2009.

[18] J. Haupt, R. Castro, R. Nowak, G. Fudge, and A. Yeh, “Compressive sampling for signal classification,” Asilomar Conf. Signals, Systems and Computers, Pacific Grove, pp. 1430-1434, 2006.

[19] Y. Yang, J. Wright, Y. Ma and S. S. Sastry, “Feature Selection in Face Recognition: A Sparse Representation Perspective”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 31 (2), pp. 210-227, 2009.

[20] A. Majumdar and R. K. Ward, “Generalized Sparse Classifier: Application to Single Image per Person Face Recognition”, submitted to Pattern Recognition Letters.

[21] A. Majumdar and R. K. Ward, “Classification via Group Sparsity Promoting Regularization”, IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 873-876, 2009.

[22] A. Majumdar and R. K. Ward, “Fast Group Sparse Classifier” IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, Victoria, Canada, August 2009 (accepted).

[23] A. Majumdar and R. K. Ward, “Nearest Subspace Classifier” submitted to International Conference on Image Processing 2009.

[24] T. Blumensath, M. E. Davies; “On the Difference between Orthogonal Matching Pursuit and Orthogonal Least Squares”, manuscript 2007.

[25] H. Zou and T. Hastie, “Regularization and variable selection via the elastic net”, Journal of Royal Statistical Society B., Vol. 67 (2), pp. 301-320, 2005.

[26] Y. Kim, J. Kim, and Y. Kim, “Blockwise sparse regression”, Statistica Sinica, Vol. 16, pp. 375-390, 2006.

[27] L. Meier, S. van de Geer, and P. Buhlmann, “The group lasso for logistic regression”, J. R. Statist. Soc. B, Vol. (70), pp. 53-71, 2008.

[28] B. Turlach, W. Venables, and S. Wright, “Simultaneous Variable Selection”, Technometrics, Vol. 47, pp. 349-363, 2005.

[29] M. Yuan and Y. Lin, “Model selection and estimation in regression with grouped variables”, J. R. Statist. Soc. B, Vol. 68, pp. 49-67, 2006.

[30] J. Huang and T. Zhang, “The Benefit of Group Sparsity” arXiv:0901.2962v1.

[31] E. van den Berg, M. Schmidt. M. Friedlander, K. Murphy, “Group sparsity via linear-time projection”, Technical Report TR-2008-09, Department of Computer Science, University of British Columbia, June 2008.

[32] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad. “Orthogonal Matching Pursuit: Recursive function approximation with applications to wavelet decomposition”, Asilomar Conf. Sig., Sys., and Comp., 1993

[33] Y. C. Eldar and H. Bolcskei, “Block Sparsity: Uncertainty Relations and Efficient Recovery”, International Conference on Acoustics, Speech, and Signal Processing 2009 (accepted).

[34] J. Haupt and R. Nowak, “Compressive Sampling for Signal Detection”, International Conference on Acoustics, Speech, and Signal Processing, Vol.3, pp.III-1509-III-1512, 2007.

[35] http://www.dsp.ece.rice.edu/cs

[36] http://archive.ics.uci.edu/ml/

[37] http://dsp.rice.edu/cscamera

[38] P. Sinha, B. Balas, Y. Ostrovsky, R. Russell, Face Recognition by Humans: 19 Results All Computer Vision Researchers Should Know About, Proceedings of the IEEE, Vol. 94 (11), pp. 1948-1962, 2006.

[39] J.I. Yellott Jr., “Spectral analysis of spatial sampling by photoreceptors: Topological disorder prevents aliasing”, Vision Research, Vol. 22, pp. 1205-1210, 1982.

[40] M. A. Z. Dippe and E. H. Wold, “Antialiasing Through Stochastic Sampling”, ACM SIGGRAPH, Vol. 19 (3), pp. 69-78, 1985.

[41] D. Alleysson, S. Süsstrunk, and J. Hérault, “Linear Demosaicing inspired by the Human Visual System”, IEEE Transactions on Image Processing, Vol. 14(4), pp.1-10, 2005

CHAPTER 6: FACE RECOGNITION FROM VIDEO: RANDOM PROJECTIONS AND HYBRID CLASSIFICATION[5]

6. 1 Introduction

This paper focuses on face recognition from video sequences. The aim is to develop a face recognition system for high-security applications such as person authentication in bank ATMs or automatic access control in offices. Such a recognition system should fulfill the following practical constraints:

1. The system should have very high recognition accuracy (above 99%).
2. The computational speed of the system should be near real-time.
3. The cost of updating (dimensionality reduction and training) the system with new data should be minimal.

The first requirement is obvious. The second requirement constrains the system to have a fast testing (operational) time. The third requirement constrains the system to have a fast training time. When new data are added, the third constraint is best satisfied if the dimensionality reduction and training on this new data are adaptive in the sense that they do not require access to previous data. This precludes use of all data-dependent dimensionality reduction methods such as Principal Component Analysis, Linear Discriminant Analysis, Locally Linear Embedding, etc., as well as standard classification tools like Support Vector Machine and Artificial Neural Network, since they all need to retrain the system every time new data are added.

There are two standard approaches to address the problem of face recognition from video. The first is the image-to-image approach – this approach applies image-based face recognition

[5] A version of this chapter has been submitted for publication. A. Majumdar, R. K. Ward and P. Nasiopoulos, “Face Recognition from Video: Random Projections and Hybrid Classification”.

techniques to the video sequence on a frame-by-frame basis. The second approach is the video-to-video approach which uses dynamic machine learning techniques for face recognition.

All previous methods, unfortunately, fail to satisfy all three constraints simultaneously. Earlier studies in video-based face recognition like [1-6] use PCA for dimensionality reduction. In [7] a Singular Value Decomposition based dimensionality reduction is employed while [8], use Independent Component Analysis for the same purpose. All these studies use data dependent dimensionality reduction methods and do not satisfy constraint 3. A recent work [9] uses Support Vector Machine for classification, and also violates constraint 3.

Video-to-video based recognition approaches (10-11) satisfy constraints 1 and 3. However, these methods are computationally expensive and do not satisfy constraint 2.

Some previous studies [13 and 14] use Active Appearance Models (AAM) for face recognition in video and they achieve high recognition accuracy. However, AAM is a semi-automatic technique and requires human intervention; therefore it can not be used for automatic face authentication.

There are no existing methods that can satisfy all the three constraints. In this paper, we propose a new video-based face recognition method that satisfies all three constraints. We propose the use of a lesser popular dimensionality reduction technique – Random Projection. The Random Projection method is very cheap to compute and is data-independent. Our classification algorithm is based on a fusion of results from video-to-video and image-to-image based methods, with each of these approaches being data-independent.

In the rest of the paper we first discuss our proposed dimensionality reduction. In Section 6.3, we discuss our classification method. Section 6.4 describes the implementation details of our proposed method. In Section 6.5, we show the experimental results. Finally in Section 6.6, we discuss the conclusions of this work.

6. 2 Non-Adaptive Dimensionality Reduction

Adaptive dimensionality reduction methods such as PCA, LDA and LLE have been traditionally widely used in face recognition. Such techniques preserve the structure of the original data required for classification. However, such dimensionality reduction techniques are adaptive and cannot, thus, be employed in practical scenarios for the reasons mentioned above. We are seeking a dimensionality reduction step that is non-adaptive, i.e., independent of previous training data, but at the same time preserving the structure of the data that is necessary for classification.

A survey of dimensionality reduction methods [17] provides the detailed information pertaining to adaptive linear methods like PCA, Factor Analysis, Projection Pursuit and their non-linear counterparts, but cursorily mentions Random Projection (RP). We are interested in a RP as it is a non-adaptive method. Since, RP is not a much explored field in dimensionality reduction, a few theoretical and applied studies that use RP for dimensionality reduction will be briefly reviewed.

Almost all theoretical studies on RP dimensionality reduction are extensions of the Johnson-Lindenstrauss lemma. In [16] it is showed that by setting each column of the projection matrix to be i.i.d (independently and identically distributed) Gaussian and orthonormalizing the columns by Gram-Schmidt lead to a RP projection matrix well suited for dimensionality reduction of Gaussian mixtures. In [17] it is proposed a novel method to create an RP matrix from a Bernoulli type distribution. It is shown in [18] that, by creating a simple RP matrix with its column drawn from i.i.d (independently and identically distributed) Gaussian and projecting it to a suitable number of dimensions, it is possible to preserve some important structure in the data, like pairwise distances, distances from a line, areas of triangles etc. In [19] it is showed that, similarity (cosine distances) is approximately preserved under RP.

Theoretical studies in RP [16 and 20] compares RP with PCA as a dimensionality reduction scheme for machine learning problems. These studies find that PCA is better than RP for moderate number of lower dimensional projections. However, for a very low number of projections, PCA cannot preserve the structure of the data and the results are drastically worse. The results from RP degrade smoothly as the number of projections is decreased. As the number of projections increase the recognition results obtained from RP become comparable to that of PCA. These theoretical findings are corroborated by practical studies [21-23]. However, while applying RP for practical problems [21-23], it was found that a single RP is unstable and multiple RP matrices are required for stabilizing the results.

Random projection is a lesser used dimensionality reduction method. Previous studies do not indicate how many projections are required for good classification results. Theoretical work advocates a pessimistic (large) number of projections, thus previous practical studies have fixed the number of lower dimensional projections by trial and error. To determine the number of projection, we consult the Compressive Sampling (CS) literature [24, 25]. First we discuss how CS dimensionality reduction can be employed for sparse signals. Then, we describe how to practically achieve RP dimensionality reduction following CS concepts.

Compressive Sampling (CS) states that if a signal is sparse, then all the information in the original signal remains embedded after the signal undergoes a lower dimensional random projection. This is in the sense that the original signal can be reconstructed from its random projections by tractable algorithms. Consider an *'n'* dimensional signal *'x'* that is *k-sparse*. All the information in the original signal x can be conserved after it undergoes a lower dimensional (*m*) random projection A :

$b = Ax \in \mathbb{R}^m$, A $\in$ $\mathbb{R}^{m \times n}$ – Random Projection matrix

To reconstruct the original signal x from its lower random projection *b*, [26], has recently shown that $m(\geq 2k+1)$ projections are sufficient. The reconstruction is done by the following optimization process:

$$\min \|x\|_0 \;\; \textit{such that } Ax = b$$

Thus $m(= 2k+1)$ is the theoretical lower limit of the minimum number of projections which ensure that all the information in the original *k-sparse* signal is retained in its projected vector. Solving the above l_0 minimization problem, however, is an NP hard problem. Approximate methods to solve the l_0 problem typically require a larger number of samples than *2k+1*. Practical studies [27] have demonstrated that perfect signal reconstruction can be obtained with $m = C \cdot k$, (where C=5) random projections.

The above applies to signal construction. This is a more ambitious goal than signal classification, which is our interest. In [28] it is shown that it is possible to achieve good classification at a much lower dimensional random projection than indicated by the CS literature for signal reconstruction.

To apply CS techniques for dimensionality reduction, the first step would be to sparsify the image vector *I* via some orthogonal transform Φ:

$$y = \Phi I \tag{1}$$

where *y* is the resulting sparse vector.

The sparsity of the signal y is estimated by looking at the energy concentration of its coefficients. Figure 6-1 shows the sparsity of a face image in the DCT domain. About 98% of the signal energy is concentrated in the top 10% DCT coefficients.

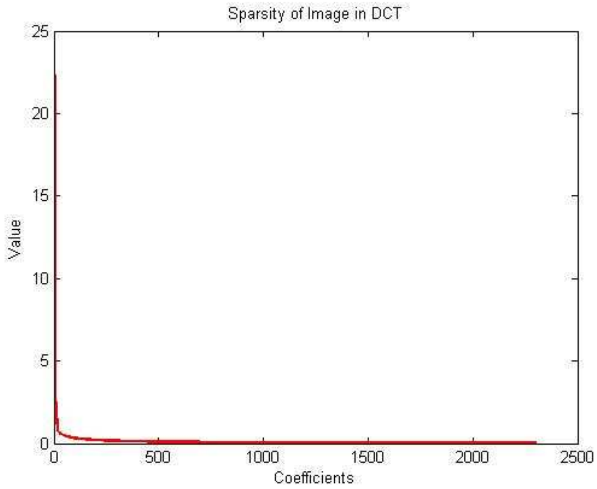

Figure 6-1: Sparsity of a Face Image in DCT Domain

To reduce the dimensionality of the sparsified vector *y* by a random projection matrix *A*, we use the Restricted Isometric Property [25]:

$$(1-\delta) \| y \|_2 \leq \| Ay \|_2 \leq (1+\delta) \| y \|_2 \tag{2}$$

where *A* is the RP matrix. Condition (2) ensures that all the information in the sparse high dimensional vector could be retained in the lower dimensional projection when δ is zero. The energy in the projected signal *Ay* could thus be made very close to that of the original *y*, by increasing the number of projections until δ is small enough. This would determine the number of projections needed.

Condition (2) holds for all sparse vectors. We show below that this condition also holds for images represented in the pixel domain, even though these are not sparse.

Let *A* be a RIP matrix for the sparse vector coefficient vector *y*:

$$(1-\delta)\,||\,y\,||_2 \leq ||\,Ay\,||_2 \leq (1+\delta)\,||\,y\,||_2 \qquad (3)$$

Substituting *y* from (1)

$$(1-\delta)\,||\,I\,||_2 \leq ||\,A\Phi I\,||_2 \leq (1+\delta)\,||\,I\,||_2 \quad \because \ \Phi \text{ is orthogonal} \qquad (4)$$

The original RIP matrix *A* is i.i.d Gaussian. Therefore, the linear combination B (=*AΦ*) of it is also i.i.d Gaussian, i.e., the matrix *B* = *AΦ* is i.i.d Gaussian. Instead of creating *A* and then post-multiplying it by *Φ*, we can directly create an i.i.d Gaussian matrix *B* and use it for random dimensionality reduction of the original image I; there will always be a matrix $A = \Phi^T B$ which will be a RIP matrix for *y*. Therefore, (4) can be re-written as:

$$(1-\delta)\,||\,y\,||_2 \leq ||\,BI\,||_2 \leq (1+\delta)\,||\,y\,||_2 \qquad (5)$$

In equation (5), the value of δ depends on the dimensionality of the original vector, its sparsity and the number of lower dimensional projections made [25]. Empirical studies have shown that, if the number of lower dimensional random projections is equal to greater than 5 times the sparsity of the original signal, the value of δ is sufficiently small. In this work, we have used random projections about 6 times the sparsity of the signal.

Equation (5) ensures that the RIP condition holds for random projection of the non-sparse image as well. Thus, it is not required to apply the intermediate sparsifying step. Once we are aware of the sparsity of the image, RP dimensionality reduction can be directly applied to the image. The sparsity of the signal must be known for determining how many lower dimensional projections are to be made.

6. 3 Classification

We employ a fusion of results from 2 methods – an image-to-image based method that we proposed and a video-to-video based scheme. We employ a Hidden Markov Model for the video-to-video based method. Our image-to-image method is based on a group sparsity promoting optimization which we call GOMP.

6.3.1 Video-to-Video Method

Liu and Chen's, 2003 HMM based recognition method is fast and highly accurate, but it uses PCA dimensionality reduction. Consequently, the overall system does not satisfy constraint (3). To overcome this problem we use Random Projection (RP) for dimensionality reduction. Previous studies [22 and 23] employed RP image-based face recognition methods. In our work, RP dimensionality reduction is applied for the first time in a dynamic learning framework (HMM).

We use the same HMM scheme proposed by [3] in our work. HMM is parameterized by the triplet , and has a finite number of states (N) - $\{S_1, S_{2,..}S_N\}$ which are hidden from the observer. At each instant of time, the chain occupies one of the states given by q_t , $1 \le t \le T$, where T is the total length of the sequence.

The initial state distribution is given by π, where $\pi_i = P(q_1 = S_i),\ i = 1$ to N .

From one instance to the next, the transition occurs according to a Markov state transition matrix A. $A = \{a_{i,j}\}$, where

$$a_{i,j} = P(q_t = S_j \mid q_{t-1} = S_i),\ 1 \le i, j \le N$$
$$\text{and} \sum_{j=1}^{N} a_{i,j} = 1$$

These state vectors (S) are not observable; only the observation vectors O is measurable. In [3] the observation vector 'O' is comprised of Principal Components of a frame. In our case 'O' is the RP of a frame.

B ($= b_i(O)$) is the observation probability density function where

$b_i(O_t) = p(O_t \mid q_t = S_i)$

Generally, $b_i(O)$ is modeled as a Gaussian Mixture Model (GMM).

The problem now is to estimate the triplet (A,B,π), given a sequence of observations O_t, $t = 1$ to T . This can be done by the Expectation Maximization algorithm, which is the standard method for HMM parameter estimation.

Suppose there are video sequences of C people. During training, one video sequence for each person is available (if we consider the bank ATM scenario, this video sequence is shot when a customer opens a new bank account). Each frame is projected to a lower dimensional subspace by an RP projection matrix. We fit an HMM for each video sequence and obtain a triplet (A_k, B_k, π_k), where $k = 1$ to C .

During testing, the video sequence of the person is recorded (e.g., by a camera at the ATM) and the frames are projected to a lower dimensional subspace by the same RP matrix. Then, the likelihood score of the observation vectors (O), given each HMM, is computed as $P(\mathrm{O} \mid (A_k, B_k, \pi_k))$. The video sequence is assigned to a person having the maximum likelihood score.

6.3.2 Proposed Image-to-Image Method

The aim is to design a classifier which does not require re-training when new data are added. The Nearest Neighbour (NN) classification is an option. However, NN does not yield very good recognition results. A classifier that does not need to be trained and gives much better recognition results than NN was recently proposed in [29]. This classifier gives good results, but its optimization function is relatively slow. In this work, we propose an alternate optimization procedure that keeps the accuracy of the previous work but is considerably faster.

Using all the video frames for training is not efficient as they contain redundant information. In [2] it is suggested that instead of using all the frames from the video sequence, one can randomly select the frames for classification. The classification accuracy is shown not to degrade much; at the same time the classification speed is considerably improved. Following this work, random selection of frames is employed for our image-to-image classification algorithm.

The method in [29] is based on the assumption that the training samples of a particular class approximately form a linear basis for a new test sample belonging to the same class. If v_{test} is the test sample that belongs to the kth class then,

$$\begin{aligned} v_{k,test} &= \alpha_{k,1}v_{k,1} + \alpha_{k,2}v_{k,2} + \ldots + \alpha_{k,n_k}v_{k,n_k} + \varepsilon_k \\ &= V_k\alpha_k + \varepsilon_k, \text{ where } V_k = [v_{k,1} \,|\, v_{k,2} \,|\, \ldots \,|\, v_{k,n_k}] \text{ and } \alpha_k = [\alpha_{k,1}, \alpha_{k,2}, \ldots, \alpha_{k,n_k}]' \end{aligned} \quad (6)$$

where $v_{k,i}$ are the training sample vectors and ε_k is the approximation error vector.

The above equation expresses the assumption in terms of the training samples of a single class. Alternately, it can be expressed in terms of all the training samples so that

$$v_{k,test} = V\alpha + \varepsilon \quad (7)$$

where matrix $V = [V_1 \,|\, V_2 \,|\, \ldots \,|\, V_C]$ and the coefficient vector $\alpha = [\alpha'_1 \,|\, \alpha'_2 \,|\, \ldots \,|\, \alpha'_C]'$.

The problem is to identify the class v_{test} given the training samples in *V*. This requires solving the above equation, with a constraint that the only α_k corresponding to the actual class should have finite values and the rest should be all zeros, i.e., in other words solution α is group-sparse. Mathematically, this constraint is expressed as the $l_{1,2}$ norm minimization problem [30]:

$$\begin{aligned} &\min_{\alpha} \|\alpha_1\|_2 + \|\alpha_2\|_2 + \ldots + \|\alpha_C\|_2 \\ &\text{such that } \|v_{k,test} - V\alpha\|_2 < \varepsilon \\ &\text{where } \alpha_i = [\alpha_{i,1}, \alpha_{i,2}, \ldots, \alpha_{i,n_i}], \text{ for i = 1,2,...,C} \end{aligned} \quad (8)$$

Solving the $l_{1,2}$ optimization problem is the workhorse behind the classification algorithm. However, even the fastest solver of the $l_{1,2}$ minimization problem [30] is comparatively slow for our task. To ameliorate this problem we propose a new approximate group sparsity promoting inversion algorithm which we call Group Orthogonal Matching Pursuit (GOMP).

6.3.2.1. Group Orthogonal Matching Pursuit

Our objective is to find a computationally fast solution to the equation

$v_{test} = V\alpha + \varepsilon$, where $V = [V_1 | V_2 | ... | V_C]$, and $\alpha = [\alpha'_1 | \alpha'_2 | ... | \alpha'_C]'$

such that, the solution is group sparse.

The greedy solution algorithm we propose (called Group Orthogonal Matching Pursuit (GOMP)) follows a notation similar to [31]).

The inputs of the algorithm are:

- An dXn dimensional matrix V – d is the dimensionality of the inputs and n is the number of training samples.
- An d dimensional test vector v_{test}.
- The class (1 to m) of each column in V.
- An error estimate ε.

The outputs from this algorithm are:

- A group sparse estimate of the solution α.

The different steps of the GOMP algorithm are as follows:

5. Initialize the residual $r_0 = v$, the index set $\Lambda_0 = \varnothing$, the matrix of chosen atoms $\Phi_0 = \varnothing$, and the iteration counter t = 1.

6. At the iteration = t, find $\lambda_t = \underset{j=1...m}{\arg\max}\, avg(|< r_{t-1}, \varphi_j >|)$

7. Augment the index set $\Lambda_t = \Lambda_{t-1} \cup \{class(\lambda_t)\}$ and the matrix of chosen atoms $\Phi_t = [\Phi_{t-1}\ V_{class(\lambda_t)}]$. Here $\text{class}(\lambda_t)$ refers to all the indices between 1 and N that belong to the class of λ_t .

8. Get the new signal estimate $x_t = \Phi_t^{\dagger} v$. Here $\Phi_t^{\dagger}$ denotes the pseudo inverse of Φ_t .

9. Calculate the new approximation and the residual $a_t = \Phi_t x_t$ and $r = v - a_t$.

10. Increment t and return to step 2 if $\| r \| \geq \varepsilon$.

The proposed GOMP algorithm is a fast approximation of the $l_{1,2}$ problem. The algorithm is based on the similarity (inner product) of the residual vector with the columns of V. In [19] it is showed that inner products between two vectors are approximately preserved under random projection. We already know that the group sparsity promoting inverse problem ($l_{1,2}$ minimization) gives good classification results. Hence, it is realistic to assume that its approximation GOMP will have a performance. Following, [19] result, we are also assured that GOMP will give good results under RP dimensionality reduction as well. Therefore, our classification algorithm is of the form:

1. Solve GOMP given V and $v_{k,test}$.

2. Repeat the following steps for all the classes:

    a. Reconstruct a sample for each class by a linear combination of the training samples in that class via the equation $v_{recon}(i) = \sum_{j=1}^{n_i} \alpha_{i,j} v_{i,j}$.

    b. Find the error between the reconstructed sample and the given test sample by $error(v_{test}, i) = \| v_{k,test} - v_{recon(i)} \|_2$.

3. Once the $error(v_{test}, i)$ for every class i is obtained, choose the class having the minimum error as the class of the given test sample.

6. 4 Proposed Method

In this section, we will discuss the actual implementation of a fast and accurate face recognition system.

6.4.1 Pre-Processing

The first step is to detect the face image in each frame. Let us consider the scenario of a bank ATM. When a customer comes in, his/her face is detected. This is achieved by employing the [32] face detection algorithm which detects only the face part from the forehead to the chin. Consequently, the recognition system becomes invariant to changes in hairstyle. This algorithm runs real time and can process more than 30 frames per second. Once the face is detected in a frame, it is normalized to a 48X48 pixel square.

For the video-to-video approach, face detection is the only necessary pre-processing step. For the image-to-image approach as mentioned earlier, we need one more step so that only a few frames from the entire sequence are selected. We randomly select the frames from the video sequence. We use random jitter sampling [33] in place of ordinary random sampling to guarantee that the chosen frames are not temporally very close to each other.

6.4.2 Dimensionality Reduction

The first step towards classification is dimensionality reduction. For this step, we use RP dimensionality reduction. Five sets of random projections were used for stabilizing the classification results as suggested by [22 and 23].

Five random projection matrices are created by normalizing the columns of an i.i.d Gaussian matrix. After pre-processing, each 48X48 frame is concatenated to form a vector of length 2304. The vector is projected to a lower dimension by the RP projection matrices. Thus, 5 sets of lower dimension vectors are obtained for each frame.

6.4.3 Classification

We propose a fusion of results from two classification methods – the video-to-video and the image-to-image. The theory behind the two classification algorithms has been discussed in Section 6.4. Here, we explicitly discuss how we mix the outputs from the two approaches to come up with the final decision regarding the test sequence.

6.4.3.1 HMM Classification

Let each training video sequence be denoted by S be the sequence (after pre-processing and concatenation to vector). Therefore, S is a matrix with 2304 rows (dimensionality of the input vector) and columns equal to the number of frames in the video sequence. Let R_i, $i = 1$ to 5 be the RP matrices.

1. Make the 5 lower dimensional projections of the sequence S by $I_i = R_i S$, i=1 to 5 .

2. For each projection, create a Hidden Markov Model for the person whose video sequence is S. The model is parameterized by $(A_{c,i}, B_{c,i}, \pi_{c,i})$, where i=1:5 and c denotes the person (class).

3. During testing, the pre-processing and dimensionality reduction steps of the test sequence are the same as during training. There are 5 sets of lower dimensional test sequences $I_{test,i}$ i=1:5.

4. For each lower dimensional projection of the test sequence, compute the likelihood of $I_{test,i}$ for each of the 5 sets of HMM parameters for each person, $L(c,i) = P(I_{test,i} \mid A_{c,i}, B_{c,i}, \pi_{c,i})$, i=1:5 and c=1:C.

6.4.3.2 GOMP Classification

Let P be the number of frames randomly selected for classification. Let R_i, $i = 1$ to 5 be the RP matrices.

1. For each frame I_p, p = 1 to P , repeat the following steps.

2. Project I_p to lower dimension $I_p(i) = R_i I_p$.

3. For each projection, solve the GOMP problem and find the error $e_p(c,i)$ for all the classes $c = 1$ to C .

4. Find the aggregate error for all the frames $error(c,i) = \sum_{p=1}^{P} e_p(c,i)$.

6.4.3.3 Hybrid Classification: Mixing outputs of HMM with GOMP

Now we need to combine the results of HMM and GOMP. There are several ways to combine the results. Performance evaluations have shown that the following combination gave the best recognition results.

1. Find the score for each projection for each person $score(c,i) = L(c,i) \times 1 / error(c,i)$.

2. Find the aggregate the score for each projection $AggScore(c) = \sum_{i=1}^{5} score(c,i)$

3. The test sequence is assigned to the class having the maximum aggregate score.

6.5 Performance Evaluation and Discussion

We tested our hybrid recognition approach on the CMU Faces in Action (FIA) database [34]. This database has both indoor and outdoor shots of each person. We envisage the application of our work in indoor environments, viz. client authentication in ATMs or employee authentication in offices. Therefore, we test our method with the indoor video sequences from the FIA database. The indoor database consists of 20-second videos of face data from 153 participants mimicking a passport checking scenario. There are three video sequences for each person. The gap between video sequences was around 3 months to emulate realistic conditions.

It was mentioned in Section 6.2 that if *k* be the sparsity of a vector, then at least *2k+1* random projections are required to capture all the information of the sparse vector. We found that, the top 20 highest values DCT transform coefficients of 48X48 pixel images carry over 98% of the signal energy. Even though not strictly sparse, we can assume face images to be highly compressible in the DCT domain. Therefore, we start with 40 random projections and keep increasing in multiples of 40 until we are satisfied with the results.

In our image-to-image method, instead of using all the frames in the video sequences, we use jitter sampling to randomly choose the frames. The following graph shows the recognition accuracy from image-to-image approach for different numbers of randomly selected frames. The X-axis shows the number of selected frames and the Y-axis shows the recognition accuracy.

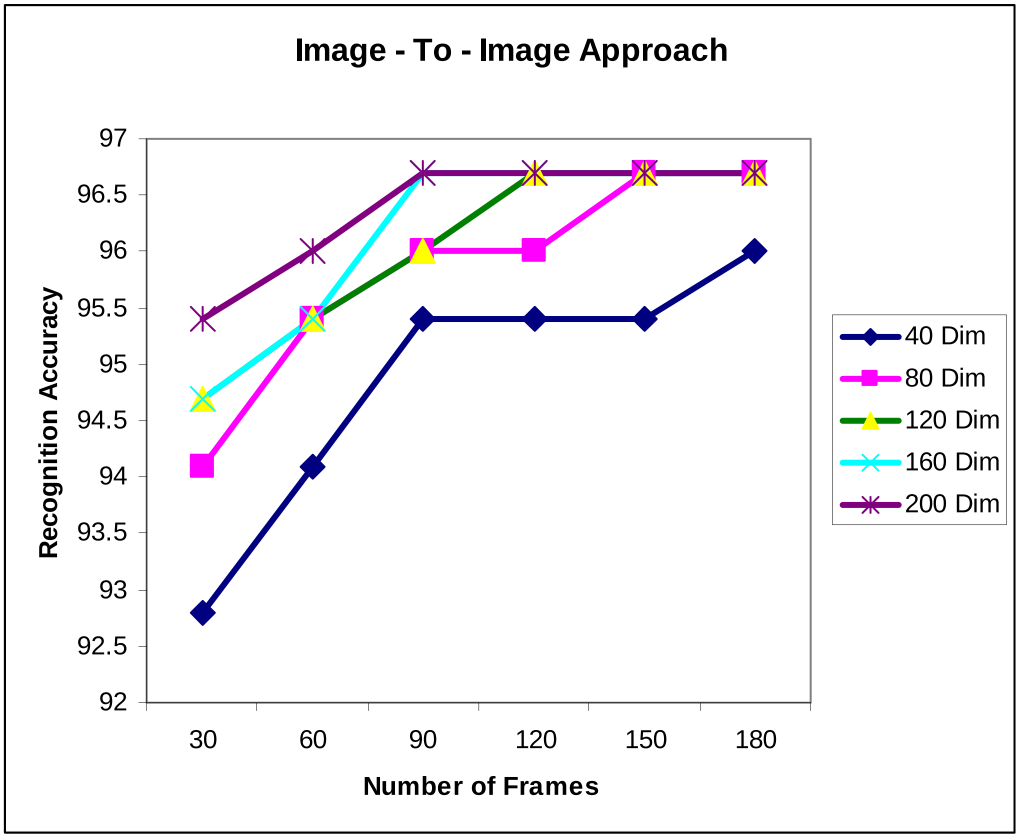

Figure 6-2: Recognition Accuracy vs Number of selected frames

From Figure 6-2, we observe that keeping the dimensionality (number of projections) fixed and increasing the number of frames results in an initial increase in recognition accuracy, but saturates after a while. This is because, with too many frames, some frames look very similar to others and do not add value to the training set in terms of variability.

For the video-to-video approach, the number of hidden states in the HMM must be decided. The number of hidden states is varied to study the effect on overall recognition accuracy keeping the number of observations fixed. From Figure 6-3, we observe that the recognition accuracy increases to a certain extent after which it either stops increasing or begins to fall. This is because, the more the number of hidden states the better is the modeling, but at the same time the greater is the number of HMM parameters to be estimated. Since the length of the video sequence is fixed, as the number of parameters to be estimated increases, the model begins to overfit and consequently the recognition accuracy decreases.

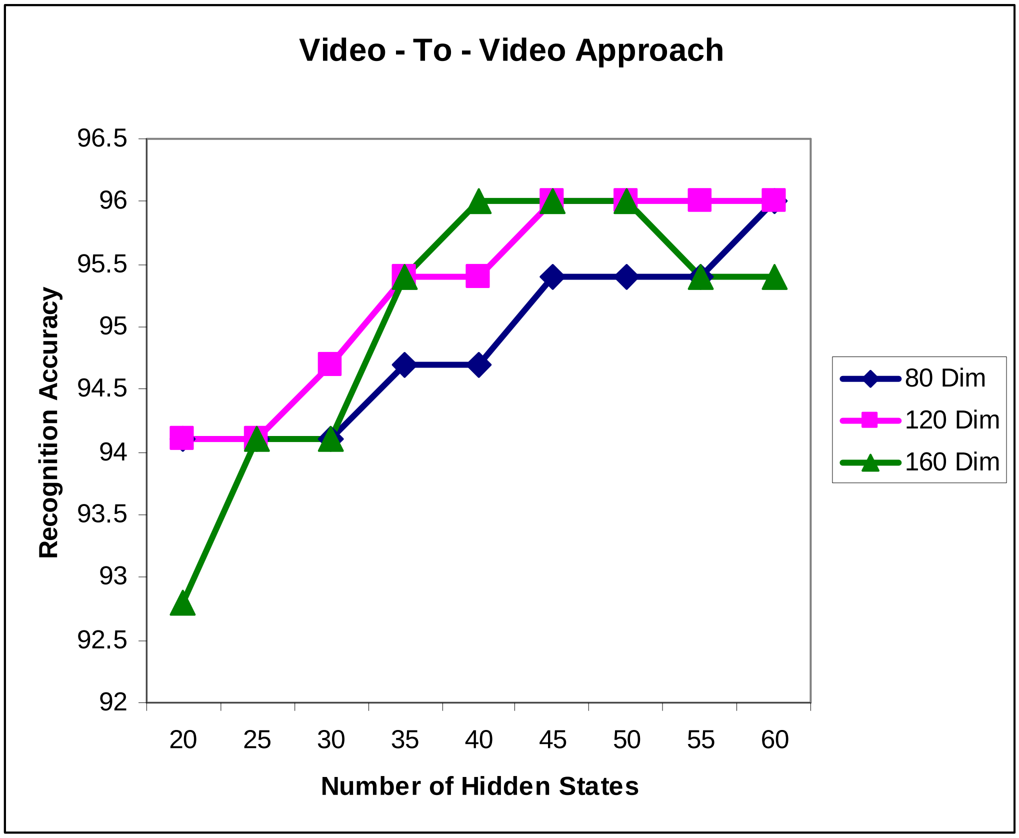

Figure 6-3: Recognition Accuracy vs Number of Hidden States

For our hybrid approach, we combine the results from Figures 6-1 and 6-2. For the image-to-image approach, the best results can be obtained at 120 frames or more. To keep the computational cost minimal we use 120 dimensions and 120 frames. For the video-to-video approach, the best results are obtained for 120 dimensions with 45 hidden states. These are the setting used in our proposed approach.

In Table 6-1 our hybrid recognition methods are compared with 2 video-to-video [3 and 12] based methods and 2 image-to-image [9 and 1] based methods.

Our proposed method outperforms the previous methods by a considerable margin. Compared to state-of-the-art method [3] our approach reduces error by 65%. Moreover, we achieve this high recognition accuracy in spite of the three practical constraints.

Table 6-1: Recognition Accuracy from Different Methods

Method	Recognition Accuracy (%)
Liu and Chen	98.0
Aggarwal et al	96.0
Majumdar and Nasiopoulos	96.7
Price and Gee	92.8
Proposed	99.3

6.6 Conclusion

This paper studies a practical problem in video based face recognition, where new video sequences of new faces are added any time. We propose a computationally feasible method, where the recognition system is data-independent, i.e. cost of updating the system with new data (dimensionality reduction and training) is independent of previous data. The proposed method is shown to give better recognition results compared to other state-of-the-art data-dependent methods in this field. We are able to cut the error by at least 65% from the previous ones.

The data-independence in dimensionality reduction is achieved via Random Projections. Random projections – an alternative to data-driven dimensionality reduction methods that has never been used for video based face recognition before. Our classification method is hybrid, in the sense that it combines video-to-video and image-to-image based recognition results. The video-to-video method uses a person specific Hidden Markov Model. The image-to-image method uses a fast group-sparsity promoting classification method based on a greedy algorithm which we call Group Orthogonal Matching Pursuit – GOMP. The video-to-video and the image-to-image methods are each data-independent, i.e. updating the system does not depend on previous data. We fused the results from these two methods in such a way that the combination is also data-independent.

The proposed GOMP algorithm is a greedy version of the $l_{1,2}$ minimization problem [30]. Although proposed here for classification, GMPL can find potential use in other areas of machine learning where the requirement is to select a group of variables. Such examples arise in multi-response linear regression and multinomial logistic regression; linear models augmented with higher-order interactions; multiple kernel learning; and multi-class Markov random fields and conditional random fields.

6.7 References

[1] J. R. Price and T. E. Gee, “Towards robust face recognition from video”, Applied Imagery Pattern Recognition Workshop, pp. 94-100, 2001.

[2] X. Tang, and Z. Li, “Video based face recognition using multiple classifiers”. IEEE International Conference on Automatic Face and Gesture Recognition, pp. 345-349, 2004.

[3] X. Liu, and T. Chen, “Video-based face recognition using adaptive hidden Markov models”, IEEE Conference on Computer Vision and Pattern Recognition, pp. 340-345, 2003.

[4] T. E. Campos, R. Feris, and R. M. Cesar Junior, “A framework for face recognition from video sequences using gwn and eigenfeature selection”, Workshop on Artificial Intelligence and Computer Vision, pp. 141–145, 2000.

[5] K. C. Lee, J. Ho, M. H. Yang, and D. Kriegman, „Video-Based Face Recognition Using Probabilistic Appearance Manifolds”. IEEE Conference on Computer Vision and Pattern Recognition, pp. 313-320, 2003.

[6] S. A. Berrani, and C. Garcia, „Enhancing face recognition from video sequences using robust statistics”, IEEE Conference on Advanced Video and Signal Based Surveillance, pp. 324-329, 2005

[7] T. J. Chin, U. James, K. Schindler, and D. Suter, “Face Recognition from Video by Matching Image Sets”, Digital Image Computing: Techniques and Applications, pp. 28-28, 2005.

[8] A. Padilha, J. Silva, and R. Sebastião, “Improving Face Recognition by Video Spatial Morphing”, Face Recognition, K. Delac and M. Grgic, Eds, pp.558-577, 2007, I-Tech, Vienna, Austria.

[9] A. Majumdar, and P. Nasiopoulos, P., 2008. Frontal Face Recognition from Video, International Symposium on Visual Computing, pp. 297-306, 2008.

[10] R. Chellappa, V. Kruger and S. Zhou, “Probabilistic recognition of human faces from video”. International Conference on Image Processing, pp.41-44, 2002.

[11] R. Chellappa, V. Kruger and S. Zhou, “Face recognition from video: a CONDENSATION approach”, IEEE International Conference on Automatic Face and Gesture Recognition, pp. 221-226, 2002.

[12] G. Aggarwal, A. K. Roy-Chowdhury, and R. Chellappa, “A System Identification Approach for Video-based Face Recognition”, IEEE International Conference on Pattern Recognition, Vol. 4, pp. 175-178, 2004.

[13] N. Faggian, A. Paplinski and T. J. Chin, “Face Recognition From Video using Active Appearance Model Segmentation”, IEEE International Conference on Pattern Recognition, pp.287-290, 2006.

[14] U. Park, A. K. Jain and A. Ross, “Face Recognition in Video: Adaptive Fusion of Multiple Matchers”, IEEE Conference on Computer Vision and Pattern Recognition, pp. 17-22, 2007.

[15] I. K. Fodor, “A survey of dimension reduction techniques”, Technical Report, UCRL-ID-148494, 2002.

[16] S. Dasgupta, “Experiments with random projection”, Uncertainty in Artificial Intelligence, pp.143-151, 2000.

[17] D. Achlioptas, “Database-friendly random projections”, ACM Symposium on the Principles of Database Systems, pp. 274–281, 2001.

[18] A. Magen, “Dimensionality reductions that preserve volumes and distance to affine spaces, and their algorithmic applications”, International Workshop on Randomization and Approximation Techniques in Computer Science, pp. 239-253, 2002.

[19] S. Kaski, “Dimensionality reduction by random mapping: Fast similarity computation for clustering”, IEEE International Joint Conference on Neural Networks, pp. 413-418, 1998.

[20] D. Fradkin and D. Madigan, “Experiments with random projection for machine learning”, ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 517-522, 2003.

[21] X. Z. Fern and C. E. Brodley, “Random projection for high dimensional data clustering: A cluster ensemble approach”, International Conference on Machine Learning, pp. 186-193, 2003.

[22] N. Goel, G. M. Bebis and A. V. Nefian, “Face recognition experiments with random projections”, SPIE Conference on Biometric Technology for Human Identification, pp. 426-437, 2005.

[23] Y. Yang, J. Wright, Y. Ma and S. S. Sastry, “Feature Selection in Face Recognition: A Sparse Representation Perspective”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 31 (2), pp. 210-227, 2009.

[24] D. L. Donoho, “Compressed sensing”, IEEE Transactions on Information Theory, Vol. 52, (4), pp. 1289–1306, 2006.

[25] E. J. Candès, J. Romberg and T. Tao, “Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information”, IEEE Transactions on Information Theory, Vol. 52 (2), pp. 489–509, 2006.

[26] R. Saab. and O. Yilmaz, “Sparse recovery by non-convex optimization - instance optimality”, preprint.

[27] T. T. Y. Lin and F. J. Herrmann, “Compressed wavefield extrapolation”, Geophysics, Vol. 72 (5), pp. SM77-SM93, 2007.

[28] J. Haupt, R. Castro, R. Nowak, G. Fudge, and A. Yeh, “Compressive sampling for signal classification,” Asilomar Conf. Signals, Systems and Computers, Pacific Grove, pp. 1430-1434, 2006.

[29] A. Majumdar and R. K. Ward, “Classification via Group Sparsity Promoting Regularization”, International Conference on Acoustics, Speech, and Signal Processing 2009 (accepted).

[30] E. van den Berg, M. Schmidt. M. Friedlander, K. Murphy, “Group sparsity via linear-time projection”, Technical Report TR-2008-09, Department of Computer Science, University of British Columbia, June 2008.

[31] J. A. Tropp and A. C. Gilbert, “Signal recovery from random measurements via Orthogonal Matching Pursuit”, IEEE Trans. Info. Theory, Vol. 53 (12), pp. 4655-4666, 2007.

[32] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features”, IEEE Conference on Computer Vision and Pattern Recognition, pp. 511-518., 2001.

[33] G. Hennenfent and F. J. Herrmann, „Simply denoise: wavefield reconstruction via jittered undersampling”, Geophysics, Vol. 73(3), pp. V19-V28, 2008.

[34] R. Goh, L. Liu, X. Liu and T. Chen, „The CMU Face In Action (FIA) Database”, IEEE International Conference of Computer Vision, pp. 225-263, 2005.

CHAPTER 7: CONCLUSION

7.1 Introduction

This thesis proposes a new approach for automated face recognition. The approach is specially suited for practical face recognition systems in dynamic environments (situations where new data is added regularly). Given the difficulty of the problem, commendable recognition accuracy has been achieved by this approach.

It can be argued that the recognition accuracy is still not ‘perfect’ – if perfection is defined as ‘100%’ recognition rate. But that ‘perfect’ recognition rate can not be achievable by humans as well. In recent studies [1-3] it was found that human recognition accuracy was vastly over rated. The following points were observed:

- Human beings are only good at recognizing familiar faces from different views (profile, frontal, tilted etc.)
- For faces with illumination variation, automatic recognition algorithms surpass human recognition rate.
- Human beings are not consistent in their recognition results.

In situations where our proposed approach is to be deployed, human beings are not suitable for face recognition because of the large scale of the problem. Automated face recognition is the only possibility. It has been shown in the thesis that the proposed methods fare better over state-of-the-art methods in computer-based face recognition.

As mentioned in Chapter 1, the three different approaches towards feature extraction for face recognition are global, local and hybrid. Although there are many proponents of local and hybrid methods, in a seminal work [4] it was pointed out that contrary to popular belief, human vision is primarily global; that is we do not remember parts of the face and assimilate this information to recognize a person, rather we remember the person’s face as a whole. This thesis is based on a global feature extraction scheme – random projection; the entire face image is projected onto a lower dimensional random subspace and classification is carried out on this subspace.

Random projections are also intimately connected with human vision. The vision system has two phases – i) acquisition, and ii) learning. The data acquisition phase is a better understood topic of the two. An early work [5] finds that the human visual system employs random projections for acquiring the data, thereby reducing the aliasing effects during reconstruction. This phenomenon is emulated in computer simulations in [6]. A recent work in image demosaicing [7] is based on the human visual model and uses random projections as well.

The aim of this thesis is to emulate human vision for recognition of human faces. Most studies treating this problem consist of three phases – data acquisition, feature extraction and classification/recognition. This thesis follows the human visual system more closely and has exactly the two phases (acquisition and learning) mentioned above. In the experiments the data acquisition via random projections was simulated. In section 7.4, as a course of future work, we will discuss how such data can be acquired from the real world.

7.2 Summary of Contribution

This work proposes a combined data acquisition and feature extraction scheme which was simulated by randomly projecting the already acquired data onto a lower dimensional space. Once the random projections of the face images are obtained they need to be classified for recognition. The classification results need to be consistent, i.e. the results should be approximately the same no matter which random projection matrix is used for dimensionality reduction. Of the known traditional classifiers the Nearest Neighbour (NN) classifier and the Support Vector Machine (SVM) are known to be robust to dimensionality reduction by random projections. SVM is however not a suitable classifier in our scenario, because it suffers from the problem of over-fitting when limited number of training samples are available and leads to poor generalization ability during testing. Therefore the only traditional classifier which can be considered in the present context is the NN. However the classification accuracy from NN needs improvement. This motivated us to propose a class of new classifiers which is consistent to dimensionality reduction via random projections and at the same time provide higher recognition rate compared to NN.

The Sparse Classifier (SC) [9] was the first work that proposed an improvement over NN for randomly projected face images. The recognition accuracy from SC was about 17% better than that of NN. The work in [9] was an experimental study and did not provide any theoretical proof regarding the robustness of the classifier; besides it had two shortcomings. The major shortcoming was regarding the optimization function employed during classification. The other issue was related to the speed of classification.

The classification assumption made in [9] did not reflect in the optimization function. We corrected this issue in Chapter 2 [11] by proposing an alternate optimization function which followed the assumption accurately. Consequently, the recognition accuracy was improved from that of [9] by about 3%. The classifier proposed in Chapter 2 is named Group Sparse Classifier (GSC).

GSC is however fraught with the same problem as SC in terms of speed. Both of them solve an optimization problem during the classification stage which is time consuming. The limitation in speed was overcome by replacing the optimization problem by fast greedy (approximate) algorithms in Chapter 3 [12]. The classifiers proposed in Chapter 3 are named the Fast Group Sparse Classifiers (FGSC). The FGSC classifiers are faster than GSC by about two orders of magnitude but at the cost of 0.5% reduction in recognition accuracy.

In our related work [13] the Nearest Subspace Classifier (NSC) is proposed as discussed in Chapter 4. The basis of NSC is a simplified form of the assumption made in Chapter 2 and 3. This slight simplification of the classification assumption led to a great reduction in the computational complexity during optimization. Consequently the NSC is faster than its peers (SC, FSC, GSC and FGSC).

Chapters 2-4 propose three new classifiers (GSC, FGSC and NSC). In Chapter 5, the robustness of these classifiers (along with that of SC) to dimensionality reduction via random projection is theoretically proved. The different classifiers are also compared on a common platform (Yale B and USPS databases).

Chapter 6 shows how the ideas of dimensionality reduction via random projection can be extended to address the problem of video-based face recognition. Our method gives considerably better result (99.3% accuracy) than comparable algorithms (98% or less).

7.3 Discussion

The objective of this thesis is to search for an alternate face recognition method than those provided by traditional machine learning tools. Conventional machine learning solutions to dimensionality reduction and classification require all the data to be present beforehand, i.e. whenever new data is added, the system can not be updated in online fashion, rather all the calculations need to be re-done from scratch. This creates a computational bottleneck for large scale implementation of face recognition systems.

The face recognition community has started to appreciate this problem in the recent past and there have been some studies that modified the existing dimensionality reduction methods for online training [14, 15]. The classifier employed along with such online dimensionality reduction methods has been the traditional Nearest Neighbour.

One way to reduce the data-dependency of the dimensionality reduction step is to find the high-to-low dimensional projection matrix for a large sample and use this projection function for reducing the dimensionality of new data. In [16], the Fisher Linear Discriminant projection matrix was computed for a particular dataset, and this projection matrix was used to reduce the dimensionality of new samples. Thus the dependency of the projection function on new data was removed. NN classification was used in [16]. In place of NN, one may use the classifiers proposed in this work. It remains to be seen how such a dimensionality reduction method performs. However, it will not be possible to prove the robustness of such an ad hoc dimensionality reduction scheme without making strong assumptions on the distribution of the data.

This work addresses the aforesaid problem from a completely different perspective. It is based on recent theoretical breakthroughs in signal processing [17, 18]. The non-adaptive dimensionality reduction method proposed in this thesis is computationally more efficient than modified adaptive methods such as [14, 15]. This work proposed several classifiers in Chapters 2-5 which we proved to be robust to the proposed dimensionality reduction via random projections.

7.4 Future Work

Although the work is nearly complete, there are two areas to be studied in the future. The first is on the theoretical front. In Chapter 5, the approximate robustness of the compressive classifiers to dimensionality reduction via random projections is theoretically proven. In the future we would like to work on theoretically quantifying the bounds arising out of the approximations. The other area we intend to concentrate is on is the data acquisition problem. This work simulated the data acquisition process by taking random projections of the images. Recently practical hardware has been built to acquire the projections directly. The Single Pixel camera [8] is such a device.

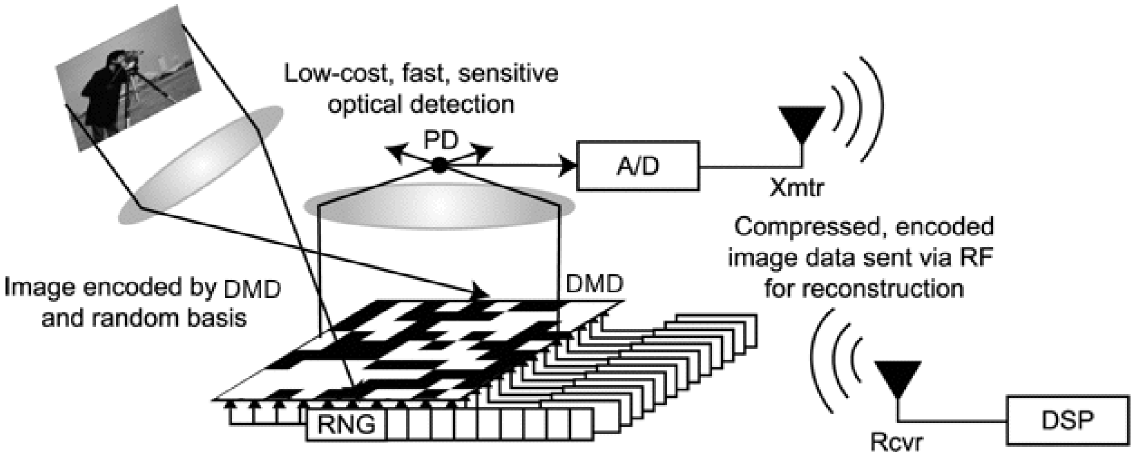

Figure 7-1: Single Pixel Camera

The prototype single pixel camera was developed for experiments on signal reconstructions from random projections. The schematic view of the camera is shown in Figure 7-1. The scene is projected (via a lens) onto a digital micromirror device (DMD). The flipping of the mirrors in the DMD, is controlled by a pseudorandom number generator (RNG). At each instant some of the mirrors are directed towards the projector lens and some of the mirrors are directed away from it.

There is a single optical sensor (PD) at the focal point of the lens. This optical sensor sends the acquired information for further processing.

The objective of the camera is to obtain a random projection of the original scene ‘*x*’ of the form

$$y = Ax$$

where A is a random projection matrix.

The projected data (*y*) is acquired sequentially. At each instant the pseudorandom number generator outputs a binary pattern which randomly flips the mirrors. The projection of the scene onto the flipped mirrors is optically added at the photo sensor (PD) to give one element of the vector ‘*y*’. At the next instant, the RNG generates a new random binary pattern and the next element of the vector ‘*y*’ is obtained. This process is repeated until a suitable number of random projections is obtained.

The camera was built for signal reconstruction applications; Figure 7-1 shows a DSP block at the end. Generally DSP block contains algorithms for image reconstruction. We intend to have a DSP block comprising of Compressive Classification algorithms so that there is an integrated system for classification.

7.5 References

[1] A. J. O'Toole, P. J. Phillips, J. Fang, J. Ayyad, N. Penard and H. Abdi, H "Face Recognition Algorithms Surpass Humans Matching Faces Over Changes in Illumination," IEEE Transactions on Pattern Analysis and Machine Intelligence, , Vol.29 (9), pp.1642-1646, 2007.

[2] A. Adler, and M. E. Schuckers, "Comparing Human and Automatic Face Recognition Performance," IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, , Vol.37 (5), pp.1248-1255, 2007

[3] M. Meytlis and L. Sirovich, "On the Dimensionality of Face Space," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 29 (7), pp. 1262-1267, 2007.

[4] P. Sinha, B. Balas, Y. Ostrovsky, R. Russell, “Face Recognition by Humans: 19 Results All Computer Vision Researchers Should Know About”, Proceedings of the IEEE, Vol. 94 (11), pp. 1948-1962, 2006.

[5] J.I. Yellott Jr., “Spectral analysis of spatial sampling by photoreceptors: Topological disorder prevents aliasing”, Vision Research, Vol. 22, pp. 1205-1210, 1982.

[6] M. A. Z. Dippe and E. H. Wold, “Antialiasing Through Stochastic Sampling”, ACM SIGGRAPH, Vol. 19 (3), pp. 69-78, 1985.

[7] D. Alleysson, S. Süsstrunk, and J. Hérault, “Linear Demosaicing inspired by the Human Visual System”, IEEE Transactions on Image Processing, Vol. 14(4), pp.1-10, 2005.

[8] http://dsp.rice.edu/cscamera

[9] Y. Yang, J. Wright, Y. Ma and S. S. Sastry, “Feature Selection in Face Recognition: A Sparse Representation Perspective”, IEEE Trans. PAMI, Vol. 31 (2), pp. 210-227, 2009.

[10] A. Majumdar and Rabab K. Ward, “Compressive Classification”, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics (submitted).

[11] A. Majumdar and R. K. Ward, “Classification via Group Sparsity Promoting Regularization”, IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 873-876, 2009.

[12] A. Majumdar and R. K. Ward, “Fast Group Sparse Classifier” IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, Victoria, Canada, August 2009 (accepted).

[13] A. Majumdar and R. K. Ward, “Nearest Subspace Classifier” IEEE International Conference on Image Processing, Cairo, Egypt, November 2009 (submitted).

[14] T. J. Chin and D. Suter, “Incremental Kernel Principal Component Analysis”, IEEE Transactions on Image Processing, Vol. 16, (6), pp. 1662-1674, 2007.

[15] H. Zhao and P. C. Yuen, “Incremental Linear Discriminant Analysis for Face Recognition”, IEEE Trans. on Systems, Man, And Cybernetics—Part B: Cybernetics, Vol. 38 (1), pp.210-221, 2008.

[16] A. Majumdar and R. K. Ward, "Pseudo-Fisherface Method for Single Image per Person Face Recognition", IEEE International Conference on Acoustics, Speech, and Signal Processing, pg. 989-992, 2008.

[17] D. L. Donoho, “Compressed sensing,” IEEE Transactions on Information Theory, Vol. 52 (4), pp. 1289–1306, 2006.

[18] E. J. Cand`es and T. Tao, “Near optimal signal recovery from random projections: Universal encoding strategies?”, IEEE Transactions on Information Theory, Vol. 52 (12), pp. 5406–5425, 2006.