



DEEP DICTIONARY LEARNING

by

VANIKA SINGHAL

Under the Supervision of Dr. Angshul Majumdar

Indraprastha Institute of Information Technology Delhi

August, 2019

©Indraprastha Institute of Information Technology (IIITD), New Delhi, 2019



DEEP DICTIONARY LEARNING

by

VANIKA SINGHAL

Submitted

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

to the

Indraprastha Institute of Information Technology Delhi

August, 2019

Certificate

This is to certify that the thesis titled *Deep Dictionary Learning* being submitted by *Vanika Singhal* to the Indraprastha Institute of Information Technology Delhi, for the award of the degree of Doctor of Philosophy, is an original research work carried out by her under my supervision. In my opinion, the thesis has reached the standard fulfilling the requirements of the regulations relating to the degree.

The results contained in this thesis have not been submitted in part or full to any other university or institute for the award of any degree or diploma.

August, 2019

Dr. Angshul Majumdar

Indraprastha Institute of Information Technology Delhi

New Delhi 110020

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisor Dr. Angshul Majumdar for the continuous support of my Ph.D study and related research, for his patience, motivation, and immense knowledge. His guidance helped me throughout the time of research and writing of this dissertation. I would like to thank him for providing excellent research facilities.

I would like to thank my committee members Dr. Debarka Sengupta and Dr. Saket Anand for their insightful comments. I am grateful to Indraprastha Institute of Information Technology for providing excellent infrastructure and research environment.

I thank my fellow colleagues and labmates namely Shikha Singh, Megha Gupta Gaur, Jyoti Maggu, Aanchal Mongiya, Divya Sitani, Shalini Sharma and Niharika Aggarwal for their help and support. I gratefully acknowledge their help.

I would like to thank my family, especially my mother for always believing in me, for their continuous love and supporting me throughout my life. I owe a special thanks to my husband, Hemant Goyal for supporting me and encouraging me throughout my Ph.D. To my beloved son Aarav, I would like to express my thanks for always cheering me up.

Abstract

Currently there are three basic frameworks in deep learning - stacked autoencoders (SAE), deep belief network (DBN) and convolutional neural network (CNN); SAE and DBN can be applied to arbitrary inputs but CNN can only be applied to natural signals having local correlations (speech, image, ECG, EEG etc.). I am working on developing a new framework for deep learning - deep dictionary learning (DDL). Just as SAE uses autoencoders as basic units and DBN uses restricted Boltzmann machines, DDL uses dictionaries as the basic unit. In lay man's terms, DDL is formed by stacking one dictionary after another such that the output (features) from the shallower layer feeds into the next (deeper) layer as input.

The initial work on DDL was a greedy sub-optimal solution, i.e. each of the layers were solved separately. My first work has been on proposing an optimal solution to jointly learn all the layers. This was a solution for unsupervised feature extraction using DDL. Later I worked on a supervised version of deep dictionary learning with a plug-and-play approach. The supervised version is general enough to perform classification, multi-label classification and regression. Proposed supervised deep dictionary learning for multi-label classification has been used for solving a practical problem of Non-Intrusive Load Monitoring (NILM). We also proposed a technique called deep blind compressed sensing which combines the analytic power of deep learning with reconstruction ability of compressed sensing. The objective of this work was to classify biomedical signals from their compressive measures.

Next, we work on Siamese DDL networks. These are usually required for verification problems in biometrics. We applied them on face verification problem in disguise detection and kinship verification. It can also be applied in a variety of other situations. For example in estimating dense depth from an image and sparse depth with a complete learning based approach. Or in problems arising

in BP estimation from multiple sources (multi-channel ECG and PPG).

Finally, we worked on deeply coupled dictionary learning. These networks are used to generate a linear mapping between samples of different domains. Existing approaches learn a linear map between the source and target domains. We propose to use deep dictionary learning to solve for complex mapping. Such networks can be used in applications like image denoising, image super-resolution, image reconstruction etc.

Contents

Acknowledgements

Abstract	i
-----------------	----------

List of Tables	vii
-----------------------	------------

List of Figures	x
------------------------	----------

0.1	Introduction	2
0.1.1	Autoencoder	4
0.1.2	Restricted Boltzman Machine	6
0.1.3	Convolutional Neural Network	7
0.2	Shortcomings of Traditional Deep Learning	8
1	Deep Dictionary Learning - Classification and Clustering Problems	11
1.1	Literature Review	12
1.2	Proposed Formulation	15
1.2.1	Training	15
1.2.2	Testing	18
1.2.3	Computational Complexity	19
1.3	Experimental Evaluation	20
1.3.1	Classification	20

1.3.2	Clustering	22
1.4	Discussion	24
2	Deep Dictionary Learning - Inverse Problems	25
2.1	Literature Review	26
2.1.1	Compressed Sensing based methods	26
2.1.2	Neural Network based Techniques	28
2.2	Proposed Formulation	30
2.3	Experiments and Results	33
2.3.1	Denoising	33
2.3.2	Super-resolution	37
2.3.3	Reconstruction	39
2.4	Discussion	41
3	Supervised Deep Dictionary Learning	42
3.1	Literature Review	43
3.2	Proposed Formulation	46
3.2.1	Training	47
3.2.2	Testing	49
3.3	Experiments and Results	51
3.3.1	Single Label Classification	51
3.3.2	Multilabel Classification	55
3.3.3	Simultaneous Detection of Multiple Appliances from Smart-Meter Measurements via Multi-label Consistent Deep Dictionary Learning	58
3.4	Discussion	64

4	Semi Supervised Deep Blind Compressed Sensing	66
4.1	Literature Review	67
4.1.1	Blind Compressed Sensing	67
4.2	Proposed Formulation	69
4.2.1	Reconstruction Ability	73
4.2.2	Contributions	73
4.3	Experimental Results	74
4.3.1	Seizure Detection from EEG	74
4.3.2	Arrhythmia Classification from ECG	80
4.4	Discussion	85
5	Siamese Deep Dictionary Learning	87
5.1	Motivation	87
5.2	Proposed Formulation	88
5.2.1	Training	91
5.2.2	Testing	95
5.3	Experiments and Results	97
5.3.1	Results on IIIT-D Disguise Face Database	100
5.3.2	Results on CrowdFace Database	101
5.3.3	Results on Kinship Databases	102
5.4	Discussion	104
6	Coupled Deep Dictionary Learning	105
6.1	Introduction	106
6.2	Literature Review	108
6.2.1	Coupled Representation Learning	108

6.3	Proposed Formulation	111
6.3.1	Training	112
6.3.2	Testing	115
6.4	Experiments and Results	116
6.4.1	Denoising	117
6.4.2	Deblurring	119
6.4.3	Super-resolution	122
6.4.4	Reconstruction	123
6.5	Discussion	126
7	Conclusions	127
7.1	Summary of Contribution	128
7.1.1	Chapter 1 : Deep Dictionary Learning - Classification and Clustering Problem	128
7.1.2	Chapter 2 : Deep Dictionary Learning - Inverse Problem	129
7.1.3	Chapter 3 : Supervised Deep Dictionary Learning	129
7.1.4	Chapter 4 : Semi-Supervised Deep Blind Compressed Sensing	130
7.1.5	Chapter 5 : Siamese Deep Dictionary Learning	130
7.1.6	Chapter 6 : Coupled Deep Dictionary Learning	131
7.2	Future Work	132
7.2.1	Residual Deep Dictionary Learning	132
7.2.2	Recurrent Deep Dictionary Learning	133
	References	135

List of Tables

1.1	Comparison on KNN	21
1.2	Comparison on SRC	22
1.3	Comparison on SVM	22
1.4	Clustering Results on COIL20	23
1.5	Clustering Results on YaleB	23
2.1	Lena-Effect of Patchsize and Layers	34
2.2	Peppers-Effect of Patchsize and Layers	35
2.3	Comparative Denoising Performance ($\sigma = 10$)	35
2.4	Comparative Denoising Performance ($\sigma = 50$)	36
2.5	Comparative Super-resolution Performance (Magnification = 2)	38
2.6	Comparative Super-resolution Performance (Magnification = 4)	38
2.7	Reconstruction Accuracy in SNR (dB)	40
3.1	Classification Accuracy on Face Recongntion	54
3.2	Classification Accuracy on Scene and Yeast datasets	57
3.3	Performance Evaluation on REDD dataset	63
3.4	Performance Evaluation on Pecan street dataset	63
3.5	Appliance Level Evaluation on REDD Dataset	64
3.6	Appliance Level Evaluation on Pecan Dataset	64

4.1	Effect of Depth at 4:1 Compression on Class-wise Accuracy . . .	76
4.2	Effect of Depth at 2:1 Compression on Class-wise Accuracy . . .	76
4.3	EEG Classification Results (%age) at 4:1 Compression	77
4.4	EEG Classification Results (%age) at 2:1 Compression	77
4.5	EEG Reconstruction Results	79
4.6	Train and Test Set Details	82
4.7	ECG Classification Results (%age) on 4:1 Compression	83
4.8	ECG Classification Results (%age) on 2:1 Compression	83
4.9	ECG Classification Results (%age) on 4:1 Compression	83
4.10	ECG Classification Results (%age) on 2:1 Compression	84
4.11	ECG Reconstruction Results	84
5.1	Verification performance of the proposed and existing algorithms on the ID V1 database [1, 2]. The results are reported in terms of genuine accept rates at different false accept rates.	101
5.2	Face verification performance of commercial systems, VGG- Face and the proposed algorithm on the CrowdFace database [3]. The results are reported in terms of genuine accept rate at different false accept rates.	102
5.3	Kinship verification accuracy (%) of the existing and proposed kinship verification algorithms on kinship datasets. The results are reported on pre-existing benchmark protocols of individual databases. The results of existing algorithms are reported from the original papers. Top-2 results are highlighted which shows that the proposed algorithm yields comparable results relative to state of the art algorithms.	104
6.1	Denoising Results	117
6.2	Deblurring Results	119
6.3	Super-resolution Results	122

6.4 Reconstruction Results 124

List of Figures

1	(a)Neural Network (b)Segregated Representation	2
2	Why Deep networks?	4
3	(a) Autoencoder (b) Stacked Autoencoder	5
4	(a) Restricted Boltzman Machine (b) Deep Belief Network . . .	6
5	Lenet-5 Architecture	8
1.1	(a) Dictionary Learning (b) Deep Dictionary Learning	12
2.1	Stacked Autoencoder	29
2.2	Top. Denoising Difference Image of Brain MRI. Bottom. De- noising Hyperspectral Image (band 50) WDC. Left to Right- KSVD, Transform Learning, SSDA, BM3D,DnCNN and Pro- posed	37
2.3	Results with 4 times magnification. Left to Right - Original, CDL, SRCNN and Proposed 1 layer	39
2.4	Reconstructed Images. Top- Ex Vivo, Bottom- In Vivo. Left to Right- Original,TL,DL,Proposed 2 layer and Proposed 3 layer .	40
2.5	Difference Images. Top- Ex Vivo, Bottom- In Vivo. Left to Right- TL, DL, Proposed 2 layer and Proposed 3 layer	41
3.1	Sample Images from YaleB Dataset	51
3.2	Sample Images from AR Dataset	52

4.1	Confusion Matrices for EEG. Top - For 4:1 Compression. Bottom - For 2:1 Compression. For every matrix, the columns are the actual classes and the rows are the predicted classes	78
4.2	EEG reconstruction. (a) CS [4] Reconstruction. (b) Proposed. . .	79
4.3	ECG reconstruction. (a) CS [4] Reconstruction. (b) Proposed. . .	85
5.1	Typical Siamese Neural Network	88
5.2	Siamease Deep Dictionary Learning	89
5.3	Example Images from IIIT-D Disguise Face Dataset (ID V1) . .	98
5.4	Example Images from KinFaceW-I and KinFaceW-II datasets . .	99
5.5	ROC curve on the IIIT Delhi disguise face database.	100
5.6	ROC curve on the CrowdFace database.	103
6.1	Coupled representation Learning	109
6.2	Barbara- Denoising Results. a)Ground Truth b)Noisy c)Transform d)BM3D e)DnCNN f) Proposed	118
6.3	Man - Deblurring Results. a) Ground Truth b) Blurred image c) SPB d) VCNN e) CDAE f) Proposed	120
6.4	Baboon - Super-resolution Results. a) Ground Truth b) Upsampled c) CTL d) SRCNN e) SRN f) CDAE g) Proposed	121
6.5	Foreman - Reconstruction Results. a) Ground Truth b) RODEO c) ReconNet d) Transform e) Proposed	124
7.1	Residual Dictionary Learning	132
7.2	Dense Dictionary Learning	133
7.3	Recurrent Dictionary Learning	134

Introduction

0.1 Introduction

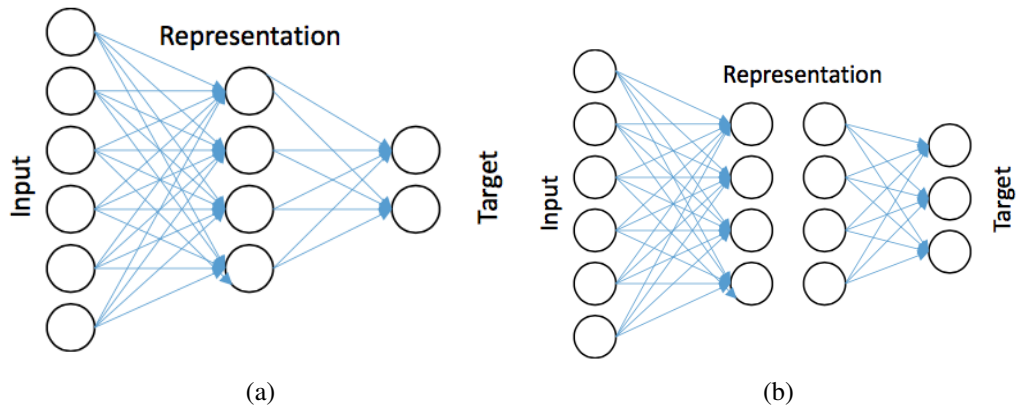


Figure 1: (a)Neural Network (b)Segregated Representation

In typical neural network there is an input layer where samples are presented, a hidden or representation layer and an output layer with the target values (Figure 1a). The network is learned using back-propagation.

As shown in Figure 1b learning of such network can be perceived as two sub-problems: learning the weights between input and the representation layer and between representation and output layer. If the representations are known, learning the weights between hidden and output layer is trivial. Its a simple regression problem since both the input (representations of the input data) and

output are known. Learning the weights between input and representation layer is a challenging task, since both weights and output are unknown. This is called representation learning.

There are three approaches for representation learning: autoencoder [5], restricted boltzman machine (RBM) [6] and convolutional neural network. Training single layer neural network is a simple task; RBM and autoencoder are not used to solve such shallow networks.

Recurrent Neural Networks (RNNs) [7] are designed to handle time series data. RNNs have shown significant performance in applications like image captioning and machine translation. In this thesis we are not working on time series data hence, we are not discussing it any further.

Deep neural network have multiple representation layers. Training such networks directly is a difficult task. Representation learning techniques like autoencoder and RBM are used to train deep networks. Deep networks are formed by cascading multiple layers of basic building blocks.

Why Deep networks?

Traditional machine learning methods are limited in their ability to work with raw data. In order to perform any machine learning task careful feature engineering is used which in turn requires domain expertise. Using domain knowledge, a feature extractor is build to transform the raw data into a suitable internal representation. This internal representation is then fed to a learning subsystem, often a classifier to detect pattern in the input data [8].

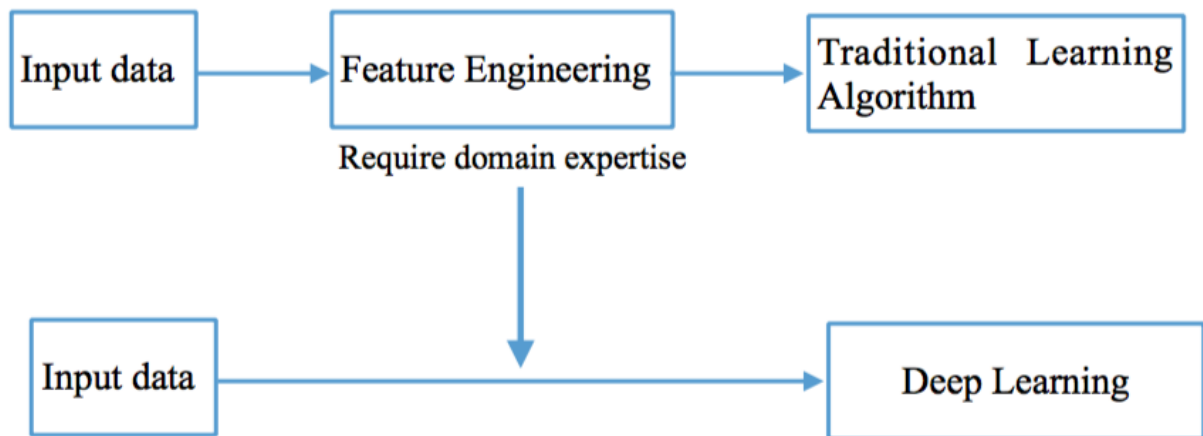


Figure 2: Why Deep networks?

Deep learning methods have multiple levels of learning, in each layer a more abstract representation of the raw data is learned via non-linear transformations. Using such non-linear modules of transformation, very complex functions can be learned. As shown in Figure 2, the key idea of deep learning is that the feature engineering is not required, the features are learned by the network itself from the raw data.

A deep network can be formed by using autoencoder and RBM as a basic building block. All three techniques are discussed briefly in the next subsections.

0.1.1 Autoencoder

The architecture of an autoencoder is shown in Figure 3a. Autoencoder is a self-supervised network, it learns encoding and decoding weights between input and itself. It consists of two parts encoder and decoder. Encoder maps the input to a latent representation and the decoder maps the encoded input back to the data.

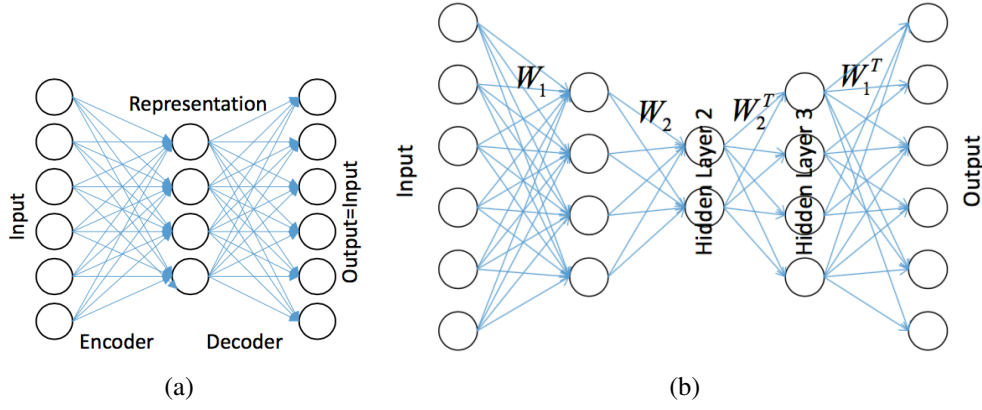


Figure 3: (a) Autoencoder (b) Stacked Autoencoder

For a given input x the representations can be expressed as:

$$z = \phi(Dx) \quad (1)$$

Where, ϕ is the non-linearity and D are the encoding weights. The decoder maps the representations back to the data space:

$$x = D' \phi(Dx) \quad (2)$$

The problem is to learn the encoding and decoding weights D and D' . These are learnt by minimizing the euclidean cost:

$$\min_{D, D'} \|x - D' \phi(Dx)\|_2^2 \quad (3)$$

Idea behind representation learning is to preserve information content of the input in the representations. In autoencoder, information is preserved at the representations in euclidean sense. To form a neural network, decoder part of the autoencoder is removed. The encoder forms the first part (input to representa-

tion layer) of neural network. Targets are attached to the representations and backpropagation is used to train the network.

Deep networks can be formed using autoencoders by nesting them one inside the another. These networks are called stacked autoencoders (Figure 3b). Once this network is learned the decoder part is removed and targets are attached to representations of deepest layer. And the network is fine tuned using backpropagation.

0.1.2 Restricted Boltzman Machine

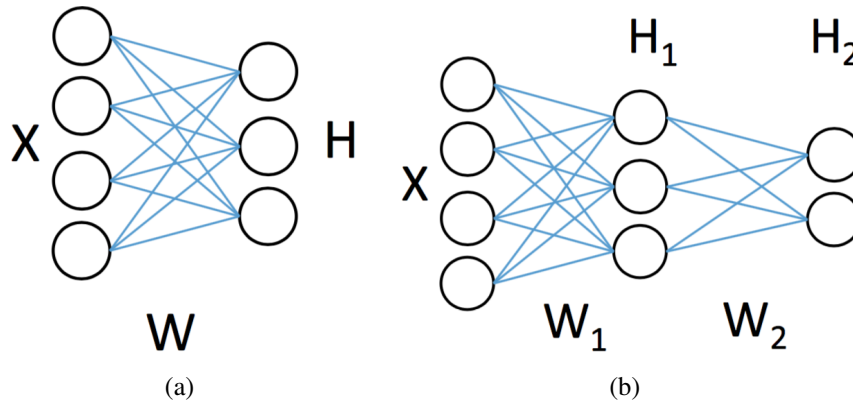


Figure 4: (a) Restricted Boltzman Machine (b) Deep Belief Network

Second approach to representation learning is the restricted boltzmann machine (Figure 4a). RBM learns by maximizing the similarity between the projection of the data and representations. As it has no output, backpropagation cannot be used for training. It is solved by contrastive divergence [6]. Once the RBM is learnt, targets are attached to its output to form a complete neural network.

Deep Belief Network [9] is formed by stacking one RBM unit after the other.

The architecture is shown in Figure 4b. The targets are attached to the final representation layer and network weights are tuned using backpropagation.

0.1.3 Convolutional Neural Network

Convolutional neural networks are special kind of multi-layer neural networks. Like other neural networks, CNN are trained with backpropagation but the architecture is different. CNN is a sequence of layers. Mainly there are three types of layers: convolutional layer, pooling layer, and fully connected layer. These layers are stacked one after the other to form a complete convolutional neural network architecture. Each layer of CNN convert one layer of activations to another through a differentiable function. CNN takes an input image, process it and classify it under certain categories. Each input pass through convolutional layers with filters, pooling layers, fully connected layer and a softmax function is applied for classification.

Figure 5 shows the diagram of Lenet-5 architecture introduced by Lecun et. al in 1998 [10]. It was designed to recognize handwritten and machine printed digits. Lenet-5 is 7 layer CNN, among which 3 are convolutional layers (C_1 , C_2 and C_3). Convolutional layer consists of 5 by 5 filters with stride 1. There are two 2 by 2 average pooling layers (S_2 and S_4). The layer F_6 denote the fully connected layer. Tanh activation function is used throughout the network.

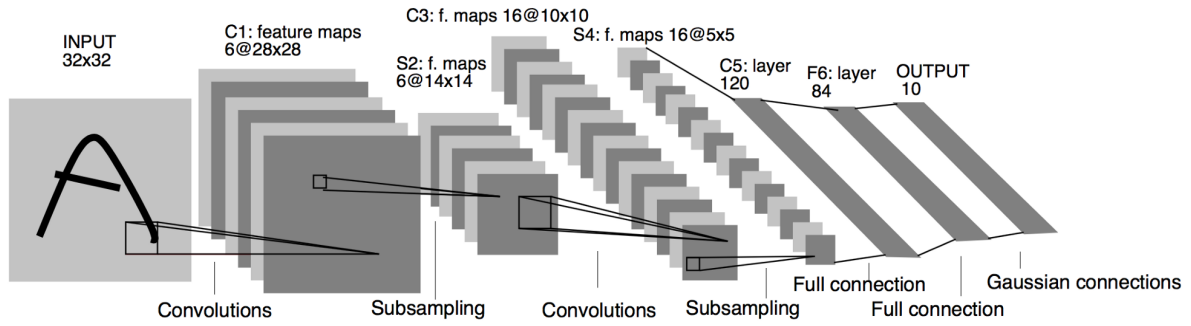


Figure 5: Lenet-5 Architecture

0.2 Shortcomings of Traditional Deep Learning

There are three pillars of deep learning - convolutional neural network (CNN), deep belief network (DBN) and stacked autoencoder (SAE). CNNs give excellent results in every perceivable image analysis task. However, the applicability of CNN is limited largely to images in the visible range. There is literature on application of CNNs to areas outside this range (hyperspectral imaging, radar imaging etc.) but the success has been limited. This is largely because CNNs need humongous volumes of supervised data to train; this is only available in the visible range of the spectrum. The pre-training fine-tuning paradigm does not generalize well outside this range; hence the applicability of CNN has been largely restricted in scientific imaging modalities. This also precludes their applicability in unsupervised tasks like clustering.

DBN uses the restricted Boltzmann machine (RBM) as the building block. Mathematically the cost function for RBM is cumbersome and cannot be solved efficiently; at best it can be solved only approximately. This is a serious impediment that precludes any serious mathematical modification to the learning

model. Besides, the inputs to RBM / DBN need to be either binary or in the range between 0 and 1 (Gaussian Bernoulli RBM); this is highly restrictive. For any signal where the dynamic range is high, when the input is normalized (between 0 and 1) the small numbers reach the limits of machine precision and are treated as zeroes or garbage values.

SAE overcomes the limitations of DBN; it has a mathematically tractable cost function and can handle any input value. However the problem of this model is that it requires learning twice the number of parameters / connections (encoders and decoders) than other neural networks. This makes the SAE susceptible to over-fitting.

From the shortcomings of existing deep learning frameworks, we understand that any new deep learning approach needs to satisfy three criteria -

1. **Mathematically efficient to solve**, i.e. should have a cost function amenable to modifications over time. For example should be able to handle changing signal models such as sparsity penalties of the past decade to current trends in graph signal processing. Neither CNNs nor DBNs satisfy this criteria.
2. **Should be amenable to unsupervised and semi-supervised learning.** This would help address problems like clustering and segmentation. Semi-supervised learning would allow handling tasks in scientific imaging modalities that are currently not well dealt with by CNN.
3. **Should learn optimal number connections.** The first two points are satisfied by SAE, but as mentioned before, they overfit owing to the requirement

of learning larger number of parameters / connections.

All the three demands are met by the developing field of deep dictionary learning.

Chapter 1

Deep Dictionary Learning - Classification and Clustering Problems

The concept of deep dictionary learning (DDL) is motivated by the success of deep learning techniques like stacked autoencoder, deep belief networks and convolutional neural networks. In DDL multiple levels of dictionaries are learned to represent the data in terms of coefficients. So far, the problem could only be solved in greedy fashion. In greedy learning one layer is learned at a time, the coefficients learned in the previous layer acted as the input to the subsequent layer. Input to the first layer is the training data. The problem with this approach is that there is feedback from shallower layers to the deeper layers but not the other way. Other deep learning tools also follow a greedy learning paradigm, but the issue of feedback from deeper layer to shallower layers is resolved during fine-tuning stage.

This chapter addresses this shortcoming of greedy learning by proposing a solution to DDL where all the dictionaries and coefficients are solved simul-

taneously. Section 1.1 presents literature review. In section 1.2 the proposed algorithm is discussed in details followed by experimental evaluation.

1.1 Literature Review

Standard interpretation of dictionary learning is shown in Figure 1.1a. In DDL, the idea is to learn multiple levels of dictionaries. DDL proposes to extend the shallow dictionary learning into multiple layers leading to deep dictionary learning, see Figure 1.1b. Greedy approach to deep dictionary learning learns

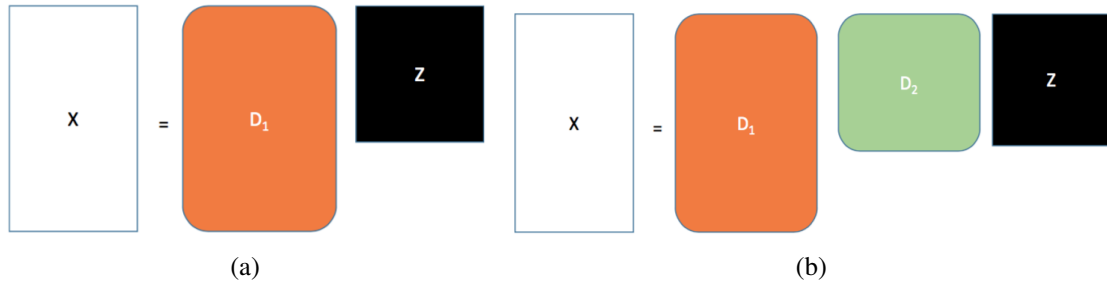


Figure 1.1: (a) Dictionary Learning (b) Deep Dictionary Learning

one layer at a time [11]. First layer of the dictionary learns from the training data. The representations and dictionaries are learned for the first layer. In the second layer, representations learned in the first layer act as the input and it learns the dictionary for the second layer. Same concept can be extended to multiple layers. Formulation for deep dictionary learning with non-linear activations can be expressed as:

$$X = D_1 \phi(D_2 \phi(\dots \phi(D_N Z))) \quad (1.1)$$

Where, X is input data, D_1 to D_N are N level dictionaries and Z are the final level representations. ϕ is an activation function. ϕ is usually a tanh or sigmoid function. This is an unsupervised formulation since there are no labels. Following optimization problem need to be solved.

$$\min_{D_1, D_2 \dots D_N, Z} \|X - D_1 \phi(D_2 \phi(\dots \phi(D_N Z)))\|_F^2 \quad (1.2)$$

In greedy approach one layer is learned at a time. For the first layer of dictionary we can express $Z_1 = \phi(D_2 \phi(\dots \phi(D_N Z)))$. Therefore, greedy learning of the first layer is given by equation 1.3.

$$\min_{D_1, Z_1} \|X - D_1 Z_1\|_F^2 \quad (1.3)$$

This is a basic dictionary learning problem and can be solved as a single layer dictionary learning problem. Dictionary D_1 and coefficients Z_1 are learned by:

$$Z_1 \leftarrow \min_{Z_1} \|X - D_1 Z_1\|_F^2 \quad (1.4a)$$

$$D_1 \leftarrow \min_{D_1} \|X - D_1 Z_1\|_F^2 \quad (1.4b)$$

This is method of optimal directions [12]. Both, equations 1.4a and 1.4b are simple least squares problems and have closed form solutions.

Similarly for the second layer, we substitute $Z_2 = \phi(D_3 \dots \phi(D_N Z))$ which gives $Z_1 = \phi(D_2 Z_2)$, or can be expressed as $\phi^{-1}(Z_1) = D_2 Z_2$. Inverse can be applied simply because activation function is applied term wise. Second layer of deep dictionary learning can be formulated as a single level dictionary

learning problem as given below:

$$\min_{D_2, Z_2} \|\phi^{-1}(Z_1) - D_2 Z_2\|_F^2 \quad (1.5)$$

This can be solved by alternate minimization as in the case of first layer 1.4a, 1.4b.

Continuing in the same manner till the last layer; one will substitute $Z_{N-1} = \phi(D_N Z)$ or $\phi^{-1}(Z_{N-1}) = D_N Z$. In the last layer coefficients Z can be sparse but its not necessary. For learning sparse features, l_1 norm is applied on the features of last layer. This is given by:

$$\min_{D_N, Z} \|\phi^{-1}(Z_{N-1}) - D_N Z\|_F^2 + \lambda \|Z\|_1 \quad (1.6)$$

This too is solved using alternate minimization.

$$Z \leftarrow \min_Z \|\phi^{-1}(Z_{N-1}) - D_N Z\|_F^2 + \lambda \|Z\|_1 \quad (1.7a)$$

$$D_N \leftarrow \min_{D_N} \|\phi^{-1}(Z_{N-1}) - D_N Z\|_F^2 \quad (1.7b)$$

As before, 1.7b has a closed form solution. 1.7a on the other hand can be solved using Iterative Soft Thresolding Algorithm (ISTA) [13]. This concludes the training stage.

During testing the learned dictionaries are used to generate the test coefficients. Given a test sample x_{test} one needs to solve,

$$\min_{z_{test}} \|x_{test} - D_1 \phi(D_2 \dots \phi(D_N z_{test}))\|_2^2 \quad (1.8)$$

Using the substitution $z_1 = \phi(D_2 \phi(\dots \phi(D_N z_{test})))$, learning the feature from

the first layer turns out to be,

$$\min_{z_1} \|x_{test} - D_1 z_1\|_2^2 \quad (1.9)$$

This has a simple analytic solution in the form of pseudoinverse. With the substitution $Z_2 = \phi(D_3 \dots \phi(D_N Z))$, one can generate the features at the second level by solving,

$$\min_{z_2} \|z_1 - \phi(D_2 z_2)\|_2^2 \equiv \min_{z_2} \|\phi^{-1}(z_1) - D_2 z_2\|_2^2 \quad (1.10)$$

The equivalent form has a closed form solution as well. Continuing in this fashion till the final layer, one has

$$\min_{z_{test}} \|z_{N-1} - \phi(D_2 z_{test})\|_2^2 \equiv \min_{z_{test}} \|\phi^{-1}(z_{N-1}) - D_2 z_{test}\|_2^2 \quad (1.11)$$

This completes the testing phase. Once the test features are obtained one can use any classifier to make predictions.

1.2 Proposed Formulation

1.2.1 Training

Greedy method for solving deep dictionary learning is sub-optimal since the shallower layers affect the deeper layers but not the other way around. We propose to solve 1.2 in an optimal manner where all the coefficients and dictionaries are updated in one go.

In order to solve the formulation in Equation 1.2 we use proximal variable splitting method [14]. For the first layer we substitute $Z_1 = \phi(D_2\phi(\dots\phi(D_N Z)))$ in 1.2. With this substitution the augmented lagrangian [15] is given by:

$$\min_{D_1, D_2, \dots, D_N, Z_1, Z} \|X - D_1 Z_1\|_F^2 + \mu_1 \|Z_1 - \phi(D_2\phi(\dots\phi(D_N Z)))\|_F^2 \quad (1.12)$$

Similarly for the second layer we substitute $Z_2 = \phi(D_3 \dots \phi(D_N Z))$. The augmented lagrangian now becomes:

$$\begin{aligned} \min_{D_1, D_2, \dots, D_N, Z_1, Z_2, Z} & \|X - D_1 Z_1\|_F^2 + \mu_1 \|Z_1 - \phi(D_2 Z_2)\|_F^2 + \\ & \mu_2 \|Z_2 - \phi(D_3 \dots \phi(D_N Z))\|_F^2 \end{aligned} \quad (1.13)$$

Continuing in this manner the final augmented lagrangian is given by:

$$\begin{aligned} \min_{D_1, D_2, \dots, D_N, Z_1, Z_2, \dots, Z_{N-1}, Z} & \|X - D_1 Z_1\|_F^2 + \mu_1 \|Z_1 - \phi(D_2 Z_2)\|_F^2 + \dots + \\ & \mu_{N-1} \|Z_{N-1} - \phi(D_N Z)\|_F^2 \end{aligned} \quad (1.14)$$

The multiplicative terms μ' s needs to be tuned in general. But in this case it determines the relative importance of each layer. Since there is no reason to give one layer more weightage over the other, we take their values as unity. Therefore the simplified expression is given by:

$$\begin{aligned} \min_{D_1, D_2, \dots, D_N, Z_1, Z_2, \dots, Z_{N-1}, Z} & \|X - D_1 Z_1\|_F^2 + \|Z_1 - \phi(D_2 Z_2)\|_F^2 + \dots + \\ & \|Z_{N-1} - \phi(D_N Z)\|_F^2 \end{aligned} \quad (1.15)$$

Equation 1.15 can be solved using alternating direction method of multipliers (ADMM) technique [16]. The idea is to solve for each variable separately while

keeping the others constant. This is a non convex problem, but each of the sub problems is convex having closed form solution. Theoretically, there are no convergence guarantees. This is an iterative algorithm, practically we stop the iteration when the cost converges. Applying ADMM 1.15 can be segregated into following sub problems:

$$P_1 : \min_{D_1} \|X - D_1 Z_1\|_F^2$$

$$P_2 : \min_{D_2} \|Z_1 - \phi(D_2 Z_2)\|_F^2 \equiv \min_{D_2} \|\phi^{-1}(Z_1) - D_2 Z_2\|_F^2$$

.

.

.

$$P_N : \min_{D_N} \|Z_{N-1} - \phi(D_N Z)\|_F^2 \equiv \min_{D_N} \|\phi^{-1}(Z)_{N-1} - D_N Z\|_F^2$$

$$P_{N+1} : \min_{Z_1} \|X - D_1 Z_1\|_F^2 + \|Z_1 - \phi(D_2 Z_2)\|_F^2$$

$$P_{N+2} : \min_{Z_2} \|Z_1 - \phi(D_2 Z_2)\|_F^2 + \|Z_2 - \phi(D_3 Z_3)\|_F^2 \equiv \min_{Z_2} \|\phi^{-1}(Z_1) - D_2 Z_2\|_F^2 + \|Z_2 - \phi(D_3 Z_3)\|_F^2$$

.

.

.

$$P_{2N} : \min_Z \|Z_{N-1} - \phi(D_N Z)\|_F^2 \equiv \min_Z \|\phi^{-1}(Z)_{N-1} - D_N Z\|_F^2$$

All these sub problems are simple least square problems having solution in the form of pseudo inverses. Since ϕ is a unitary operator it can be easily inverted. Such inversion has been applied successfully in previous work [17] on stacked autoencoder.

One may also want to generate sparse features. It can be done by applying

an l_1 norm constraint on coefficients of last layer Z . In that case the formulation is given by:

$$\min_{D_1, D_2, \dots, D_N, Z_1, Z_2, \dots, Z_{N-1}, Z} \|X - D_1 Z_1\|_F^2 + \|Z_1 - \phi(D_2 Z_2)\|_F^2 + \dots + \|Z_{N-1} - \phi(D_N Z)\|_F^2 + \lambda \|Z\|_1 \quad (1.16)$$

Here λ is a regularization parameter. Addition of l_1 norm only changes the update of Z .

$$\min_Z \|Z_{N-1} - \phi(D_N Z)\|_F^2 + \lambda \|Z\|_1 \quad (1.17)$$

This is an l_1 minimization problem and can be solved using iterative soft thresholding algorithm [13]. This completes the training of unsupervised DDL. Next subsection describes the testing part.

1.2.2 Testing

In the testing phase given a test sample x , the testing coefficients z needs to be generated. It require solving the optimization problem of the form:

$$\min_z \|x - D_1 \phi(D_2 \phi(\dots \phi(D_N z)))\|_2^2 \quad (1.18)$$

This can be solved using the variable splitting method as before. Firstly we substitute $z_1 = \phi(D_2 \phi(\dots \phi(D_N z)))$; then for second layer we substitute $z_2 = \phi(\dots \phi(D_N z))$ and so on untill the second last layer. The augmented lagrangian

with all these proxies is given by:

$$\min_{z_1 \dots z_{N-1}, z} \|x - D_1 z_1\|_2^2 + \|z_1 - \phi(D_2 z_2)\|_2^2 + \dots + \|z_{N-1} - \phi(D_N z)\|_2^2 \quad (1.19)$$

Here $D's$ are the dictionaries learned in the training phase. The problem can be seggregated into following sub problems:

$$S_1 : \min_{z_1} \|x - D_1 z_1\|_2^2$$

$$S_2 : \min_{z_2} \|z_1 - \phi(D_2 z_2)\|_2^2 \equiv \min_{z_2} \|\phi^{-1}(z_1) - D_2 z_2\|_2^2$$

.

.

.

$$S_N : \min_z \|z_{N-1} - \phi(D_N z)\|_2^2 \equiv \min_z \|\phi^{-1}(z)_{N-1} - D_N z\|_2^2$$

This is the derivation for obtaining the testing coefficients. Once the coefficients are obtained for training and testing samples, these are fed to any third party classifier to perform classification or clustering.

1.2.3 Computational Complexity

For deep dictionary learning, all the sub-problems require solving a least square problem having a pseudo-inverse. Both the training and testing time complexities of the proposed method are $O(n^2)$. For training, it is faster than most deep learning frameworks but for testing, it is slower than the most. This is because other deep learning tools have a test time complexity of $O(n)$ since they are non-iterative and require only matrix product.

1.3 Experimental Evaluation

1.3.1 Classification

Experiments are performed on following benchmark datasets:

- **MNIST Dataset [18]**

It consists of 28×28 images of handwritten digits ranging from 0 to 9. The dataset has 60000 training images and 10000 testing images. We also tested on variations of MNIST dataset. These datasets are challenging because they have fewer training samples 10000 and more testing samples 50000. There are five variations of MNIST dataset.

1. basic (smaller subset of MNIST)
2. basic-rot (images with random rotations)
3. bg-rand (images with uniformly distributed random noise in the background)
4. bg-img (images with random image background)
5. bg-img-rot (images with random image background with rotation)

- **20-newsgroup Dataset [19]**

The training set consists of 11269 samples and the test set contains 7505 examples. We have used 5000 most frequent words for the binary input features. We follow the same protocol as outlined in [20].

- **GTZAN [21]**

It is a music genre dataset. It contains 10000 three-second audio clips,

equally distributed among 10 musical genres: blues, classical, country, disco, hip-hop, pop, jazz, metal, reggae and rock. Each example in the set is represented by 592 Mel-Phon coefficient (MPC) features. These are a simplified formulation of the Mel-frequency cepstral coefficients (MFCCs) that are shown to yield better classification performance. Since there is no predefined standard split and fewer examples, we have used 10-fold cross validation, where each fold consisted of 9000 training examples and 1000 test examples.

In this work the training is completely unsupervised (no labels are used during training). We are comparing with unsupervised deep learning methods; stacked denoising autoencoders [22], K-sparse autoencoder [23], contractive autoencoder [24] and deep belief networks [25]. Since the goal of this work is to show that proposed approach to solve DDL is better than the existing greedy DDL, we are also comparing with this. Learned models are available for popular datasets used in this work. For DDL a three layer network is used where number of atoms are halved after each layer.

Dataset	SDAE	KSAE	CAE	DBN	Greedy DDL	Proposed
MNIST	97.33	96.90	92.83	97.05	97.75	97.91
basic	95.25	91.64	90.92	95.37	95.80	96.07
basic-rot	84.83	80.24	78.56	84.71	87.00	87.23
bg-rand	86.42	85.89	85.61	86.36	89.35	89.77
bg-img	77.16	76.84	78.51	77.16	81.00	81.09
bg-img-rot	55.21	50.27	47.10	50.47	57.77	58.40
20-newsgroup	70.84	71.22	71.08	70.09	70.48	71.64
GTZAN	83.31	82.91	82.67	80.99	83.31	83.89

Table 1.1: Comparison on KNN

Three classifiers are trained using the learned features in the deepest layer:

Dataset	SDAE	KSAE	CAE	DBN	Greedy DDL	Proposed
MNIST	98.33	97.91	87.91	88.43	97.99	98.33
basic	96.91	95.07	95.03	87.49	96.38	96.97
basic-rot	90.04	88.85	88.63	79.47	89.74	90.23
bg-rand	91.03	83.59	82.25	79.67	91.38	91.61
bg-img	84.14	84.12	85.68	75.09	84.11	84.67
bg-img-rot	62.46	58.06	54.01	49.68	62.86	63.27
20-newsgroup	70.49	71.90	71.08	71.02	71.41	72.43
GTZAN	83.37	84.09	82.70	81.21	84.72	85.71

Table 1.2: Comparison on SRC

Dataset	SDAE	KSAE	CAE	DBN	Greedy DDL	Proposed
MNIST	98.50	98.46	97.74	98.53	98.64	98.71
basic	96.96	97.02	96.61	97.07	97.28	97.53
basic-rot	89.43	88.75	72.54	89.05	90.34	90.75
bg-rand	91.28	90.07	85.20	89.59	92.38	92.62
bg-img	84.86	80.17	78.76	85.46	86.17	86.67
bg-img-rot	60.53	60.01	60.97	58.27	63.85	64.76
20-newsgroup	71.29	72.05	71.68	71.18	71.97	72.89
GTZAN	83.42	81.61	82.99	81.83	84.92	85.18

Table 1.3: Comparison on SVM

K-nearest neighbor (KNN), sparse representation based classifier (SRC) and support vector machine (SVM) with rbf kernel. Results are shown in Tables 1.1, 1.2 and 1.3 respectively. In the previous study [11], it has been shown that DDL performs better than other deep learning tools like SAE and DBN. It can be seen from the results that in most of the cases proposed approach not only outperform SAE and DBN, it also improves upon KSAE and CAE. It can also be noted that proposed approach consistently perform better than the greedy DDL.

1.3.2 Clustering

Experiments are performed on clustering problems as well. The experimental results have been compared with three studies; the first one being [11].

The other two are GraphEncoder [26] and deep subspace clustering [27]. We follow the experimental protocol found in [27].

In [27], experiments were carried out on the COIL20 (object recognition) and Extended YaleB (face recognition) datasets. For both the datasets DSIFT (dense scale invariant feature transform) and HOG (histogram of oriented gradients) features were extracted. They were further reduced by PCA to a dimensionality of 600. Since the groundtruths (class labels) for these datasets are available, clustering accuracy was measured in terms of NMI (normalized mutual information), ARI (adjusted rand index) and F-score. The results are shown in Table 1.4 (COIL20) and Table 1.5 (YaleB).

Method	DSIFT			HOG		
	NMI	ARI	F-Score	NMI	ARI	F-Score
GraphEncoder [26]	89.42	78.94	80.01	84.47	78.72	80.40
Deep subspace [27]	91.19	84.80	85.58	91.19	81.92	82.86
Greedy DDL [11]	91.04	84.60	83.54	90.12	80.20	81.30
Proposed	91.38	84.92	85.56	91.37	82.24	82.82

Table 1.4: Clustering Results on COIL20

A four tier architecture is used for experiments, 600 – 300 – 150 – 75 - this yielded the best results. K-means clustering is used on the features from the deepest level. Except the GraphEncoder, all others (including proposed DDL) use tanh activation function; the GraphEncoder uses sigmoid.

Method	DSIFT			HOG		
	NMI	ARI	F-Score	NMI	ARI	F-Score
GraphEncoder [26]	85.29	78.18	79.08	92.76	82.84	83.31
Deep subspace [27]	90.85	83.00	83.45	96.91	90.25	89.46
Greedy DDL [11]	90.20	81.83	83.42	96.82	88.97	89.13
Proposed	91.26	83.62	83.86	97.06	90.43	90.06

Table 1.5: Clustering Results on YaleB

In [27], thorough experimentation was carried out with a host of other clustering techniques. The deep subspace clustering method yielded the best results. So we only compare with [27]. We see that the previous greedy DDL technique does marginally worse than deep subspace clustering. But with our proposed optimal algorithm, the results improve significantly and we are able to outperform [27] (only in the F-score we do slightly worse for the COIL20 dataset in the second place of decimal).

1.4 Discussion

A new framework for deep learning called deep dictionary learning has been proposed recently. Idea is to represent the training data as a non-linear combination of several layers of dictionaries. Existing work on DDL solve the problem greedily. It was a sub-optimal approach since there was no flow of information from deeper layers to the shallower layers. In this work all the dictionaries and coefficients are solved simultaneously in a single optimization problem. It is an unsupervised method, any classifier can be used on the learned features to make predictions. Results are computed on benchmark classification and clustering datasets. The performance is compared with generic methodology. In both the cases the proposed approach produced significantly better results than the existing methods.

Chapter 2

Deep Dictionary Learning - Inverse Problems

Many problems in imaging can be generalized to a linear inverse problem for example image denoising, deblurring, reconstruction, inpainting and image super-resolution. Existing dictionary learning based methods try to learn coefficients and dictionary from the data adaptively during the inversion process. In this chapter a method is proposed to adaptively learn multiple levels of dictionaries during inversion. This results in a deep dictionary learning based inversion process. First, a brief literature review is described followed by proposed methodology. Then section 2.3 presents the experiments and results. Experiments have been carried out on denoising, super-resolution and reconstruction.

2.1 Literature Review

In generalized form, an inverse problem can be given by the equation:

$$y = Ax + n \quad (2.1)$$

Here x is the image to be estimated, y is the measurement and n is noise (assumed to be normally distributed). The operator A defines the problem; for denoising it is simply identity, for deblurring it is convolution (with the blurring kernel); for super-resolution it is sub-sampling, for inpainting it is a restriction; and for reconstruction it is a projection. Literature on solving inverse problem in image can be broadly sub-divided into two categories: compressed sensing (CS) based techniques and neural network based inversion. Next subsections briefly discuss both the techniques.

2.1.1 Compressed Sensing based methods

Traditionally sparsity / compressed sensing (CS) based techniques have been used for solving inverse problems [28, 29, 30, 31]. The basic approach in these studies is to exploit the sparsity of the image in some domain (wavelet, DCT etc.) in order to recover it. Typically 2.1 is expressed as follows after incorporating transform (ϕ) domain sparsity,

$$y = Ax + n = A\phi\alpha + n \quad (2.2)$$

Here α is the sparse representation of the image in ϕ . Usually l_1 norm minimization is used to recover sparse coefficients.

$$\hat{\alpha} = \min_{\alpha} \|y - A\phi\alpha\|_2^2 + \lambda\|\alpha\|_1 \quad (2.3)$$

Once the sparse coefficients are learned the image is approximated by,

$$\hat{x} = \phi\hat{\alpha} \quad (2.4)$$

Equation 2.3 represents synthesis prior formulation. One can directly solve for the image by formulating the inversion as a co-sparse analysis prior formulation.

$$\hat{x} = \min_{\alpha} \|y - Ax\|_2^2 + \lambda\|\phi^T x\|_1 \quad (2.5)$$

These CS techniques assumes that the sparsifying basis is known i.e. DCT, wavelet etc. But the natural question that arises is: are such fixed basis the best possible option? The performance of CS is largely determined by the sparsifying capability of the transform; the sparser the representation the better is the outcome [32, 33]. The limitation of fixed transform paved way for learning based denoising. The most popular and successful techniques are based on adaptive learning of the sparsifying basis. K-SVD [34] is the most well known one; it is based on the dictionary learning approach. In dictionary learning, a basis (dictionary) is learnt from the patches of the image it is estimating so that the corresponding coefficients are sparse. The

entire formulation is given by,

$$\min_{x,D,Z} \|y - Ax\|_2^2 + \lambda \left(\sum_i \|P_i x - D z_i\|_2^2 + \gamma \|z_i\|_1 \right) \quad (2.6)$$

Here the first term $\|y - Ax\|_2^2$ is called the data consistency term and the rest is called dictionary learning term. Z is formed by stacking z_i^s as columns and $P_i x$ represents the i^{th} patch of image; this is represented by dictionary D and sparse coefficients z_i .

Dictionary learning has been used profusely in solving inverse problems in imaging. In [35] it has been used for denoising. In [36] it has been used for super-resolution; in [37] for inpainting and in [38] for reconstruction.

2.1.2 Neural Network based Techniques

In the recent past, stacked autoencoder based techniques were popular for solving denoising problems [39, 40, 41, 42]. The input and the output for stacked autoencoders are in the same domain. It has multiple levels of encoders and decoders; diagrammatically shown in Figure 2.1. The input to the stacked denoising autoencoder is a corrupted version of the image and at the output is the clean image. During training the autoencoder learns to clean the image. During the operational / testing stage a noisy image is input to the stacked autoencoder and at the output one expects the clean version of it. This idea was extended in [43] to solve reconstruction problems. It is well known in compressed sensing literature that applying the A^T on the measurements ($y = Ax$) generates a noisy version of the signal.

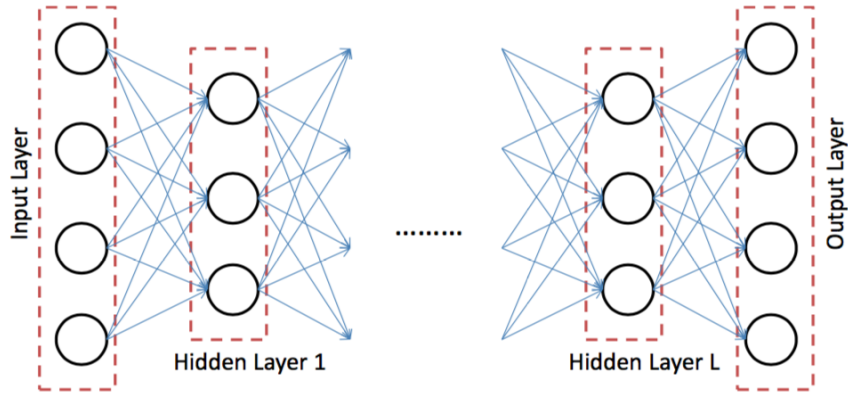


Figure 2.1: Stacked Autoencoder

This noisy version is input to the stacked autoencoder during training and the clean version is presented at the output.

In recent times CNN based inversion has been gaining popularity. It has been used for solving problems in denoising and deblurring [44], and super-resolution [45]. Recently it has been used for solving compressively sampled images [46] in the same fashion proposed in [43].

One advantage of neural network (autoencoder or CNN) based inversion is that, one does not need to change the training algorithm in any fashion depending on the noise / blurring / projection model. Only the training data needs to be corrupted. However, even for one type of noise, different neural networks need to be trained for different amounts of noise / blur / projection; i.e. an autoencoder learnt to denoise Gaussian noise with 0 mean and standard deviation 25 will not be able to clean noise of 0 mean and standard deviation 5. Two different SSDAs need to be learnt. The same applies for other problems. A CNN learnt to denoise motion blur will not be able to remove other kinds of blur. Neural network based inversion

techniques are not adaptive.

The other problem of neural network based inversion is that it is heavily dependent on training data; it expects the test data to be similar to the training data, i.e. belonging to the same modality. The experiments in the most of previous studies have been carried out on natural images. However, natural images are hardly corrupted in practice. It is the scientific imaging modalities, like SAR, satellite, MRI, CT, USG, hyper-spectral etc. that need inversion in reality.

2.2 Proposed Formulation

We propose an adaptive deep learning based approach for solving inverse problems. It is based on deep dictionary learning. In standard (shallow) dictionary learning, each patch ($P_i x$) of the image is expressed as a product of the learned basis (D) and sparse coefficients (z_i).

$$P_i x = D z_i \quad \forall i \quad (2.7)$$

The formulation has already been mentioned in 2.6 but it is repeated here for convenience.

$$\min_{x, D, Z} \|y - Ax\|_2^2 + \lambda \left(\sum_i \|P_i x - D z_i\|_2^2 + \gamma \|z_i\|_1 \right)$$

We extend from single level to multi-level deep dictionaries. Therefore each patch is represented by multiple levels of dictionaries.

$$P_i x = D_1 \phi(D_2 \phi(\dots \phi(D_N z_i))) \quad (2.8)$$

We are showing the derivation for three layers, but the approach we follow is easily extendable to fewer or more layers.

As in shallow dictionary learning, we will have a data consistency term $\|y - Ax\|_2^2$ and a deep dictionary learning term $\left(\sum_i \|P_i x - D_1 \phi(D_2 \phi(D_3 z_i))\|_2^2 + \gamma \|z_i\|_1 \right)$. Putting both together, we get the final formulation:

$$\min_{x, D_1, D_2, D_3, Z} \|y - Ax\|_2^2 + \lambda \left(\sum_i \|P_i x - D_1 \phi(D_2 \phi(D_3 z_i))\|_2^2 + \gamma \|z_i\|_1 \right) \quad (2.9)$$

We use variable splitting augmented lagrangian approach [14, 47] to solve for all the variables jointly. Our first substitution introduces the proxy variable $z_{1,i} = \phi(D_2 \phi(D_3 z_i))$; this is basically the coefficients after the first layer. With this proxy the augmented lagrangian is given by:

$$\min_{x, D_1, D_2, D_3, Z_1, Z} \|y - Ax\|_2^2 + \lambda \left(\sum_i \|P_i x - D_1 z_{1,i}\|_2^2 + \gamma \|z_i\|_1 \right) + \mu \|Z_1 - \phi(D_2 \phi(D_3 Z))\|_F^2 \quad (2.10)$$

Here Z_1 is formed by stacking the $z_{1,i}$'s as columns. The next substitution is $Z_2 = \phi(D_3 Z)$; this is the coefficients at the second stage of dictionary learning. The corresponding augmented Lagrangian is given by:

$$\min_{x, D_1, D_2, D_3, Z_1, Z_2, Z} \|y - Ax\|_2^2 + \lambda \left(\sum_i \|P_i x - D_1 z_{1,i}\|_2^2 + \gamma \|z_i\|_1 \right) + \mu \left(\|Z_1 - \phi(D_2 Z_2)\|_F^2 + \|Z_2 - \phi(D_3 Z)\|_F^2 \right) \quad (2.11)$$

Here Z_2 is formed by stacking the $z_{2,i}$'s as columns. Similarly Z is formed

by stacking the z 's as columns.

For more layers, we can continue in a similar fashion. The idea is to substitute the coefficients for the next stage of dictionary learning. But for the current three layer problem 2.11 is the final form.

Using alternating direction method of multipliers [16], 2.11 can be segregated into the following sub-problems.

$$P_1 : \min_x \|y - Ax\|_2^2 + \lambda \sum_i \|P_i x - D_1 z_{1,i}\|_2^2$$

$$P_2 : \min_{D_1} \sum_i \|P_i x - D_1 z_{1,i}\|_2^2 \equiv \min_{D_1} \|X - D_1 Z_1\|_F^2$$

$$P_3 : \min_{D_2} \|Z_1 - \phi(D_2 Z_2)\|_F^2 \equiv \min_{D_2} \|\phi^{-1}(Z_1) - D_2 Z_2\|_F^2$$

$$P_4 : \min_{D_3} \|Z_2 - \phi(D_3 Z)\|_F^2 \equiv \min_{D_3} \|\phi^{-1}(Z_2) - D_3 Z\|_F^2$$

$$P_5 : \min_{Z_1} \|X - D_1 Z_1\|_F^2 + \mu \|Z_1 - \phi(D_2 Z_2)\|_F^2$$

$$P_6 : \min_{Z_2} \|Z_1 - \phi(D_2 Z_2)\|_F^2 + \mu \|Z_2 - \phi(D_3 Z)\|_F^2 \equiv \min_{Z_2} \|\phi^{-1}(Z_1) - D_2 Z_2\|_F^2 + \mu \|Z_2 - \phi(D_3 Z)\|_F^2$$

$$P_7 : \min_Z \|Z_2 - \phi(D_3 Z)\|_F^2 + \lambda \gamma \|Z\|_1 \equiv \min_Z \|\phi^{-1}(Z_2) - D_3 Z\|_F^2 + \lambda \gamma \|Z\|_1$$

Since the activation functions are sigmoid or tanh, and operate element-wise, they are trivial to invert. We find that the sub-problems P_1 to P_6 are all simple linear least squares problems having a closed form solution. Only P_7 is an l_1 -norm minimization problem that needs to be solved iteratively. But well known existing algorithms like iterative soft thresholding are there to solve it.

This concludes the derivation of the algorithm. It is an iterative algorithm. The variables need to be initialized; this is done randomly. In every iteration, the aforesaid sub-problems are solved to update the variables. The

problem is non-convex therefore there is no global convergence guarantee. In practice we stop the iterations when there is no change between two successive iterations of the objective function. Once the algorithm converges, the image is estimated. In all the substitutions we have used the same multiplicative factor μ . This is because this factor controls the relative importance of each level. Since there is no reason to favor one level over the other, they are kept the same. Moreover, the inputs and the other levels are given equal importance, hence the value of μ is fixed at unity.

2.3 Experiments and Results

2.3.1 Denoising

The first set of experiments has been carried out on image denoising. We carry out experiments on both natural images and scientific modalities. For natural images, the standard test images - Lena, Barbara, Cameraman and Peppers are used. For scientific modality we show results on magnetic resonance imaging (MRI) (brain) and hyperspectral imaging (Washington DC [42]).

The proposed method requires specifying only two parameters λ and γ . In [48] it has been argued that the value of this should be the standard deviation of noise divided by 30. We follow the same empirical formula. For the sparsity parameter, the value of γ is tuned on a separate validation image; we find that when its value is .1, it yields consistently good perfor-

mance. We have used tanh activation functions for all the experiments.

In the initial experiments we study the effect of number of layers and patch-size on denoising. For the initial study, we only concentrate on natural images since they are available in different resolutions. We take each of the images in 128×128 , 256×256 and 512×512 resolution. Gaussian noise is added with standard deviation $\sigma = 25$ and mean 0. For deep dictionary learning the number of nodes in subsequent layers of the encoder is halved from the previous one. Results for Lena and Peppers images are shown in Tables 2.1 and 2.2; for both the experiments overlapping patches have been taken. The quantitative measure for image quality will be peak signal to noise ratio (PSNR).

Image size	Patch size	#layers = 1	#layers = 2	#layers=3	#layers =4
128×128	8×8	31.32			
	16×16	31.01	30.24		
256×256	8×8	31.37			
	16×16	31.11	31.43		
	32×32	31.06	31.78	31.04	
512×512	8×8	31.38			
	16×16	31.19	31.58		
	32×32	31.14	31.91	31.99	
	64×64	31.14	32.04	32.25	32.15

Table 2.1: Lena-Effect of Patchsize and Layers

We find that for small images, going deep does not help. This is expected; because there are fewer samples which in turn lead to over-fitting. But when the size of the image is larger, the deeper version outperforms the shallow one. This is because for larger images, there are enough samples for deep dictionary learning and hence the deep model does not overfit.

Image size	Patch size	#layers = 1	#layers = 2	#layers=3	layers =4
128×128	8×8	29.73	29.34		
	16×16	29.60			
256×256	8×8	29.74	29.91	29.61	
	16×16	29.70			
	32×32	29.70	30.08		
512×512	8×8	29.78	30.01	30.68	30.55
	16×16	29.78			
	32×32	29.77	30.43		
	64×64	29.76	30.76	30.92	

Table 2.2: Peppers-Effect of Patchsize and Layers

We also compare with the standard denoising approaches - dictionary [35], transform [49], BM3D [50], stacked sparse denoising autoencoder (SSDA) [41] and DnCNN [44]; the first three are adaptive techniques while the last two are not.

Experiments have been carried out on two noise levels, $\sigma = 10$ and $\sigma = 50$. The natural images are of size 512×512 . The brain MRI image is of size 512×512 ; it is publicly available from [17]. The WDC mall hyper-spectral image is of size $1152 \times 256 \times 192$. We use a 3 layer dictionary learning network for the purpose of denoising since it yields the best result at this resolution.

Dataset/Method	KSVD	Transform	SSDA	BM3D	DnCNN	Proposed
Lena	36.91	37.62	37.90	37.92	35.82	37.71
Barbara	34.42	34.55	34.67	34.21	33.66	34.42
Cameraman	33.72	33.87	33.26	33.28	33.17	34.02
Pepper	34.67	35.19	33.32	35.75	33.42	35.15
Brain	38.54	38.78	30.42	38.79	31.78	39.23
WDC	33.48	33.67	28.77	33.41	29.13	33.99

Table 2.3: Comparative Denoising Performance ($\sigma = 10$)

Dataset/Method	KSVD	Transform	SSDA	BM3D	DnCNN	Proposed
Lena	27.79	28.01	26.24	29.05	29.39	29.31
Barbara	25.47	25.92	25.22	27.22	26.22	27.37
Cameraman	26.13	26.45	26.98	26.13	27.03	27.61
Pepper	26.66	27.10	26.24	26.68	27.32	27.11
Brain	24.73	24.83	19.46	24.97	19.94	25.22
WDC	22.38	22.30	17.81	22.59	18.11	22.97

Table 2.4: Comparative Denoising Performance ($\sigma = 50$)

Experimental results in Tables 2.3 and 2.4, show that our proposed method consistently yields good results. In inverse problems in imaging a 0.5dB variation in PSNR is considered minor. It is seen that even when our proposed method does not perform the best, we are within 0.5dB of the very best.

What is interesting to note is that the non-adaptive neural network based methods - SSDA and DnCNN, yields results at par with the popular benchmark algorithms only for natural images; but yield results significantly sub-par for scientific imaging modalities like MRI and hyper-spectral imaging. This is expected. Since SSDA and DnCNN have been trained on a large volume of natural images, its performance on similar images are good, but fails to generalize on unseen imaging modalities.

For visual evaluation we show results on the brain MRI and the one spectral band (number 50) of the hyperspectral image of WDC in Figure 2.2. For the brain, we follow the convention of medical imaging. Instead of showing the denoised image, we show the difference (between denoised and original) image. This helps in visually understanding the denoising artifacts. Difference image from our proposed method is almost completely

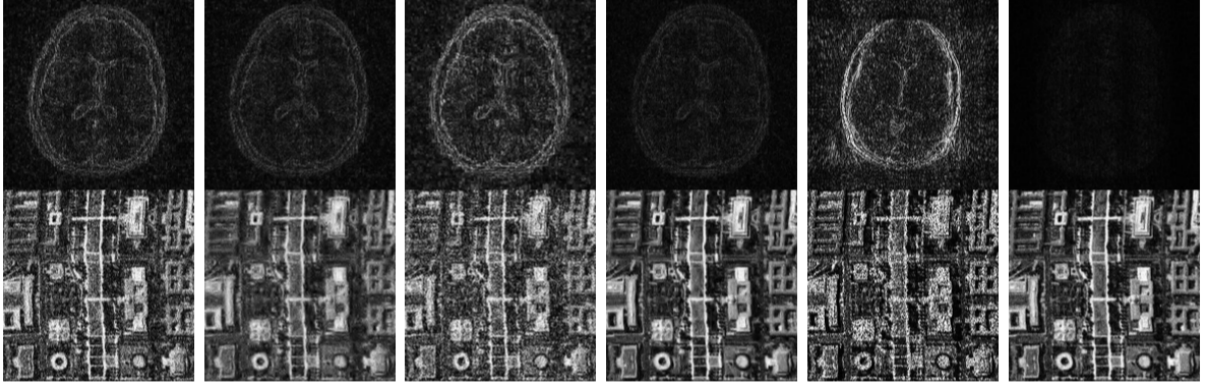


Figure 2.2: Top. Denoising Difference Image of Brain MRI. Bottom. Denoising Hyperspectral Image (band 50) WDC. Left to Right-KSVD, Transform Learning, SSDA, BM3D,DnCNN and Proposed

dark, meaning that there are hardly any denoising artifacts. The artifacts are slightly more pronounced in BM3D, Transform learning and KSVD; but are the worst in SSDA.

For the hyperspectral image, we show denoising results for low Gaussian noise ($\sigma = 10$); this is a realistic noise level. The conclusions remain the same. KSVD is a bit noisy but maintains detailed edges. The transform learning approach overtly smooths the image. But the worst one is from SSDA and DnCNN. BM3D yields good results; but proposed is better.

2.3.2 Super-resolution

In super-resolution, the operator A (used in our method) is a sub-sampling operator. For our proposed method, we show results for 1, 2 and 3 layers. Experiments have been carried out on the same set of popular test images - Lena, Barbara, Cameraman and Peppers. We used 256×256 as low resolution inputs for 4 times magnification and 512×512 as low resolution inputs for 2 times magnification; the 1024×1024 was the ground truth

for comparing. The parameters λ and γ are learned on separate validation images; we found the values to be .25 and .1 respectively.

In [45] it is shown that CNN based super-resolution (SRCNN) excels over dictionary learning methods [31]. Coupled dictionary learning (CDL) [51] also beats dictionary learning [31]. Therefore in this work, we will only compare with the learning based techniques - SRCNN and CDL.

Images/ Method	CDL	SRCNN	DL	Proposed 2	Proposed 3
Lena	27.92	28.39	27.76	28.94	29.13
Barbara	24.76	26.10	23.59	24.85	25.30
Cameraman	26.36	27.91	26.22	26.57	27.04
Peppers	26.87	28.31	26.78	28.23	28.68

Table 2.5: Comparative Super-resolution Performance (Magnification = 2)

Images/ Method	CDL	SRCNN	DL	Proposed 2	Proposed 3
Lena	20.11	21.84	20.16	21.47	21.73
Barbara	17.17	18.63	17.11	18.54	18.65
Cameraman	18.03	19.74	18.31	19.78	20.12
Peppers	18.45	20.03	18.49	19.99	20.37

Table 2.6: Comparative Super-resolution Performance (Magnification = 4)

The results for different magnification factors from different techniques are shown in Tables 2.5 and 2.6. The metric for quantitative assessment is PSNR. The comparative results are consistent with prior works; the single layer formulation (DL) performs worse than CDL and SRCNN. What we find is that with more layers, our result starts to improve. We get the best results with three layers; we tried four layers but the performance degraded. With three layers, we outperform the others in most cases.

For visual evaluation, the Barbara image is shown in Figure 2.3. We see



Figure 2.3: Results with 4 times magnification. Left to Right - Original, CDL, SRCNN and Proposed 1 layer

that although CDL and SRCNN yield higher PSNR, the image is not visually as good as proposed; they are too smooth; it is well known that PSNR can be improved by smoothening out an image, it is not the best metric for visual quality.

2.3.3 Reconstruction

The experimental evaluation was carried out on ex-vivo and in-vivo T2 weighted images of a rat's spinal cord. The data were collected with a 7T MRI Bruker scanner in accordance to the university's ethics policies. The original ground-truth data is of size 256×256 acquired by spin-echo sequence.

We are comparing with following MRI reconstruction techniques: dictionary learning (DL) [38] and transform learning (TL) [52]. It has been shown in the aforesaid studies that these techniques improves upon compressed sensing based methods; hence we are not competing with them.

For our proposed method we evaluate a two-tier and a three-tier architecture. Dictionary based technique is equivalent to our one-tier architecture. The number of basis is reduced by half in each layer. The value of the pa-

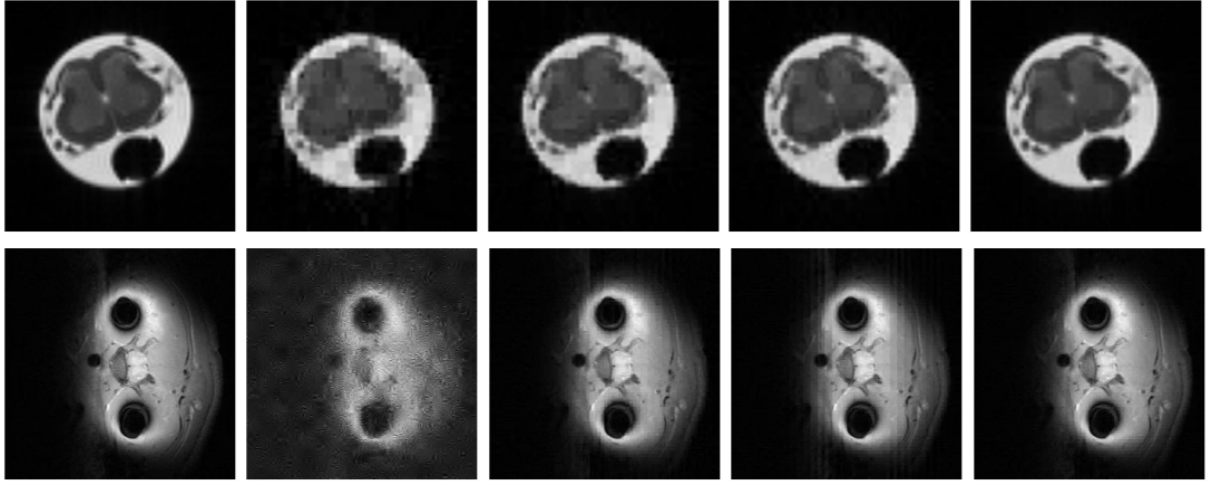


Figure 2.4: Reconstructed Images. Top- Ex Vivo, Bottom- In Vivo. Left to Right- Original, TL, DL, Proposed 2 layer and Proposed 3 layer

rameters λ and γ has been tuned on a separate MRI image; the values we used here are .2 and .1 respectively.

Method	Ex Vivo Imaging			In Vivo Imaging		
	128 lines	64 lines	32 lines	128 lines	64 lines	32 lines
DL	24.7	20.6	16.9	23.9	20.2	15.6
TL	24.8	20.9	17.1	24.0	20.2	15.8
Proposed-2 layer	26.4	22.7	20.2	24.5	21.1	17.6
Proposed-3 layer	26.9	23.1	20.9	24.9	21.7	18.0

Table 2.7: Reconstruction Accuracy in SNR (dB)

Table 2.7 shows results for Ex Vivo and In Vivo imaging for all acceleration factors. Signal to noise ratio (SNR) is used as an evaluation metrics. It can be seen that proposed approach outperforms the other popular benchmark methods.

For visual evaluation we have shown the reconstructed images and the difference images in Figure 2.4 and 2.5 respectively. The difference image gives an idea regarding reconstruction artifacts; smaller the artifacts darker the image and better the reconstruction. We find that TL performs the worst

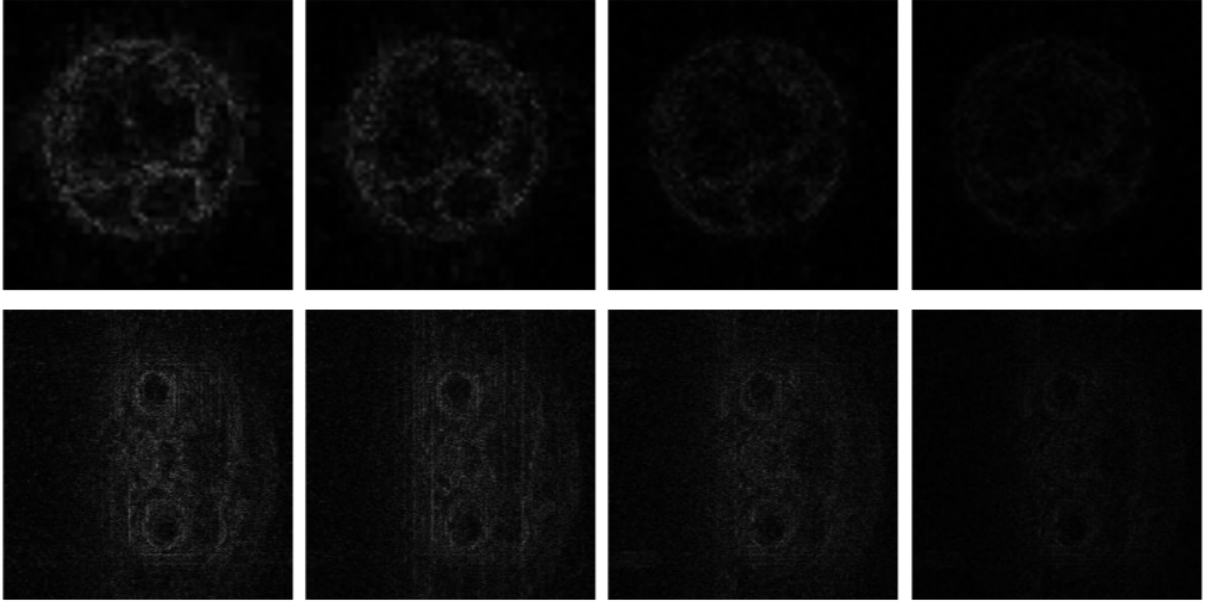


Figure 2.5: Difference Images. Top- Ex Vivo, Bottom- In Vivo. Left to Right- TL, DL, Proposed 2 layer and Proposed 3 layer

and DL improves the results upon it. As we go deeper, the results improves. The best results are obtained with three layers. We also computed results with four layers, but the performance degrades owing to overfitting.

2.4 Discussion

In this work a deep dictionary learning based adaptive approach is proposed to solve inverse problems. Experiments have been carried out on benchmark inverse problems- denoising, super-resolution and reconstruction. Results are compared with popular benchmark adaptive and CNN based non-adaptive solutions. Proposed method outperform the existing methods. The disadvantage of the proposed method is that it is slower as compared with compressed sensing based methods. Therefore, it cannot be used where reconstruction speed is of essence.

Chapter 3

Supervised Deep Dictionary Learning

The prior studies have used DDL in an unsupervised fashion; they needed a third party classifier. The unsupervised DDL based formulations are optimal for solving problems like denoising, reconstruction and inpainting but it may not lead to optimal solution in classification tasks. In such tasks the goal is to have the dictionaries and coefficients with high discriminative capabilities. This motivates to learn DDL in a supervised framework that can classify directly without any need of an external classifier. The formulation can be generalized to perform both single label and multi-label classification. First, proposed algorithm is discussed in Section 3.2. Experiments are performed on benchmark single label and multi label classification problems. In section 3.3.3 proposed algorithm is applied on a practical application of non intrusive load monitoring.

3.1 Literature Review

Dictionary learning is a well researched topic. It has roots in signal decomposition in some predefined basis such as wavelet [53]. Representation of data into these predefined basis assumes that these basis are general enough to represent any kind of signal. Which is not true in practice. In recent research it is seen that learning the basis from data leads to state-of-art results in many applications like image compression [54], denoising [35], image super-resolution [31], medical imaging [55], signal classification [56] etc. Performance of unsupervised DL methods are sub-optimal in classification tasks, as label information is not used while training. This motivates supervised dictionary learning techniques which uses label information during training phase. One of the first studies in supervised dictionary learning was proposed in [57]. Here on top of the dictionary learning term there is an extra term which account for classification error.

$$\min_{D,Z,\theta} \|X - DZ\|_F^2 + \lambda C(y, f(Z, \theta)) \quad (3.1)$$

where $y \in \{-1, 1\}$ and $C(\theta) = \log(1 + e^{-x})$ is the classification error penalty that is very similar to the hinge loss used in SVM; $f(\theta, Z) = \theta Z + b$. In various recent studies different classification error terms are introduced. In the rest of the section some of the supervised dictionary learning based techniques have been discussed.

– **Sparse Representation based Classifier (SRC) [58]**

SRC learns a separate dictionary for each class. 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. Let us assume that x_{test} is the test sample, training data $X = [X_1, X_2 \dots X_c]$ where X_i is the data belonging to class i and $\alpha = [\alpha_1, \alpha_2 \dots \alpha_c]$ are the coefficients. The task is to assign the test sample to the correct class. The test data can be expressed in terms of all the training samples:

$$x_{test} = X\alpha + \eta \quad (3.2)$$

According to the SRC assumption only those α 's corresponding to the correct class are non-zero. Therefore, one needs to solve the inverse problem 3.2 with sparsity constraints on the solution. This is formulated as:

$$\min_{\alpha} \|x_{test} - X\alpha\|_F^2 + \lambda \|\alpha\|_1 \quad (3.3)$$

In the next step, residual error is computed using the following equation:

$$r_i(x_{test}) = \|x_{test} - X f_i(\alpha)\|_2^2 \quad (3.4)$$

Where f_i selects the coefficients with class i . Therefore once the residual for every class is obtained, the test sample is assigned to the class having the minimum residual.

Advantage of this approaches is the ease of computation of dictionary.

SRC based method does not need learning of a dictionary since it uses training data as dictionary. However, the main disadvantage of such method is the size of the dictionary, as the size of the dictionary increases with number of classes. To obtain a smaller dictionary, Yang et al. [59] proposed an approach to learn small dictionary per class and merge them into one dictionary.

– **Joint Dictionary and Classifier Learning**

Unlike SRC where a separate dictionary is used for each class, [48, 60, 57] learn the dictionary and classifier parameters in a joint optimization problem.

Discriminative K-SVD (DK-SVD) [48] and label consistent K-SVD (LCK-SVD) [60] are such methods. DK-SVD learns the dictionary and classifier parameters together, without alternating between them. It prevents the solution to get stuck in local minima. However, the classifier in DK-SVD is linear, which may not be an ideal choice for a difficult classification problem. In DK-SVD there is no way of ensuring that the dictionaries learned are discriminative. To overcome this problem, LC-KSVD add a label consistency constraint on the dictionary.

3.2 Proposed Formulation

The formulation for deep dictionary learning is given by

$$\min_{D_1, D_2, \dots, D_N, Z} \|X - D_1 \phi(D_2 \phi(\dots \phi(D_N Z)))\|_F^2 \quad (3.5)$$

We add a label consistency term to the formulation (3.5) to incorporate supervision. In label consistent dictionary learning (LCDL) [61], a penalty term is added that minimizes the error between the mappings of the learned representations and target binary class labels. For shallow dictionary learning [61], the formulation is given by:

$$\min_{D, Z, W} \|X - DZ\|_F^2 + \lambda \|T - WZ\|_F^2 \quad (3.6)$$

Here, first term is the standard dictionary learning term. In the second term, W is the linear map between the learned representations Z to the target binary class labels T . λ is a regularization parameter. In, T positions of the corresponding class labels is 1 and rest are 0's.

In [62] it is shown that label consistent formulation can be extended to multi label classification problem. In such problems a single sample may belong to multiple output classes which is the case in most standard classification problems.

In our proposed formulation we add a label consistency term on top of the

deep dictionary learning problem (LCDDL) (3.5), given by:

$$\min_{D_1, D_2 \dots D_N, Z, W} \|X - D_1 \phi(D_2 \phi(\dots \phi(D_N Z)))\|_F^2 + \lambda \|T - WZ\|_F^2 \quad (3.7)$$

Forbenius norm is not a very good choice for the penalty term here. Ideally there should be a thresholding kind of non-linearity to map the term WZ to 0 and 1, which gives:

$$\min_{D_1, D_2 \dots D_N, Z, W} \|X - D_1 \phi(D_2 \phi(\dots \phi(D_N Z)))\|_F^2 + \lambda \|T - \phi(WZ)\|_F^2 \quad (3.8)$$

Since thresholding is not differentiable, we propose to use sigmoid to approximate the ideal thresholding operation. This keeps 3.8 smooth and differentiable.

3.2.1 Training

In order to jointly learn all the layers we use a variable splitting method [63] to solve deep dictionary learning problem. In the formulation given by equation 3.8 for the first layer we substitute $Z_1 = \phi(D_2 \phi(\dots \phi(D_N Z)))$. With this substitution the augmented Lagrangian [15] of 3.8 is given by equation:

$$\begin{aligned} \min_{D_1, D_2 \dots D_N, Z_1, Z, W} & \|X - D_1 Z_1\|_F^2 + \mu_1 \|Z_1 - \phi(D_2 \phi(\dots \phi(D_N Z)))\|_F^2 \\ & + \lambda \|T - \phi(WZ)\|_F^2 \end{aligned} \quad (3.9)$$

For the second layer we substitute $Z_2 = \phi(D_3 \dots \phi(D_N Z))$. The augmented Lagrangian now becomes:

$$\min_{D_1, D_2 \dots D_N, Z_1, Z_2, Z, W} ||X - D_1 Z_1||_F^2 + \mu_1 ||Z_1 - \phi(D_2 Z_2)||_F^2 + \mu_2 ||Z_2 - \phi(D_3 \dots \phi(D_N Z))||_F^2 + \lambda ||T - \phi(W Z)||_F^2 \quad (3.10)$$

We continue until the second last layer. The final augmented Lagrangian takes the form:

$$\min_{D_1, D_2 \dots D_N, Z_1, Z_2 \dots Z, W} ||X - D_1 Z_1||_F^2 + \mu_1 ||Z_1 - \phi(D_2 Z_2)||_F^2 + \dots + \mu_{N-1} ||Z_{N-1} - \phi(D_N Z)||_F^2 + \lambda ||T - \phi(W Z)||_F^2 \quad (3.11)$$

The multiplicative terms μ' s indicate the relative weightage of each layer. These need to be tuned in general, but there is no reason to give one layer more weightage than the other. However, the last layer in case of supervised learning is different from the other layers and should be treated differently. We have kept last layer as same as the others for the sake of simplicity. Alternating direction method of multipliers (ADMM) [16] has been used to solve equation 3.11. Following the ADMM we can segregate 3.11 into following sub-problems:

$$\begin{aligned} P_1 : \min_{D_1} ||X - D_1 Z_1||_F^2 \\ P_2 : \min_{D_2} ||Z_1 - \phi(D_2 Z_2)||_F^2 &\equiv \min_{D_2} ||\phi^{-1}(Z_1) - D_2 Z_2||_F^2 \\ . \\ . \end{aligned}$$

.

$$P_N : \min_{D_N} \|Z_{N-1} - \phi(D_N Z_N)\|_F^2 \equiv \min_{D_2} \|\phi^{-1}(Z_{N-1}) - D_N Z_N\|_F^2$$

$$P_{N+1} : \min_{Z_1} \|X - D_1 Z_1\|_F^2 + \|Z_1 - \phi(D_2 Z_2)\|_F^2$$

$$P_{N+2} : \min_{Z_2} \|Z_1 - \phi(D_2 Z_2)\|_F^2 + \|Z_2 - \phi(D_3 Z_3)\|_F^2 \equiv \min_{Z_2} \|\phi^{-1}(Z_1) - D_2 Z_2\|_F^2 + \|Z_2 - \phi(D_3 Z_3)\|_F^2$$

.

.

.

$$P_{N+N} : \min_Z \|Z_{N-1} - \phi(D_N Z)\|_F^2 + \|T - \phi(W Z)\|_F^2 \equiv \min_Z \|\phi^{-1}(Z_{N-1}) - D_N Z\|_F^2 + \|\phi^{-1}(T) - W Z\|_F^2$$

$$P_{N+N+1} : \min_W \|T - \phi(W Z)\|_F^2 \equiv \min_W \|\phi^{-1}(T) - W Z\|_F^2$$

All these sub-problems are simple least square problems having closed form solutions in Moore-penrose pseudo-inverse. The activation function ϕ is unitary, applied element wise hence it is trivial to invert. This completes the training phase.

3.2.2 Testing

At the time of testing learned dictionaries are used to generate test features. In case of unsupervised formulation any appropriate classifier can be used to generate test labels. In supervised case the generated features are multiplied with map W to get the test labels.

The training part remain the same for both single label and multilabel clas-

sification problems. The only difference is in the definition of T . In single label case, only one position in the column of T has one, but in multi label case more than one positions can take value one. The testing part can be divided into two parts: generating test features from the data and assigning labels to the test samples. The first part of testing for both single label and multi label case remain the same. For generating the representations z for test sample x we need to solve the problem below:

$$\min_z ||x - D_1\phi(D_2...\phi(D_N z))||_2^2 \quad (3.12)$$

This is solved using substitution as before. For the first layer we substitute $z_1 = \phi(D_2...\phi(D_N z))$, for the second layer it is $z_2 = \phi(D_3...\phi(D_N z))$ and so on until the second last layer. The augmented lagrangian is given by:

$$\min_{z_1, z_2, \dots, z_N, z} ||x - D_1 z_1||_2^2 + ||z_1 - \phi(D_2 z_2)||_2^2 + \dots + ||z_N - \phi(D_N z)||_2^2 \quad (3.13)$$

The dictionaries learned during training are used here. The problem 3.13 is segregated into following sub problems:

$$\begin{aligned} S_1 : \min z_1 & ||x - D_1 z_1||_2^2 + ||z_1 - \phi(D_2 z_2)||_2^2 \\ S_2 : \min z_2 & ||z_1 - \phi(D_2 z_2)||_2^2 + ||z_2 - \phi(D_3 z_3)||_2^2 \equiv ||\phi^{-1}(z_1) - D_2 z_2||_2^2 + \\ & ||z_2 - \phi(D_3 z_3)||_2^2 \\ & . \\ & . \\ & . \end{aligned}$$

$$S_N : \min_z \|z_{N-1} - \phi(D_N z)\|_2^2 \equiv \|\phi^{-1}(z_{N-1}) - D_N z\|_2^2$$

This completes the first part of testing. The second part is to assign class label to the test sample x . This is different for single label and multilabel classification problem.

- For single label classification first we compute $t = \phi(Wz)$. The output t is never a binary vector. We assign the class of x to the position of maximum value of t .
- For multilabel classification $t = \phi(Wz)$ is computed. But here we cannot use maximum as one sample may belong to multiple classes. Therefore, a threshold is taken and all the positions in t which take the values above this threshold are assigned as positive classes.

3.3 Experiments and Results

3.3.1 Single Label Classification

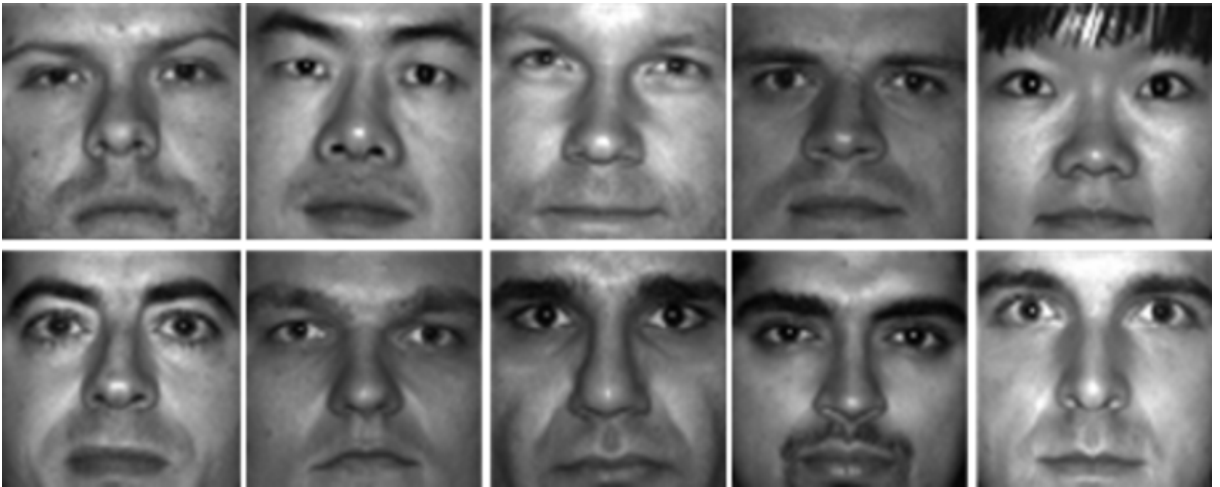


Figure 3.1: Sample Images from YaleB Dataset

– YaleB dataset

To evaluate the performance of proposed method, experiments are performed on benchmark face recognition datasets: AR face [64] and YaleB face [65]. Brief description of both the datasets is given below.

The extended Yale face dataset B (YaleB) include 2414 images of 38 people, which is challenging due to plentiful expression and varying illumination conditions. All the original images are cropped to 192×168 pixels and the projected onto 504-dimensional vectors with a randomly generated matrix (i.i.D Gaussian) to obtain random face features. Figure 3.1 shows some sample images from YaleB dataset.

– AR face dataset

The AR face database contains more than 4000 color face images of



Figure 3.2: Sample Images from AR Dataset

126 people. Each person has 26 frontal face images which are taken

during two sessions. This database includes frontal views of faces with different facial expressions, lighting conditions, and occlusion conditions (sunglasses and scarves). All the images are cropped and scaled to 165×1120 . We followed a common evaluation protocol in our experiments for this database, in which we used a subset of 2600 images pertaining to 50 males and 50 female subjects. For each subject, we randomly chose 20 samples for training and the rest for testing.

Figure 3.1 and 3.2 shows the sample images from YaleB and AR face dataset respectively. These datasets have been used in the highly cited paper on LC-KSVD [61, 66]. We have compared it with popular benchmark techniques. We compared with stacked denoising autoencoders (SDAE) [22], stacked group-sparse autoencoder (SGSA) [67], stacked label consistent autoencoder (SLCA) [17], Deep Belief Network (DBN) [9], discriminative deep DBN (DDBN) [68], deep dictionary learning (DDL) [11] and LC-KSVD [60]. For all the deep techniques, three layers are used and in each layer the number of nodes are reduced by half from the previous layer. All the techniques apart from DDL have their in-built classifiers. For DDL, a separate support vector machine (SVM) classifier with radial basis function (RBF) was used.

For the proposed supervised deep dictionary learning (SDDL) formulation we show the results from one to four layers. Note that the one layer formulation is almost exactly similar to LC-KSVD except for the non-linearity in the label consistency term introduced in our framework. Here we halve the

Method	AR	yaleB
SDAE	37.60	42.81
SGSA	32.50	66.27
SLCA	85.21	86.22
DBN	34.91	43.62
DDBN	38.20	60.34
LC-KSVD	87.67	90.80
DDL	42.66	63.35
Proposed 1-layer	89.45	91.13
Proposed 2-layer	90.31	91.17
Proposed 3-layer	96.50	94.57
Proposed 4-layer	96.16	94.24

Table 3.1: Classification Accuracy on Face Recongition

number of atoms in each subsequent layer. The classification accuracies are shown in the table 3.1.

The results shows that proposed method outperforms the rest. Even for the single layer proposed method performs better than the LC-KSVD, this is because of the non-linearity we have introduced.

There is a variation in the results with layers in the proposed method. From, one to three layers results improves. But the results in the fourth layer are worse than that of three layers; this is because of the over-fitting. For the fourth layers there are too many parameters to learn, and with limited volume of data, it over-fits, thereby reducing the classification accuracy during testing.

Although these are image classification datasets, we have not used CNN. This is because it has already been shown in the previous studies [69] that CNN does not perform on these datasets since there are limited training samples.

3.3.2 Multilabel Classification

In multilabel classification problem one sample can belong to multiple classes. In [62] it is shown that label consistent K-SVD [61] can be used for multi-label classification problem. Our work is a deeper extension and hence can be used for multi-label classification as discussed in the previous section.

We have performed experiments on two benchmark datasets: scene and yeast dataset. A brief description of these dataset is as follows:

– Scene Dataset

The dataset consists of 2407 natural scene images, each belonging to one or more out of 6 labels: desert, mountains, sea, sunset and trees. The usual protocol is to use half of the images for training and the rest for testing. Each image is represented by a 294-dimensional feature vector using the procedure described in [70]. The extracted features are spatial color moments in the LUV space, which are commonly used in the scene classification literature.

– Yeast Dataset

It has 2417 samples with 14 labels. Here 1500 samples are used for training and the rest for testing. Each sample is a 103 dimensional feature vector. The data has 14 labels.

We have used two traditional techniques as benchmarks- Random K Label Set (RAKEL) [71] and Multi-label K-Nearest Neighbor (MLKNN) [72]. We

compare against the state-of-the-art label consistent graph dictionary learning (LCGDL) technique [62]. We do not compare with LC-KSVD [61] since it has been shown in [62] that LCGDL outperforms LC-KSVD significantly. All the aforesaid techniques are shallow. We also compared with a deep technique for multi-label classification called canonical correlation autoencoder (CCAe) [73]. For all these techniques, the optimal configuration for the datasets have been taken from the respective papers.

3.3.2.1 Evaluation Metrics

Metrics used for single label classification cannot be used for multi-label classification. Prior work [71] proposed to use F_1 macro and F_1 micro evaluation measures. Given the number of true positives (TP), false positives (FP), and false negatives (FN) the F_1 measure is defined as:

$$F1 = \frac{2 * TP}{2 * TP + FP + FN} \quad (3.14)$$

F_1 macro and F_1 micro are the measures derived from F_1 score. These are label based evaluation measures which depend on the averaging method (macro or micro) used [71]. F_1 micro measure is computed by averaging the $F1$ scores for each label. Whereas, F_1 macro is computed after summing true positives, false positives and false negatives across all labels.

$$F1_{macro} = F1 \left(\sum_{i=1}^N TP_i, \sum_{i=1}^N FP_i, \sum_{i=1}^N FN_i \right) \quad (3.15)$$

$$F1_{micro} = \frac{1}{N} \sum_{i=1}^N F1(TP_i, FP_i, FN_i) \quad (3.16)$$

Here, TP_i , FP_i and FN_i denote the number of true positives, false positive and false negative for label i . N is the number of labels in the dataset.

3.3.2.2 Results

Results on both the datasets are shown in the tabel 3.2.

Method	Scene		Yeast	
	Micro F_1	Macro F_1	Micro F_1	Macro F_1
MLKNN	72.3	72.6	63.9	36.4
RAKEL	69.6	70.3	61.8	40.6
LCGDL	74.2	73.5	65.3	39.7
CCAE	65.2	69.7	57.9	33.2
Proposed 1-layer	74.9	74.1	65.2	39.2
Proposed 2-layer	77.6	77.0	67.8	43.7
Proposed 3-layer	75.9	75.8	66.8	42.9

Table 3.2: Classification Accuracy on Scene and Yeast datasets

The only deep learning technique compared against is CCAE, and it shows very poor results. This is because, there are too many parameters to learn, since it is an autoencoder based technique. There is an encoder and decoder for every level, which is twice the number of parameters in dictionary learning.

Of the previous techniques LCGDL yields the best results. Our single layer method yields results comparable to it. This is because both of them have almost the same architecture. LCGDL improves upon LC-KSVD by graph regularization, whereas we improve upon LC-KSVD by introducing non-linearity in the label consistency penalty. The deeper (two-layer) archi-

ture yields the best results for SDDL. When the number of layers are increased further, the results deteriorate a bit owing to over-fitting. That is the reason we have not gone any deeper than three layers.

3.3.3 Simultaneous Detection of Multiple Appliances from Smart-Meter Measurements via Multi-label Consistent Deep Dictionary Learning

A practical example of multi-label classification can be found in Non-Intrusive Load Monitoring (NILM). In NILM one tries to predict the states of individual appliances from the aggregated power readings [74]. Ideally NILM should be able to disaggregate the load from only the smart-meter reading. However, this is a very difficult problem (especially at low sampling rates); in practice, a learning based paradigm is followed where the training stage is intrusive, but the testing / operation stage is not. The building is instrumented during the training stage to gather data, from which machine learning models are learned. During operation, the sensors are removed, and the learned models are used to predict the consumption of each device. This paradigm is not fully non-intrusive. The training stage is intrusive requiring deployment of multiple sensors. In recent times a multi-label classification approach provides a fully non-intrusive alternative. It does not require any instrumentation; it only requires the recording of the state-of-the-appliance, i.e. whether it is ON or OFF. During the training stage, given the smart-meter reading and the recorded states of the appliances, a machine learning model learns multi-label classification; here each appli-

ance is treated as a label - since several appliances can be ON at the same time, it turns out to be a multi-label classification problem. During the operational stage, the learned model is used to predict the state of the appliances given the smart-meter readings. In order to estimate the actual power consumption the state is multiplied by the average power consumption of the device.

Prior studies have used off-the-shelf machine learning algorithms for multi-label classification [75, 76, 77, 72, 78], e.g. Multi Label K Nearest Neighbor (MLKNN) [72], Random K-label sets (RAKEL) [71]. In [79], temporal sliding window technique is employed to extract features from the aggregated power data. Binary relevance, classifier chains and LP classifiers (SVM and decision tree as base classifiers) are trained using extracted features. The paper [80], uses delay embedding to extract features from the time series data and compare the results using multi-label classification algorithms. In [79] and [80], the performance of RAKEL and MLKNN are compared on time series and Haar wavelet features extracted from the aggregated power data.

The primary shortcoming of all previous multi-label classification techniques is that they consider all combination of classes as a separate class. This leads to combinatorial growth in the number of classes. This in turn leads to the exponential growth in model parameters. With limited training volume, training so many parameters leads to over-fitting. This leads to poor results in real applications.

Standard deep neural network models based on stacked autoencoder or deep belief networks are not capable of handling multi-label classification; this is because the output layer is trained by logistic regression or soft-max both of which lead to single classes. Therefore such existing deep learning tools cannot be used for the current purpose. We propose to use multi-label consistent deep dictionary learning for estimating the appliance state from aggregated smart meter data. The formulation for label consistent deep dictionary learning has been discussed in detail in the previous sections. We are repeating it here for the sake of convenience.

$$\min_{D_1, D_2 \dots D_N, Z, W} \|X - D_1 \phi(D_2 \phi(\dots \phi(D_N Z)))\|_F^2 + \|T - WZ\|_F^2$$

Here, X is aggregated power meter data and T are the targets. Each target has the same length as the number of appliance; the appliances are in an order. If an appliance is *ON*, the corresponding value is 1 or else it is 0. The map W , projects the coefficients Z to the multi-label target labels T . Training and testing procedure for this is explained in section 3.2.1 and 3.2.2.

3.3.3.1 Experiments and Results

Experiments have been carried out on two benchmark NILM datasets: Reference Energy Disaggregation dataset (REDD) [81] and Pecan street dataset [82]. A brief description of both the datasets is as follows:

– REDD dataset

The REDD dataset consists of both aggregated and appliance level

power data from six houses at 1Hz. However we do not make use of the appliance level consumption data; we only need its state. To emulate real world conditions the samples are averaged over a time period of 1 minute for our experiments. Four high power consuming devices used in our experiments are dishwasher, kitchen outlet, lighting and washer dryer.

– Pecan street dataset

Pecan street dataset consists of one minute appliance level and building level electricity data from 240 houses. For experiments, subsets of 28 houses have been used. For this work 4 most power consuming devices (site meter, air conditioner, electric furnace and sockets) are used for experiments. To prepare training and testing data, aggregated and sub-metered data are averaged over a time period of 1 minutes. Each training sample contains power consumed by a particular device in one day while each testing sample contains total power consumed in one day in particular house. 80% of the houses are assigned to training set and 20% to the test set.

To evaluate the performance, three measures are used: F_1 macro, F_1 micro and energy error. F_1 macro and F_1 micro are from the perspective of information retrieval. For NILM problem a more useful metric would be predicted energy consumption. For that, the energy error is defined in [83].

It is defined as:

$$error = \frac{|\left(\sum_{i=1}^N Average_power_i\right) - \left(\sum_{i=1}^N Actual_power_i\right)|}{\left(\sum_{i=1}^N Actual_power_i\right)} \quad (3.17)$$

Here, N is the number of appliances. We compare with MLKNN and RAKEL as benchmarks which are used in prior studies [79, 80]. These techniques use thresholded wavelet coefficients as input features. Comparison has also been done with AFAMAP [84]; however for this technique only average energy error is reported since it is not a classification method and hence F1 measures cannot be computed.

Finally we compare with the multi-label consistent stacked autoencoder (MLCSAE) technique proposed in [85]. This is the only known prior study on deep learning based multi-label classification. We use the three layer architecture proposed in [85]. The number of nodes are halved in every layer following the usual rule of thumb in such cases.

In deep learning there is no principled way to choose the number of layers or the number of nodes in each layer; it is dependent on the collective experience of the researchers. Usually going deeper helps, but with limited volume of training data, going deeper also results in over-fitting; there is a trade-off. For moderate sized problems, as ours, a three layer architecture usually yields the best results. The number of basis used are 120 – 80 – 50. The results are shown in table 3.3 and 3.4. We have shown results for one to four layers. For the fourth layer, the number of atoms have been halved

from the third layer in each case.

Method	F_1 macro measure	F_1 micro measure	Average energy error
MLCDDL (1 layer)	0.6798	0.6846	0.0944
MLCDDL (2 layers)	0.6857	0.6905	0.0592
MLCDDL (3 layers)	0.7020	0.7046	0.0316
MLCDDL (4 layers)	0.6951	0.6964	0.0427
MLKNN	0.5931	0.6034	0.1067
RAKEL	0.5334	0.5749	0.9948
AFAMAP	-	-	0.2149
MLCSAE	0.6237	0.6301	0.1145

Table 3.3: Performance Evaluation on REDD dataset

Method	F_1 macro measure	F_1 micro measure	Average energy error
MLCDDL (1 layer)	0.7065	0.7033	0.0275
MLCDDL (2 layers)	0.7089	0.7062	0.0223
MLCDDL (3 layers)	0.7100	0.7099	0.0178
MLCDDL (4 layers)	0.7047	0.7026	0.0248
MLKNN	0.6227	0.6263	0.0989
RAKEL	0.6620	0.6663	0.9995
AFAMAP	-	-	0.2371
MLCSAE	0.6641	0.6703	0.0361

Table 3.4: Performance Evaluation on Pecan street dataset

In Tables 3.3 and 3.4, the aggregate results over all the houses are shown. We observe that our proposed deep methods significantly outperforms all other shallow and deep techniques both in terms of F_1 -score as well as energy error. The other observation is that, once we go deeper, the results improve from layers 1 to 3; but when we go even deeper, the problem of over-fitting arises and the results deteriorate. Of the pre-existing shallow techniques, RAKEL, although performs decent in terms of F_1 -score, is very poor in terms of energy error; MLKNN yields more balanced results. The AFAMAP yields better results than RAKEL but is much worse than MLKNN. The other deep learning technique MLCSAE outperforms the

shallow ones but is worse than ours.

Device	MLCDDL		MLKNN		RAKEL		AFAMA	MLCSAE	
	Error	F_1 score	Error	F_1 score	Error	F_1 score	Error	Error	F_1 score
Dishwasher	0.0179	0.5697	0.1250	0.4937	0.9964	0.3413	0.2726	0.0851	0.5124
Kitchen outlet	0.0492	0.5826	0.1647	0.5202	0.9952	0.4645	0.3249	0.1092	0.5253
Lighting	0.0099	0.6907	0.1105	0.6384	0.9943	0.5975	0.1953	0.1006	0.6591
Washer dryer	0.0929	0.5648	0.1743	0.4304	0.9964	0.4302	0.1362	0.1149	0.5011

Table 3.5: Appliance Level Evaluation on REDD Dataset

Device	MLCDDL		MLKNN		RAKEL		AFAMA	MLCSAE	
	Error	F_1 score	Error	F_1 score	Error	F_1 score	Error	Error	F_1 score
Site Meter	0.0278	0.8306	0.0696	0.8096	0.9995	0.7072	0.1761	0.0432	0.8140
Air Conditioner	0.0188	0.5164	0.1381	0.5063	0.9995	0.5041	0.2263	0.0605	0.5062
Electric Furnace	0.0229	0.5092	0.0899	0.5005	0.9995	0.4568	0.1104	0.0519	0.5001
Socket	0.0333	0.5595	0.2696	0.5483	0.9995	0.5071	0.3172	0.0824	0.5413

Table 3.6: Appliance Level Evaluation on Pecan Dataset

In Tables 3.5 and 3.6, we show the performances at the appliance level. We have shown results only for layer 3 since it yields the best results. We see that the conclusions remain the same. We yield the best results in terms of all metrics. RAKEL yields by far the worst results. MLCSAE improves upon MLKNN in terms of energy error but not in terms of F1-score. AFAMAP is only better than RAKEL.

3.4 Discussion

Previous work on DDL [11] is unsupervised and solves the problem using a greedy learning approach. The problem with the greedy approach is that the shallower layers affect the deeper layers but not the other way around. Also, it can only be used for unsupervised feature extraction. In this chapter, we propose a supervised formulation for deep dictionary learning called LCDDL. The LCDDL tackles both the shortcomings of the ex-

isting method. This work proposes a joint approach to learn all the dictionaries simultaneously as a single optimization problem. Supervision is incorporated by adding a label consistency term on top of the unsupervised formulation. It is shown that the formulation can be generalized to perform both single label and multi-label classification without needing an external classifier. The comparison has been performed on existing benchmark techniques. In most of the cases, our proposed LCDDL approach excels over the rest. Proposed method is also applied on a practical application of NILM. There also it outperforms the existing methods.

The LCDDL method can also be used to perform regression operation. In many real time applications, the output contains continuous values. For example the application of blood pressure estimation from the ECG and PPG signals. The proposed method is general enough to perform single and multivariate regression. The difference will be in the definition of T . In case of single variate regression the variable T will be a vector. There will be one value corresponding to each sample. In multi variate regression problem on the other hand, T will contain more than one output values corresponding to each sample.

Chapter 4

Semi Supervised Deep Blind Compressed Sensing

In this work the objective is to classify biomedical signals from their compressive measurements. The problem arises when compressed sensing (CS) is used for energy efficient acquisition and transmission of such signals for wireless body area network. After reconstruction, the signal is analyzed via certain machine learning techniques. This work proposes to carry out joint reconstruction and analysis in a single framework; the reconstruction ability is obtained inherently from our formulation. We put forth a new technique called semi-supervised deep blind compressed sensing that combines the analytic power of deep learning with the reconstruction ability of compressed sensing. Experimental results on EEG and ACG classification shows that the proposed technique excels over the paradigm of CS reconstruction followed by deep learning classification. In Section 4.1 literature review is discussed. Section 4.2 presents the proposed formulation

followed by experimental results.

4.1 Literature Review

4.1.1 Blind Compressed Sensing

In CS, it is assumed that the sparsifying basis is given; for biomedical signals it is usually Gabor or wavelet. However it has been known that significantly better results are obtained by learning the basis adaptively from data. This leads to blind compressed sensing (BCS) [86] formulation.

For a series of acquired signals ' X ', BCS learns a dictionary / basis ' D ' so that the signals can be expressed by sparse coefficients ' Z '. This is expressed as follows -

$$X = DZ \quad (4.1)$$

In the given scenario, the signals are compressed. Incorporating the BCS model into the compression scenario 4.1 leads to,

$$Y = AX = ADZ \quad (4.2)$$

Here A is a random projection / compression matrix ; Y is the matrix formed by stacking the measured compressed data as columns.

The basis and the coefficients are learnt by solving the following optimization problem -

$$\min_{D,Z} \|Y - ADZ\|_F^2 + \lambda \|Z\|_1 + \mu \|D\|_2 \quad (4.3)$$

In essence BCS combines compressed sensing with dictionary learning; it reconstructs the signal by learning a basis during the process. In the past it has been used for feature extraction from compressive samples using the concept of doubly sparse dictionary learning [87].

BCS / dictionary learning is based on the concept of matrix factorization; it essentially factors the data matrix into a dictionary matrix D and a coefficient matrix Z . With the success of deep learning techniques in various fields of signal analysis, recent studies proposed 'deep matrix factorization' [88]. Here the data matrix is factored into multiple levels of dictionaries and a coefficient matrix. For three levels it is shown as,

$$X = D_1 D_2 D_3 Z \quad (4.4)$$

In this formulation, the non-linearity between the different layers of dictionaries is imposed by non-negativity constraints. In a related work [11] on deep dictionary learning, the non-linearity is more explicitly incorporated so as to handle different types of activations. Incorporating the deep matrix factorization into the BCS framework, leading to -

$$Y = AD_1 D_2 D_3 Z \quad (4.5)$$

This is an unsupervised formulation. In the next section supervised version of deep BCS is discussed in detail.

4.2 Proposed Formulation

Deep learning is successful when the full data is available. When the data is partially sampled or is compressed, deep learning is unable to handle it. The issues is not confined to deep learning; in fact, most off-the-shelf techniques in machine learning are unable to perform when the data is only partially observed [89].

Given the problem of analyzing the signals from compressed measurements, there can be two options. In the first one, we can reconstruct the biomedical signals and then apply deep learning / machine learning on the reconstructed signals. The problem with this approach is that the compressed sensing (CS) reconstruction process introduces artifacts, and these artifacts affect the learning process.

The second option is to directly analyze the signals from the compressed / partially observed domain. This is the approach followed in this work. The unsupervised version of deep BCS is given by-

$$\min_{D_1, D_2, D_3, Z} \|Y - AD_1D_2D_3Z\|_F^2 + \lambda\|Z\|_1 \quad (4.6)$$

As discussed in the previous chapter, supervision is incorporated by adding a label consistency term to the equation 4.6.

$$\min_{D_1, D_2, D_3, Z, W} \|Y - AD_1D_2D_3Z\|_F^2 + \lambda\|Z\|_1 + \mu\|T - WZ\|_F^2 \quad (4.7)$$

Here T corresponds to the binary class labels and W the linear classification map. The parameter μ controls the relative importance between the data consistency term and the classification term.

In this work, we want to go a step further and show how 4.7 can be modified for semi-supervised formulation, i.e. when there is a combination of labeled and unlabeled data.

$$\min_{D_1, D_2, D_3, Z, W} \|[Y_U | Y_L] - AD_1 D_2 D_3 [Z_U | Z_L]\|_F^2 + \lambda \|[Z_U | Z_L]\|_1 + \mu \|T - W Z_L\|_F^2 \quad (4.8)$$

Here the subscripts U and L stand for unlabeled and labeled respectively. Basically, the unsupervised portion of the formulation is applicable to both labeled and unlabeled data whereas the supervised portion is applied only to the labeled data.

During the testing stage, given the test measurement y_{test} , the task is to assign a class label to it. For that, the first task is to generate the coefficient z_{test} . This is achieved by using the dictionaries learn) during the training phase as basis.

$$z_{test} \|y_{test} - AD_1 D_2 D_3 z_{test}\|_F^2 + \lambda \|z_{test}\|_1 \quad (4.9)$$

Once the coefficient / feature is generated the learnt linear classifier (W) is used to assign a class label. The test feature is simply multiplied by W . The output is never a binary label, but we assign the position of the highest magnitude coefficient in $W z_{test}$ as the class label.

First we express $X = D_1 D_2 D_3 Z$; where X is the complete signal. With this substitution, 4.8 can be expressed as,

$$\min_{D_1, D_2, D_3, Z, W, X} \|[Y_U|Y_L] - A[X_U|X_L]\|_F^2 + \lambda\|[Z_U|Z_L]\|_1 + \mu\|T - WZ_L\|_F^2$$

such that $X = D_1 D_2 D_3 Z$

(4.10)

Here $X = [X_U|X_L]$.

Ideally one would like to formulate the Lagrangian for the newly introduced constraint. However, the Lagrangian would enforce equality at each step. This is not required in practice, one would only need to enforce the constraint at convergence. This is the reason, we follow the augmented Lagrangian approach instead [90]. This is given by-

$$\min_{D_1, D_2, D_3, Z, W, X} \|[Y_U|Y_L] - A[X_U|X_L]\|_F^2 + \lambda\|[Z_U|Z_L]\|_1 + \mu\|T - WZ_L\|_F^2 +$$

$\eta\|X - D_1 D_2 D_3 Z\|_F^2$

(4.11)

In the next step we substitute introduce another proxy variable: $Z_1 = D_2 D_3 Z$.

The corresponding augmented Lagrangian is expressed as,

$$\min_{D_1, D_2, D_3, Z, W, X, Z_1} \|[Y_U|Y_L] - A[X_U|X_L]\|_F^2 + \lambda\|[Z_U|Z_L]\|_1 + \mu\|T - WZ_L\|_F^2 +$$

$\eta\|X - D_1 Z_1\|_F^2 + \eta_1\|Z_1 - D_2 D_3 Z\|_F^2$

(4.12)

In the final step, we substitute, $Z_2 = D_3 Z$. This leads to,

$$\begin{aligned} \min_{D_1, D_2, D_3, Z, W, X, Z_1, Z_2} & \| [Y_U | Y_L] - A[X_U | X_L] \|_F^2 + \lambda \| [Z_U | Z_L] \|_1 + \mu \| T - W Z_L \|_F^2 + \\ & \eta \| X - D_1 Z_1 \|_F^2 + \eta_1 \| Z_1 - D_2 Z_2 \|_F^2 + \eta_2 \| Z_2 - D_3 Z \|_F^2 \end{aligned} \quad (4.13)$$

The final form 4.13 can be solved by using the technique of alternating direction method of multipliers (ADMM) [47]. The basic idea in ADMM is to update each variable by assuming the others to be constant. This leads to the following sub-problems.

- **P1** : $\min_{D_1} \| X - D_1 Z_1 \|_F^2$
- **P2** : $\min_{D_2} \| Z_1 - D_2 Z_2 \|_F^2$
- **P3** : $\min_{D_3} \| Z_2 - D_3 Z \|_F^2$
- **P4** : $\min_Z \| Z_2 - D_3 Z \|_F^2 + \| Z \|_1$
- **P5** : $\min_W \| T - W Z_L \|_F^2$
- **P6** : $\min_X \| Y - AX \|_F^2 + \| X - D_1 Z_1 \|_F^2$
- **P7** : $\min_{Z_1} \eta \| X - D_1 Z_1 \|_F^2 + \eta_1 \| Z_1 - D_2 Z_2 \|_F^2$
- **P8** : $\min_{Z_2} \eta_1 \| Z_1 - D_2 Z_2 \|_F^2 + \eta_2 \| Z_2 - D_3 Z \|_F^2$

Apart from sub-problem P_4 , all others are simple least squares problems having a closed form solution in the form of pseudoinverse. P_4 is an l_1 -minimization problem. This can be solved using Iterative Soft Thresholding Algorithm (ISTA) [13].

This concludes the basic training algorithm. But there needs to be some ad-

ditional checks. In order to prevent degenerate solutions where the dictionaries are of high values and the coefficients of small values or vice versa, we need to normalize the columns of the dictionaries after every iteration.

4.2.1 Reconstruction Ability

The reconstruction can be done both for training and testing. Once the learning is over, the original data can be recovered simply by using the formula $X = D_1 D_2 D_3 Z$.

Similarly, once the testing features z_{test} are obtained, the full signal can be recovered via $x_{test} = D_1 D_2 D_3 z_{test}$.

Thus we see that by using our formulation we can recover signals in the physical domain for both training and testing. This is a by-product of our formulation.

4.2.2 Contributions

There are two fundamental contributions of this work.

- This is the only deep learning tool that can learn from compressive samples. No other deep framework (stacked autoencoder, deep belief network or convolutional neural network) can infer from compressed or missing data directly (without reconstruction).
- The second contribution is a by-product of our formulation. Ours is the first work that proposes a technique for matrix completion (from under-sampled projections) using a deep learning technique.

4.3 Experimental Results

4.3.1 Seizure Detection from EEG

A publicly available EEG dataset, made available by the University of Bonn [91] is used in this work. The EEG database consists of five sets (A-E). Each set contains 100 single-channel EEG segments, each with a duration of 23.6 s. Sets A and B have been recorded using the standard international 10-20 system for surface EEG recording. Five healthy volunteers were participated in these tests with eyes open (A) and eyes closed (B). For sets C, D, and E, five epileptic patients were selected for presurgical evaluation of epilepsy by using intracranial electrodes. Depth electrodes were implanted symmetrically to record EEG from the epileptogenic zone (D) and from hippocampal formation of the opposite hemisphere of the brain (C). Segments of set E were taken from contacts of all electrodes. In sets C and D, segments contain interictal intervals while seizure activities occur in set E. Each epoch was sampled at 173.61 Hz resulting in a total of 4096 samples.

There are not many studies in deep learning based seizure classification to compare our method with. In [92, 93], it was found that DBN with logistic regression yields good detection results; it was better than traditional classifiers like nearest neighbor and support vector machine. In [94] a convolutional neural net (CNN) with long short term memory (LSTM) units yielded good detection rate. The paper[95] used stacked autoencoder

(SAE) with logistic regression for classification. In this work, our goal is to classify the signals from compressive measurements; sparse binary projection matrices are used for the purpose. Traditional deep learning is not able to learn (optimally) from compressed measurements. Therefore while comparing with other deep learning techniques, we first recover the original signal from the compressive measurements via a CS; in this work we use [4].

With the advent of compressed sensing in the past decade, researchers in neural networks explored their efficacy in compressed measurement domain [96, 97, 98]. However these studies are not directly related to our work. In [96], a neural network is used for regression from compressed domain. In [97, 98] they propose a neural network solution for reconstructing the signal from compressed measurements. A very recent work [99] addresses the problem of classification from compressed domain; however it solves the problem piecemeal. First a standard technique is used for recovering the data in the physical domain and then a standard deep learning tool is applied on the recovered data. This is the same procedure we use in our study for comparison.

It is well known that deep learning (DBN [93], CNN-LSTM [95], SAE [96]) benefits from pre-training with unlabeled data; these pre-trained models are fine-tune with the actual training data from [91]. Therefore we use datasets from the BCI competitions II and III; these datasets are resampled to fit the University of Bonn dataset. Our proposed method does not require pre-

training; it uses both labeled and unlabeled data in one single formulation. We have experimented with four architectures of depths 1, 2, 3, and 4. The configurations are 200, 200-100, 200-100-50 and 200-100-50-25 respectively. The practice of reducing the number of nodes by half in subsequent layers is common in deep learning. The value of $\mu = 1$ has been used to give equal importance to the reconstruction and regression costs. The hyper-parameters η , η_1 and η_2 have been fixed at unity. This is because, they correspond to different layers of the deep learning architecture. Since there is no reason to favor one layer over the other, we keep them to be unity.

-	Details	Proposed 1 layer	Proposed 2 layer	Proposed 3 layer	Proposed 4 layer
A	Eyes open	89	91	92	90
B	Eyes closed	92	95	97	95
C	Inter-ictal (epileptic focus)	91	95	96	94
D	Inter-ictal (Hipocam. region)	92	95	96	94
E	Ictal state	94	96	98	96

Table 4.1: Effect of Depth at 4:1 Compression on Class-wise Accuracy

-	Details	Proposed 1 layer	Proposed 2 layer	Proposed 3 layer	Proposed 4 layer
A	Eyes open	91	93	94	93
B	Eyes closed	98	100	100	100
C	Inter-ictal (epileptic focus)	95	97	98	96
D	Inter-ictal (Hipocam. region)	96	97	99	97
E	Ictal state	97	100	100	100

Table 4.2: Effect of Depth at 2:1 Compression on Class-wise Accuracy

In the first set of experiments we show the influence of depth on results; these are in Tables 4.1 (4:1 compression) and 4.2 (2:1 compression). It is known in deep learning that as we go deeper, more abstract and robust representation is learnt this helps in inference and analysis. But there is a trade-off between depth and over-fitting; as one goes deeper more and

more parameters need to be learnt. With limited training data, the deeper models tend to overfit. This is the reason we see that when we go from levels 1 to 3, there is a gradual increase in accuracy but in the 4th layer there is a dip.

-	Details	DBN	CNN-LSTM	SAE	Proposed
A	Eyes open	85	85	84	92
B	Eyes closed	94	94	94	97
C	Inter-ictal (epileptic focus)	90	92	89	96
D	Inter-ictal (Hippocam. region)	91	91	90	97
E	Ictal state	88	90	89	98

Table 4.3: EEG Classification Results (%age) at 4:1 Compression

-	Details	DBN	CNN-LSTM	SAE	Proposed
A	Eyes open	89	89	89	94
B	Eyes closed	97	98	96	100
C	Inter-ictal (epileptic focus)	92	95	92	98
D	Inter-ictal (Hippocam. region)	94	95	93	99
E	Ictal state	92	94	92	100

Table 4.4: EEG Classification Results (%age) at 2:1 Compression

The configurations of the other techniques have been taken from the corresponding papers; we skip repeating them for the sake of brevity. These methods (DBN, CNN-LSTM and SAE) uses 9 hand-crafted features - area under wave, normalized decay, line length, mean energy, average peak amplitude, average valley amplitude, peak variation, and root mean square. These features are extracted from all the channels and used as input to the aforesaid deep learning classifiers.

The comparative experimental results with for 4:1 and 2:1 compression is shown in Tables 4.3 and 4.4 respectively. The corresponding confusion

DBN						
Actual						
	A	B	C	D	E	
A	85	0	4	2	7	
B	3	94	2	2	0	
C	4	2	90	0	3	
D	1	0	1	91	2	
E	7	4	3	6	88	

CNN-LSTM						
Actual						
	A	B	C	D	E	
A	85	0	2	3	6	
B	4	94	1	0	1	
C	3	3	92	3	1	
D	2	0	3	91	2	
E	6	3	2	3	90	

SAE						
Actual						
	A	B	C	D	E	
A	89	0	2	2	4	
B	3	96	2	0	1	
C	2	0	92	3	0	
D	1	1	2	93	3	
E	5	3	2	2	92	

Proposed						
Actual						
	A	B	C	D	E	
A	92	0	0	2	2	
B	1	97	1	0	0	
C	3	0	96	2	0	
D	0	0	1	96	0	
E	4	3	2	0	98	

DBN						
Actual						
	A	B	C	D	E	
A	89	0	3	2	3	
B	2	97	1	0	2	
C	3	1	92	2	1	
D	2	0	2	94	2	
E	4	2	2	2	92	

CNN-LSTM						
Actual						
	A	B	C	D	E	
A	89	0	2	1	3	
B	0	98	0	1	2	
C	4	0	95	1	0	
D	3	1	0	95	1	
E	4	1	3	2	94	

SAE						
Actual						
	A	B	C	D	E	
A	89	1	3	2	4	
B	2	96	1	1	3	
C	3	1	92	1	0	
D	2	0	0	93	1	
E	4	2	4	3	92	

Proposed						
Actual						
	A	B	C	D	E	
A	94	0	1	0	0	
B	0	100	0	0	0	
C	2	0	98	0	0	
D	2	0	0	99	0	
E	2	0	1	1	100	

Figure 4.1: Confusion Matrices for EEG. Top - For 4:1 Compression. Bottom - For 2:1 Compression. For every matrix, the columns are the actual classes and the rows are the predicted classes

matrices are also shown in Figure 4.1. Our method, always yields the best results. The results from CNN-LSTM combination are a distant second. DBN and SAE almost yield the same performance. The CNN-LSTM performs slightly better than DBN and SAE because of its ability to capture dynamical attributes. As expected, the results with higher compression ratio is lower than lower compression ratio.

It must be noted that these results cannot be compared with the prior published work because, the prior works used the complete signal and where as this one uses reconstructed values. The reconstruction introduces error which reduces the performance. It is interesting to note other methods uses hand-crafted features and we use raw compressive measurements; yet we perform significantly better than the other methods.

As discussed before, our method can inherently reconstruct the signals from the compressive measurements. Therefore we compare the recon-

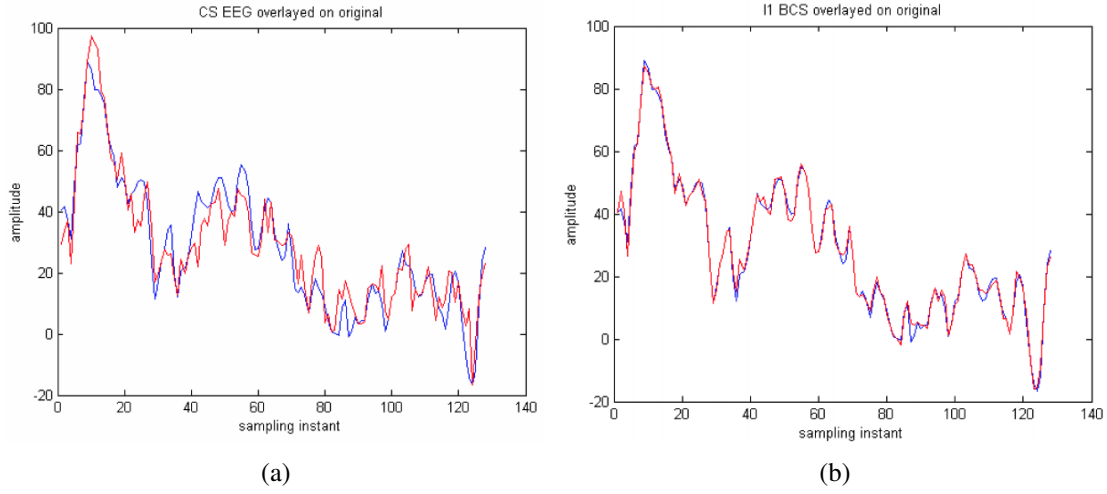


Figure 4.2: EEG reconstruction. (a) CS [4] Reconstruction. (b) Proposed.

struction performance of our proposed method with CS [4]. For visual evaluation a sample is shown in Figure 4.2 One can see that our proposed method does a much better reconstruction than standard CS. For numerical corroboration we show the mean and the standard deviation of the normalized mean squared error in the following table.

Technique	2:1 Compression	4:1 Compression
Compressed Sensing	0.112 ± 0.062	0.240 ± 0.084
Proposed - training	0.056 ± 0.008	0.102 ± 0.014
Proposed - testing	0.056 ± 0.006	0.102 ± 0.011

Table 4.5: EEG Reconstruction Results

The reconstruction results (Table 4.5) indicate that for both the training and the testing stage, we perform significantly better than CS. The results for both training and testing are equally accurate, but the testing stage has lesser variability. This is likely because the dictionaries are pre-learned.

4.3.2 Arrhythmia Classification from ECG

In this study, five types of beat classes of arrhythmia as recommended by Association for Advancement of Medical Instrumentation (AAMI) were analyzed from electrocardiogram (ECG) signals namely: non-ectopic beats, supra-ventricular ectopic beats, ventricular ectopic beats, fusion beats and unclassifiable and paced beats. The classification experiments are carried out on the MIT-BIH Arrhythmia dataset from www.physionet.org. It is well known in deep learning that 'more the merrier'. However in real-life, supervised samples are few; but it is easy to have a large number of unsupervised samples. It has been shown in [100] that augmenting the supervised learning problem (MIT-BIH) with unsupervised data from the ST-T dataset (cardiac ischemia www.physionet.org) improves the reconstruction and classification results. No class information from the second dataset is used; it is only used for semi-supervised learning. We follow the same protocol in this paper. The MIT-BIH Arrhythmia database contains 48 half hour recordings of two channel ambulatory ECG, obtained from 47 subjects in the year 1975 and 1979 by the Beth-Israel Hospital Arrhythmia Laboratory at Boston. Twenty-four hour ambulatory ECG recordings were collected from a mixed population of size 4000 having inpatients (around 60%) and outpatients (around 40%). The recordings were digitized at 360 samples per second per channel with 11-bit resolution over a 10 mV range. Two or more cardiologists independently annotated each record; consensus was made to obtain the computer-readable reference annotations for each

beat.

The European society of cardiology has provided a standard ST-T database consisting of 90 annotated samples of ambulatory ECG recordings from 79 subjects having myocardial ischemia disease. The subjects were 70 men aged from 30 to 84 years, and some women aged from 55 to 71 years. Additional selection criteria were established in order to obtain a representative selection of ECG abnormalities in the database, including baseline ST segment displacement resulting from conditions such as hypertension, ventricular dyskinesia, and effects of medication. Each record is of 2 hours duration and contains two signals. Each is sampled at 250 samples per second with 12-bit resolution over a nominal 20 mV input range.

As a pre-processing step the MIT-BIH dataset is down-sampled to 250 Hz from its native 360 Hz; this is to ensure parity between the two datasets. For classification experiments, the MIT-BIH protocol is converted to the AAMI / ANSI standard. This leads to 5 classes - Non ectopic beat (N), Supra-ventricular ectopic beats (S), Ventricular ectopic beats (V), Fusion beat (F) and Unknown beat (Q). Owing to the relative sparsity of samples in the F and Q class they are merged with V; this is following the AAMI2 protocol proposed in [101].

For the experimental protocol we follow [101]; this is repeatable protocol (also followed in [100]). The division into test set and training set is shown in Table 4.6.

The prior study [100] showed that stacked autoencoders beat other shallow

Dataset	N	S	V	F	Q	Total	Rec
Train	45844	943	3788	415	8	50998	22
Test	44238	1836	3221	388	7	49690	22
Total	90082	2779	7009	803	15	100688	44

Table 4.6: Train and Test Set Details

classifiers like optimum path forest, support vector machine, probabilistic neural network and extreme learning machine for the said task. Therefore we do not compare with the such shallow methods any more.

A recent study [102], proposed deep neural networks learned by stacked autoencoder (SAE) and deep belief network (DBN). We compare against [102]. For this task, the SAE and the DBN are pre-trained with unlabeled data (ST-T) and fine-tuned with the labeled samples (MIT-BIH). Note that neither [102] nor [100] can act directly on compressed measurements. The method proposed in [100] can only operate when the samples are missing, but not when they are compressed. In this work we assume that the data has been compressed in situ by a sparse binary matrix. Therefore before applying [102] we recover the samples from the compressive measurement via compressed sensing. As done in the previous set of experiments, we study the variation in accuracy with the number of levels for the proposed method. We use four depths - 1 layer (250 atoms), 2 layer (250-125 atoms), 3 layer (250-125-125 atoms) and 4 layers (250-125-125-63 atoms). These configurations gave the best results for each depth. The results are shown in Tables 4.7 and 4.8.

The results show the common trend envisaged before. From one layer to three layers the results improve. But above 3 layers, the results do not; it

Classifier	Acc.	F		S		V	
		Sens.	Spec.	Sens.	Spec.	Sens.	Spec.
1 layer	89.9	80.6	55.9	11.8	81.5	75.6	88.1
2 layer	93.6	87.8	60.2	18.9	89.8	84.1	92.4
3 layer	98.0	100	67.2	23.0	100	90.1	100
4 layer	98.0	100	67.1	23.0	100	89.7	98.4

Table 4.7: ECG Classification Results (%age) on 4:1 Compression

Classifier	Acc.	F		S		V	
		Sens.	Spec.	Sens.	Spec.	Sens.	Spec.
1 layer	94.9	95.0	66.4	22.9	88.4	85.9	93.8
2 layer	96.8	96.2	68.0	26.2	92.1	88.5	97.2
3 layer	100	100	70.5	28.0	100	94.1	100
4 layer	100	100	70.2	28.0	100	93.7	100

Table 4.8: ECG Classification Results (%age) on 2:1 Compression

either remains the same or dips slightly.

Next we carry out comparative experiments with other techniques. The configuration for the SAE and the DBNs are obtained from [102], since they are supposed to yield the best results. For our proposed method, we use a three layer architecture. The number of basis are 250-125-125. The value of $\lambda = 0.01$ and $\mu = 1$ is used as before.

Experimental results are reported on 4:1 (Table 4.9) and 2:1 (Table 4.10) compression. Classification Accuracy (Acc.) is the most important measure for performance; but it is a standard practice to report sensitivity (Sens.) and specificity (Spec.); the standard definitions apply for all the metrics.

Classifier	Acc.	F		S		V	
		Sens.	Spec.	Sens.	Spec.	Sens.	Spec.
DBN	59.7	60.2	25.9	0	51.3	16.8	28.1
SAE	85.6	87.3	33.9	0	82.6	39.8	81.6
Proposed	98.0	100	67.2	23.0	100	90.1	100

Table 4.9: ECG Classification Results (%age) on 4:1 Compression

Classifier	Acc.	F		S		V	
		Sens.	Spec.	Sens.	Spec.	Sens.	Spec.
DBN	94.8	95.7	66.5	23.9	100	86.9	100
SAE	93.8	96.9	62.8	20.8	100	83.2	100
Proposed	100	100	70.5	28.0	100	94.1	100

Table 4.10: ECG Classification Results (%age) on 2:1 Compression

Technique	2:1 compression	4:1 compression
CS	0.082 $\pm$ 0.036	0.118 $\pm$ 0.098
Proposed - training	0.037 $\pm$ 0.027	0.085 $\pm$ 0.061
Proposed - testing	0.037 $\pm$ 0.026	0.085 $\pm$ 0.060

Table 4.11: ECG Reconstruction Results

The results from Tables 4.9 and 4.10 show that our proposed technique yields better results than CS reconstruction followed by deep neural network classification [92]. This points to the fact that, there is indeed benefit in learning the reconstruction and analysis operations in a joint fashion.

Finally we compare the reconstruction ability of our proposed technique (Table 4.11). For comparison we employ the compressed sensing technique proposed in [4]. The results are shown in the table 4.11. We can see that our proposed method yields significantly superior results compared to compressed sensing. Between the training and the testing stages, the testing stage yields better results. This has also been seen for EEG reconstruction.

For visual evaluation we have shown one sample reconstructed by our proposed technique and CS in Figure 4.3. One clearly sees that our proposed technique follows the original signal perfectly, but CS introduces lot of artifacts. This is the reason why the performance of deep learning methods deteriorate. The artifacts introduced by CS reconstruction hampers their

performance.

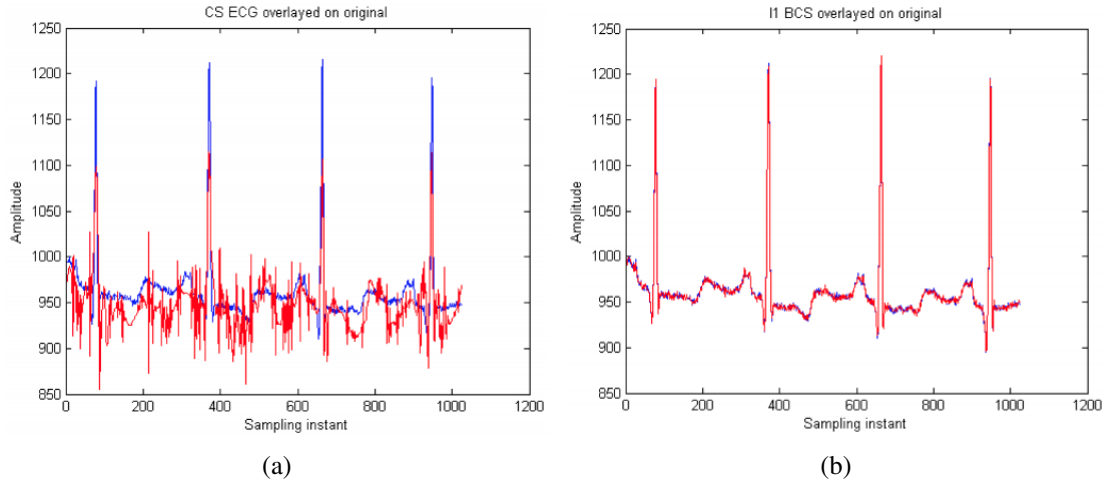


Figure 4.3: ECG reconstruction. (a) CS [4] Reconstruction. (b) Proposed.

4.4 Discussion

In this work our objective is to classify biomedical signals from their compressive measurements. In the past the process of analyzing the compressed signal was a two-stage process. This work combines the two stages (reconstruction and classification) in a single formulation. By eliminating a separate reconstruction stage, we minimize the errors and artifacts and hence improve the results. A by-product of our formulation is that it can inherently reconstruct the signals. Results show that our method excels over the other generic methodologies in both reconstruction and classification.

This work garners the reconstruction ability of compressed sensing with the power of deep learning. Although shown here for EEG and ECG signals the methodology is generic and is applicable to a wide variety of problems. For example, it can be applied for object recognition problems where parts

of the object may be occluded (can be assumed as missing data). It can be used in hyperspectral image analysis where there are missing values owing to the malfunctioning of the sensors.

In fact the methodology can also work on fully sampled problems as well. In that case instead of the projection matrix, we need to use an identity operator.

Chapter 5

Siamese Deep Dictionary Learning

In this chapter we propose a siamese network architecture with deep dictionary learning with the objective of classifying two inputs as a match or mismatch (binary classification). In section 5.1 literature review is discussed. Section 5.2 presents the proposed algorithm followed by experimental results. The experimental effectiveness is demonstrated on five different image databases pertaining to two classification problems: face verification and kinship verification.

5.1 Motivation

Consider a problem where the task is to verify if a person is trying to impersonate another by disguise or the task is to determine the relationship between two person. Or a simple problem of verification; which is to tell whether two people are same or not. All such problems are binary classification problems, given a pair inputs.

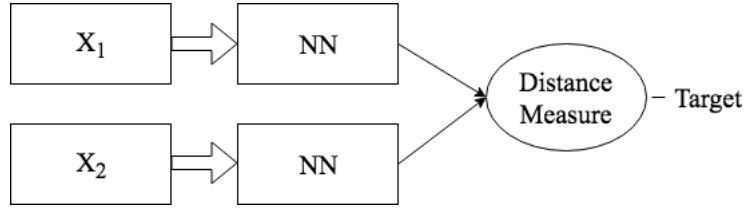


Figure 5.1: Typical Siamese Neural Network

In recent years, with the advent of deep learning, siamese neural networks [103],[104] have been used to address this problem. As shown in Figure 5.1, in a typical siamese network, there are two parallel processing pipelines for two inputs for example two face images in a face verification problem. Here X_1 and X_2 denote the two inputs to the network and NN denote the neural network (CNN in [103]) forming the two pipelines of the siamese network. Input representation is generated by neural networks. A distance measure is computed over the representations of both the inputs. And the error is backpropagated to learn the network weights. Initial studies constrained the two pipelines to have shared weights, but DeepFace demonstrated that non-shared weights can also be used to effectively train siamese network [105].

5.2 Proposed Formulation

Considering a problem of disguise detection, there are two sets of images: gallery and probe. Gallery of images consisting of the genuine people, and the probe is an image - can be either an imposter or genuine. The task is to identify if the probe is genuine or imposter. In Figure 5.2 we show some examples of Siamese Deep Dictionary Learning (SDDL) for disguise

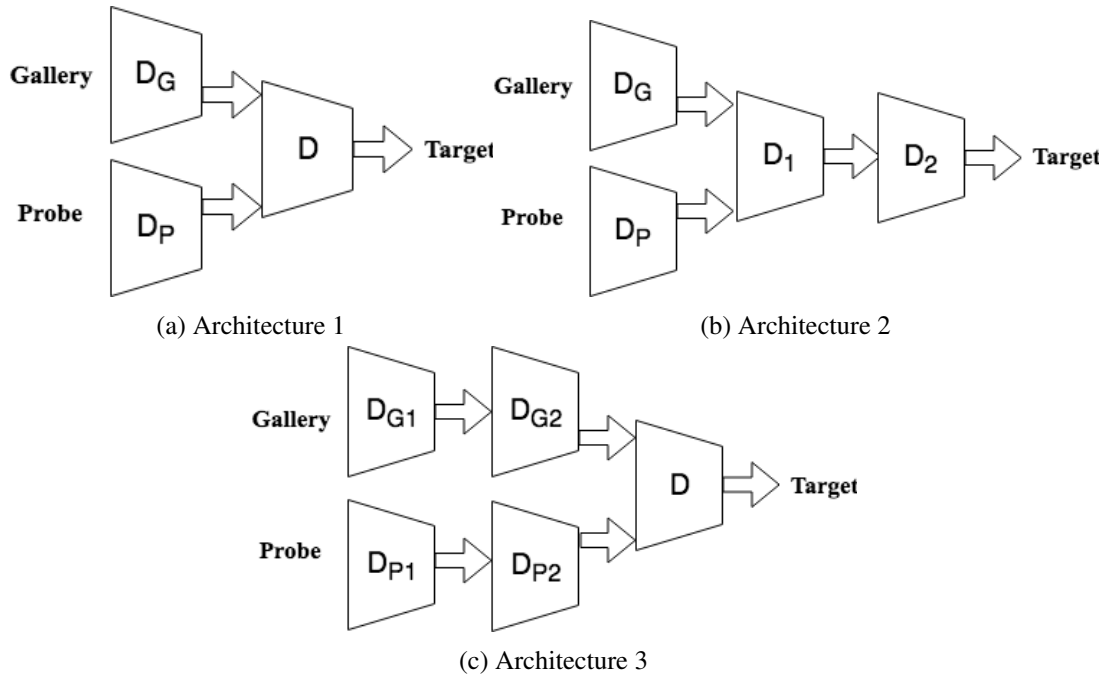


Figure 5.2: Siamease Deep Dictionary Learning

detection. In Figure 5.2a we have one level of dictionary for processing the individual probe and gallery image and another level for fusing the two. In Figure 5.2b, a single layer of dictionary processes the probe and gallery image, and two layers for fusion. In the final architecture Figure 5.2c there are two layers for processing the probes and gallery images, and one layer for fusion.

In general, there can be any number of layers for processing the probe and gallery and any number of layers for fusing the two. For the sake of generality let us assume that there are N layers for each channel (gallery / probe) and M layers for fusion. We want to keep the network flexible so we do not tie the weights for the gallery and the probes. In such a situation,

the deep dictionary learning network for gallery will be represented as,

$$G = D_{G1}\phi(D_{G2}\phi(\dots\phi(D_{GN}Z_G))) \quad (5.1)$$

Similarly the network for the probe will be represented as,

$$P = D_{P1}\phi(D_{P2}\phi(\dots\phi(D_{PN}Z_P))) \quad (5.2)$$

Here G and P denote the gallery and the probe image respectively. For the fusion layers, the input is the concatenated Z_P and Z_G , followed by M layers of dictionaries. This is represented as,

$$\begin{bmatrix} Z_G \\ Z_P \end{bmatrix} = D_1\phi(D_2\phi(\dots\phi(D_M Z))) \quad (5.3)$$

The output (final layer of coefficients - Z) is mapped to the targets in a manner similar to label consistent KSVD [61], i.e.

$$t = \phi(wZ) \quad (5.4)$$

Here t is a row vector, having $+1$'s where there is a match and -1 's elsewhere. The weights w , maps the coefficients Z to the targets and needs to be learnt.

Putting 5.1 to 5.4 as a single optimization problem leads to the following

formulation.

$$\begin{aligned}
& \min_{D'_G s, D'_P s, D' s, Z_G, Z_P, Z, w} ||G - D_{G1}\phi(D_{G2}\phi(\dots\phi(D_{GN}Z_G)))||_F^2 \\
& \quad + ||P - D_{P1}\phi(D_{P2}\phi(\dots\phi(D_{PN}Z_P)))||_F^2 \\
& \quad + || \begin{bmatrix} Z_G \\ Z_P \end{bmatrix} - D_1\phi(D_2\phi(\dots\phi(D_M Z)))||_F^2 + ||t - \phi(wZ)||_F^2
\end{aligned} \tag{5.5}$$

5.2.1 Training

For solving the optimization problem given by equation 5.5, we follow the variable splitting approach [14]. We show it for the gallery first, the rest will follow similarly. We introduce proxies: $Z_{G1} = \phi(D_{G2}\phi(\dots\phi(D_{GN}Z_G)))$ for the first layer, $Z_{G2} = \phi(D_{G3}\dots\phi(D_{GN}Z_G))$ for the second layer and so on for all the layers till the penultimate layer. This allows expressing $||G - D_{G1}\phi(D_{G2}\phi(\dots\phi(D_{GN}Z_G)))||_F^2$ in 5.6 in terms of its Augmented Lagrangian [15].

$$||G - D_{G1}Z_{G1}||_F^2 + ||Z_{G1} - \phi(D_{G2}Z_{G2})||_F^2 + \dots + ||Z_{GN-1} - \phi(D_{GN}Z_G)||_F^2 \tag{5.6}$$

In general, the multiplicative terms μ' s need to be tuned. But for our case, they have a special meaning, they indicate the relative weight of each layer. Since there is no reason to give any layer more importance than the other, we argue that all of them should be unity.

In a similar fashion, we can obtain the augmented Lagrangian for the probe:

$$\|P - D_{P1}Z_{P1}\|_F^2 + \|Z_{P1} - \phi(D_{P2}Z_{P2})\|_F^2 + \dots + \|Z_{PN-1} - \phi(D_{PN}Z_P)\|_F^2 \quad (5.7)$$

Using a similar variable splitting, $Z_1 = \phi(D_2\phi(\dots\phi(D_MZ)))$, $Z_2 = \phi(D_3\dots\phi(D_MZ))$ etc. for the fusion layers, we have the corresponding augmented Lagrangian,

$$\left\| \begin{bmatrix} Z_G \\ Z_P \end{bmatrix} - D_1Z_1 \right\|_F^2 + \|Z_1 - \phi(D_2Z_2)\|_F^2 + \dots + \|Z_{M-1} - \phi(D_MZ)\|_F^2 \quad (5.8)$$

Combining the augmented Lagrangian's 5.6, 5.7 and 5.8 into the basic formulation 5.5 leads to the following,

$$\begin{aligned} & \min_{D'_G, s, D'_P, s, D'_s, Z_G, Z_P, Z, w} \|G - D_{G1}Z_{G1}\|_F^2 + \|Z_{G1} - \phi(D_{G2}Z_{G2})\|_F^2 + \dots + \\ & \|Z_{GN-1} - D_{GN}Z_{GN}\|_F^2 + \|P - D_{P1}Z_{P1}\|_F^2 + \|Z_{P1} - \phi(D_{P2}Z_{P2})\|_F^2 + \dots + \\ & \|Z_{PN-1} - D_{PN}Z_{PN}\|_F^2 + \left\| \begin{bmatrix} Z_G \\ Z_P \end{bmatrix} - D_1Z_1 \right\|_F^2 + \|Z_1 - \phi(D_2Z_2)\|_F^2 + \dots + \\ & \|Z_{M-1} - \phi(D_MZ)\|_F^2 \end{aligned} \quad (5.9)$$

The augmented Lagrangian 5.9 can be solved efficiently using the alternating direction method of multipliers (ADMM) approach [16]. The idea is to update each of the variables separately assuming the others to be constant. Using ADMM, 5.9 can be segregated into the following sub-problems.

$$P_1 : \min_{D_{G1}} \|G - D_{G1}Z_{G1}\|_F^2$$

$$P_2 : \min_{D_{G2}} \|Z_{G1} - \phi(D_{G2}Z_{G2})\|_F^2 \equiv \|\phi^{-1}(Z_{G1}) - D_{G2}Z_{G2}\|_F^2$$

.

.

.

$$P_N : \min_{D_{GN}} \|Z_{GN-1} - \phi(D_{GN}Z_G)\|_F^2 \equiv \|\phi^{-1}(Z_{GN-1}) - D_{GN}Z_G\|_F^2$$

$$P_{N+1} : \min_{D_{P1}} \|P - D_{P1}Z_{P1}\|_F^2$$

$$P_{N+2} : \min_{D_{P2}} \|Z_{P1} - \phi(D_{P2}Z_{P2})\|_F^2 \equiv \|\phi^{-1}(Z_{P1}) - D_{P2}Z_{P2}\|_F^2$$

.

.

.

$$P_{2N} : \min_{D_{PN}} \|Z_{PN-1} - \phi(D_{PN}Z_P)\|_F^2 \equiv \|\phi^{-1}(Z_{PN-1}) - D_{PN}Z_P\|_F^2$$

$$P_{2N+1} : \min_{Z_{G1}} \|G - D_{G1}Z_{G1}\|_F^2 + \|Z_{G1} - \phi(D_{G2}Z_{G2})\|_F^2$$

$$P_{2N+2} : \min_{D_{G2}} \|Z_{G1} - \phi(D_{G2}Z_{G2})\|_F^2 + \|Z_{G2} - \phi(D_{G3}Z_{G3})\|_F^2$$

$$\equiv \min_{D_{G2}} \|\phi^{-1}(Z_{G1}) - D_{G2}Z_{G2}\|_F^2 + \|Z_{G2} - \phi(D_{G3}Z_{G3})\|_F^2$$

.

.

.

$$P_{3N} : \min_{Z_G} \|Z_{GN-1} - \phi(D_{GN}Z_G)\|_F^2 + \left\| \begin{bmatrix} Z_G \\ Z_P \end{bmatrix} - D_1 Z_1 \right\|_F^2 \equiv \min_{Z_G} \|\phi^{-1}(Z_{GN-1}) -$$

$$D_{GN}Z_G\|_F^2 + \left\| \begin{bmatrix} Z_G \\ Z_P \end{bmatrix} - D_1 Z_1 \right\|_F^2$$

$$P_{3N+1} : \min_{Z_{P1}} \|P - D_{P1}Z_{P1}\|_F^2 + \|Z_{P1} - \phi(D_{P2}Z_{P2})\|_F^2$$

$$P_{3N+2} : \min_{D_{P2}} \|Z_{P1} - \phi(D_{P2}Z_{P2})\|_F^2 + \|Z_{P2} - \phi(D_{P3}Z_{P3})\|_F^2$$

$$\equiv \min_{D_{P2}} \|\phi^{-1}(Z_{P1}) - D_{P2}Z_{P2}\|_F^2 + \|Z_{P2} - \phi(D_{P3}Z_{P3})\|_F^2$$

.

.

.

$$P_{4N} : \min_{Z_P} \|Z_{PN-1} - \phi(D_{PN}Z_P)\| + \left\| \begin{bmatrix} Z_G \\ Z_P \end{bmatrix} - D_1Z_1 \right\|_F^2$$

$$\|\phi^{-1}(Z_{PN-1}) - D_{PN}Z_P\|_F^2 + \left\| \begin{bmatrix} Z_G \\ Z_P \end{bmatrix} - D_1Z_1 \right\|_F^2$$

$$P_{4N+1} : \min_{D_1} \left\| \begin{bmatrix} Z_G \\ Z_P \end{bmatrix} - D_1Z_1 \right\|_F^2$$

$$P_{4N+2} : \min_{D_2} \|Z_1 - \phi(D_2Z_2)\|_F^2 \equiv \|\phi^{-1}(Z_1) - D_2Z_2\|_F^2$$

.

.

.

$$P_{4N+M} : \min_{D_M} \|Z_{M-1} - \phi(D_MZ)\|_F^2 \equiv \|\phi^{-1}(Z_{M-1}) - D_MZ\|_F^2$$

$$P_{4N+M+1} : \min_{Z_1} \left\| \begin{bmatrix} Z_G \\ Z_P \end{bmatrix} - D_1Z_1 \right\| + \|Z_1 - \phi(D_2Z_2)\|_F^2$$

$$P_{4N+M+2} : \min_{Z_2} \|Z_1 - \phi(D_2Z_2)\|_F^2 + \|Z_2 - \phi(D_3Z_3)\|_F^2 \equiv$$

$$\|\phi^{-1}(Z_1) - D_2Z_2\|_F^2 + \|Z_2 - \phi(D_3Z_3)\|_F^2$$

.

.

.

$$P_{4M+2M} : \min_Z \|Z_{M-1} - \phi(D_M Z)\|_F^2 + \|t - \phi(wZ)\|_2^2 \equiv \\ \| \phi^{-1}(Z_{M-1}) - D_M Z \|_F^2 + \| \phi^{-1}(t) - wZ \|_2^2$$

$$P_{4M+2M+1} : \min_w \|t - \phi(wZ)\|_F^2$$

It is trivial to invert ϕ since it is a unitary operation. We have inverted the activation function such that the sub-problem is converted to a linear least square problem. Thus we have converted all the problems to such least squares problems having a closed form solution in the form of pseudo-inverse.

5.2.2 Testing

During the testing phase, let us assume two inputs x_1 and x_2 to the network. The problem is to determine whether the images belong to the same person or not. First step is to generate the coefficients z_1 and z_2 of inputs x_1 and x_2 respectively. This can be done by solving the following problem:

$$\min_{z_1} \|x_1 - D_{P1}\phi(D_{P2}\phi(\dots\phi(D_{PN}z_1)))\|_F^2 \quad (5.10)$$

This is solved using substitution. For the first level, we substitute $z_{P1} = \phi(D_{P2}\phi(\dots\phi(D_{PN}z_1)))$. The augmented lagrangian with this substitution is given by:

$$\min_{z_1, z_{P1}} \|x_1 - D_{P1}z_{P1}\|_F^2 + \|z_{P1} - \phi(D_{P2}\phi(\dots\phi(D_{PN}z_1)))\|_F^2 \quad (5.11)$$

For the second layer we substitute $z_{P2} = \phi(D_{P3}\phi(\dots\phi(D_{PN}z_1)))$, and so on until the second last layer. The final problem now becomes:

$$\min_{z_1, z_{P1} \dots z_{PN-1}} \|x_1 - D_{P1}z_{P1}\|_F^2 + \|z_{P1} - \phi(D_{P2}z_{P2})\|_F^2 + \dots + \|z_{PN-1} - \phi(D_{PN}z_1)\|_F^2 \quad (5.12)$$

This can be solved using ADMM. The above optimization problem can be segregated into the following subproblems:

$$\begin{aligned} S_1 : \min_{z_{P1}} & \|x_1 - D_{P1}z_{P1}\|_F^2 + \|z_{P1} - \phi(D_{P2}z_{P2})\|_F^2 \\ S_2 : \min_{z_{P2}} & \|\phi^{-1}(z_{P1}) - D_{P2}z_{P2}\|_F^2 + \|z_{P2} - \phi(D_{P3}\phi(\dots\phi(D_{PN}z_1)))\|_F^2 \\ & \cdot \\ & \cdot \\ & \cdot \\ S_N : \min_{z_1} & \|z_{PN-1} - \phi(D_{PN}z_1)\|_F^2 \end{aligned}$$

Similar to the training, all these subproblems have closed form solutions. Following the similar approach coefficients z_2 of the input x_2 can be computed by solving the problem below:

$$\min_{z_2} \|x_1 - D_{G1}\phi(D_{G2}\phi(\dots\phi(D_{GN}z_2)))\|_F^2 \quad (5.13)$$

Once the coefficients of both the inputs are obtained, they are stacked $[z_1; z_2]$ to form the input of the joint network. The joint representation

z is then obtained by solving the problem:

$$\min_z \left\| \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} - D_1 \phi(D_2 \phi(\dots \phi(D_M z))) \right\|_F^2 \quad (5.14)$$

Given the representation z , result is generated by multiplying it with the learned mapping w .

$$t^{predicted} = \phi(w * z) \quad (5.15)$$

If the value of $t^{predicted}$ is above a certain threshold the input images x_1 and x_2 belongs to the same person. This completes the testing phase.

5.3 Experiments and Results

For experiments all three types of siamese DDL architectures (Figure 5.2) are used. The effectiveness of the proposed algorithm is demonstrated with two case studies: (i) face recognition and (2) kinship classification. For face recognition, the experiments are performed on two face databases: IIIT-D Disguise Face Database (ID V1) [1, 2] and CrowdFace database [3]. For kinship classification, the experiments are performed on three kinship databases: Cornell [106], UB KinFace [107], KinFaceW (KFW)-I [108], and KFW-II [108].

- **IIIT-D Disguise Face Database (ID V1):** The ID V1 database contains 681 visible spectrum images pertaining to 75 participants with disguise variations. The images per participant varies from 6 to 10,



Figure 5.3: Example Images from IIIT-D Disguise Face Dataset (ID V1)

with at least one frontal neutral and at least five disguised faces. The images are captured in proper illumination and frontal pose. The image without any disguise is termed as the “neutral face image”. The disguise variations captured in the database include variations in hair style: different styles and colors of wigs, variations due to beard and mustache, variations due to glasses: sunglasses and spectacles, variations due to cap and hat, variations due to mask: disposable doctors mask, and a combination of multiple disguise accessories.

- **CrowdFace Database:** The CrowdFaceDB dataset contains 50,152 frames from 385 videos pertaining to 257 subjects, captured at different locations and each video contains up to 14 subjects. This is one of the most challenging unconstrained face databases publicly available in the research community. The data is captured using HandyCam, Nikon Coolpix, and iPhone 4s and 5c.
- **Kinship Databases:** The problem of kinship verification entails determining whether a pair of images belongs to kin or not. This is a two-

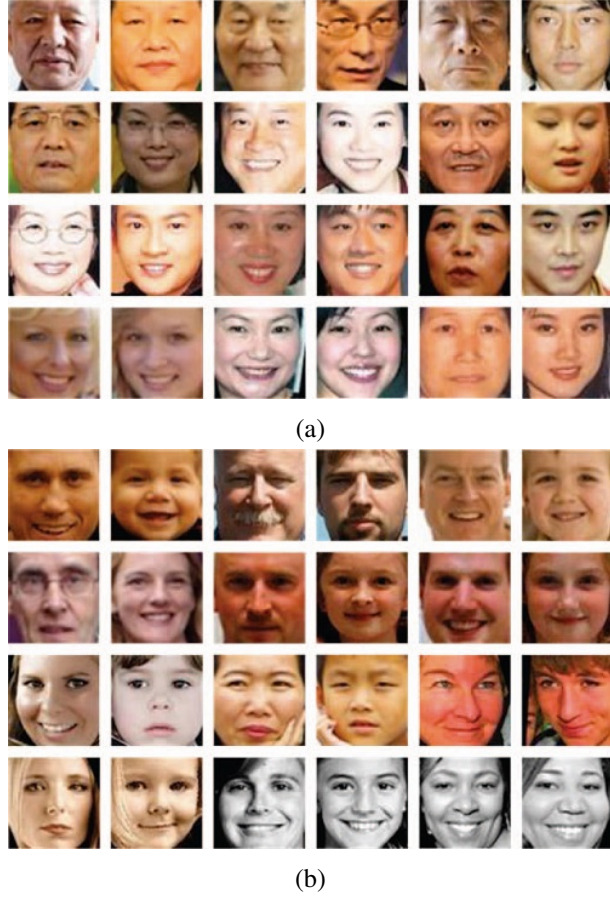


Figure 5.4: Example Images from KinFaceW-I and KinFaceW-II datasets

class problem, with the classes being kin and non-kin. In literature, several databases have been prepared for this task and in this research, we demonstrate the performance on (1) Cornell [106], (2) UB KinFace [107], (3) KinFaceW (KFW)-I [108], and (4) KFW-II [108] databases.

For face recognition, the models are pre-trained using the MultiPIE database [109] and then fine-tuned on the respective IDV1 and CrowdFace databases. The testing protocols are followed exactly as described in the original papers to ensure reproducibility and comparison with results reported on the benchmark protocols. The results of the proposed algorithm for face recognition are compared with VGGFace [110], along with the results reported

in literature on respective databases.

For kinship verification, the results are compared with the following existing kinship verification algorithms and the results are reported in the next subsection on the benchmark protocols provided with the databases.

- Neighborhood repulsed metric learning (NRML) [108],
- Prototype-based discriminative feature learning (PDFL) [111],
- Discriminative deep metric learning (DDMML) [112],
- Visual transformation aided contrastive learning (VTCL) [113], and
- Kinship verification via representation learning - filtered contractive deep belief networks (KVRL-fcDBN)-fcDBN [114]
- Multiview Deep Metric Learning (MvDML) [115]

5.3.1 Results on IIIT-D Disguise Face Database

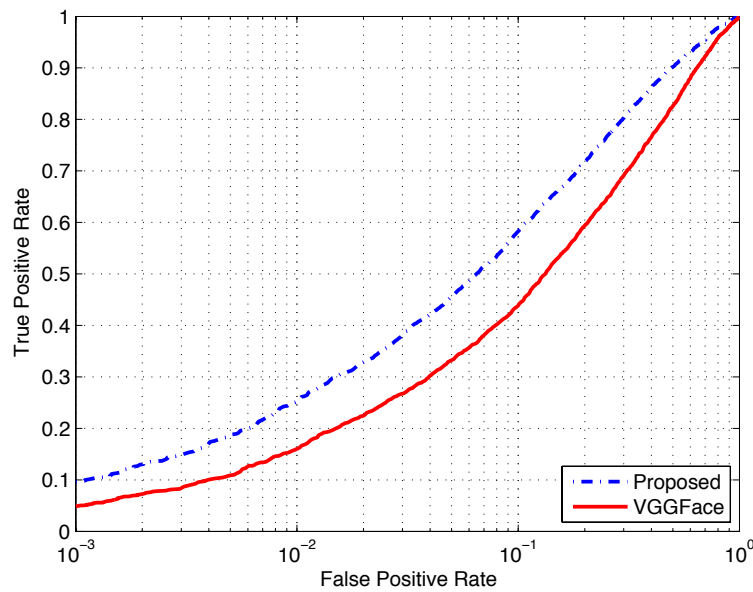


Figure 5.5: ROC curve on the IIIT Delhi disguise face database.

Approach	Verification Accuracy @ FAR		
	0.1%	1.0%	10%
SRC [58]	5.6	15.5	37.7
FaceVACS	10.9	17.1	22.5
ITE [1]	7.4	16.6	38.1
VGGFace	5.0	16.1	44.5
Proposed - Network 1	4.6	15.5	39.1
Proposed - Network 2	8.2	20.8	48.4
Proposed - Network 3	9.8	25.8	58.2

Table 5.1: Verification performance of the proposed and existing algorithms on the ID V1 database [1, 2]. The results are reported in terms of genuine accept rates at different false accept rates.

For IIIT-D Disguise Face Database, all three SDDL architectures are evaluated. Figure 5.5 showcases the receiver operating characteristics (ROC) curve pertaining to best performing SDDL network on the ID V1 database and comparison with VGGFace. Table 5.1 summarizes the results of the SDDL networks and comparison with Sparse Representation Classifier (SRC) [58], VGGFace, ITE [1], and a commercial matcher, FaceVACS. Due to the challenging nature of the IIIT-D Disguise Face database, verification accuracy at 0.1% FAR is low and comparable with the commercial matcher. However, for higher FAR values, such as 1% and 10% the proposed algorithm outperforms existing approaches by a big margin. Among three SDDL networks, third network with two layers for processing the probes and targets and one layer for fusion yields the best results.

5.3.2 Results on CrowdFace Database

As mentioned previously, CrowdFace database is one of the most challenging databases where multiple subjects appear in a given frame with unrestricted activities. Three baseline results are provided in [3]. Figure 5.6

FAR	0.1%	1%	10%
FaceVACS	2.3	5.9	19.2
OpenBR	0.0	10.0	11.0
VGG-Face	0.7	6.8	36.6
Proposed - Network 1	4.2	10.5	32.1
Proposed - Network 2	10.7	22.3	49.9
Proposed - Network 3	13.5	32.3	66.9

Table 5.2: Face verification performance of commercial systems, VGG-Face and the proposed algorithm on the CrowdFace database [3]. The results are reported in terms of genuine accept rate at different false accept rates.

showcases the ROC curve on the CrowdFace database with manually annotation of detected faces. This figure also shows comparison with VGGFace. As it can be seen, a well trained VGGFace classifier yields very poor performance for both video-to-video and video-to-frame protocols. On the other hand, the proposed SDDL algorithm showcaes drastic improvements. Further, Table 5.2 summarizes the comparison with benchmark results (for video-to-video protocol) as well as comparison among three SDDL networks. For both lower and higher FAR values, the proposed algorithm significantly (at 95% confidence interval) improves the results. The observed improvement is up to 30% at 10% FAR. Similar to the previous case study, here also we have observed that the third network performs teh best.

5.3.3 Results on Kinship Databases

Similarly, on the kinship databases, the proposed SDDL algorithm is consistently in top three performing algorithms, along with SNMAE and KVRL-*fc*DBN. Kinship verification experiments are performed on four different benchmark databases and results are compared with reported results. Table 5.3 summarizes the results of this experiment. On the kinship databases,

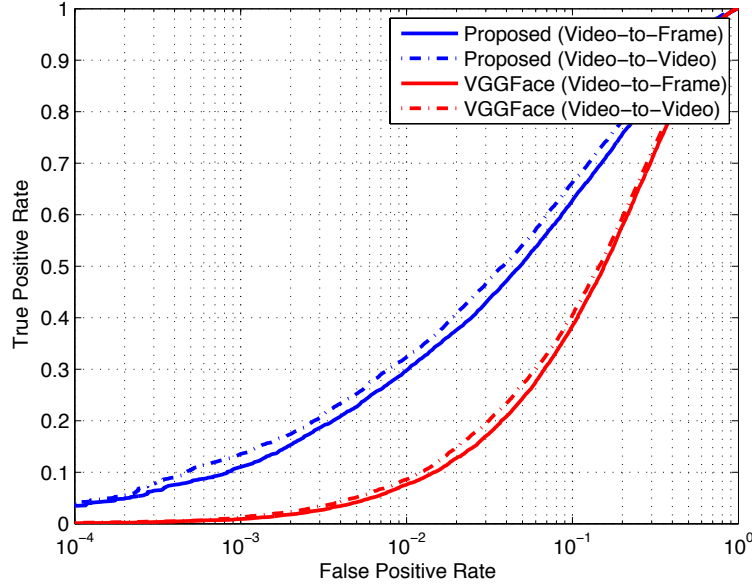


Figure 5.6: ROC curve on the CrowdFace database.

the proposed SDDL algorithm is consistently in top two performing algorithms, along with KVRL-*fc*DBN. Existing algorithms generally utilize either deep learning approaches or metric learning approaches or a combination. On the other hand, SDDL does not employ complex learning mechanism but still achieve very high performance. In this experiment as well, the third architecture showcases the best kinship classification accuracies.

These case studies have one thing in common: the third architecture of the proposed SDDL architecture showcases the best performance. To understand the effect, we performed experiments with varying levels of deep dictionary learning. In our experiments, we observe that two levels of deep dictionary learning and one level of fusion yields the optimal results.

Algorithm	Cornell	UBKin	KFWI	KFWII
NRML [108]	-	-	69.9	76.5
PDFL [111]	71.9	67.3	70.1	77.0
DDMML [112]	-	-	83.5	84.3
VTCL [113]	-	-	80.5	82.3
KVRL- <i>fc</i> DBN [114]	89.5	91.8	96.1	96.2
MvDML [115]	-	-	-	80.2
Proposed - Network 1	81.2	72.4	70.5	77.0
Proposed - Network 2	85.0	88.2	88.6	89.2
Proposed - Network 3	90.8	91.8	91.5	92.2

Table 5.3: Kinship verification accuracy (%) of the existing and proposed kinship verification algorithms on kinship datasets. The results are reported on pre-existing benchmark protocols of individual databases. The results of existing algorithms are reported from the original papers. Top-2 results are highlighted which shows that the proposed algorithm yields comparable results relative to state of the art algorithms.

5.4 Discussion

Here, we have discussed one aspect of the Siamese DDL framework, that of detection / analysis. But our framework is flexible enough to accommodate generic fusion tasks. For example, one common problem in physiological signal analysis is to infer blood pressure (BP) from ECG and PPG signals. In our framework, ECG can be input as X1 and PPG can be input as X2, and the output can be the BP. Such a fusion framework would 'learn' to infer an output from multiple sources. In general this can be extended to multi-source fusion.

What we have discussed so far are primarily analysis tasks. But the fusion framework can be applied for synthesis tasks as well. For example, in the problem of estimating the dense depth map from image and corresponding sparse depth map - this has potential applications in drone imagery. We can put two sources as inputs and the output would be the dense depth map.

Chapter 6

Coupled Deep Dictionary Learning

In this chapter, we focus on solving four standard inverse problems in imaging - denoising, deblurring, super-resolution, and reconstruction. Traditionally, signal processing techniques are used to solve such problems. However, such techniques are computationally expensive. Thus, one resorts to machine learning based solutions. In the past few years, variants of neural networks were developed to 'learn' the inversion operation. Although the learning was computationally expensive, the learnt model was light-weight and portable; suitable for deployment on leaner platforms. This work is based on the same concept, however, instead of using neural networks we treat inversion as a domain adaptation problem - a transfer from corrupted space to clean space. We propose a deep version of the well known coupled dictionary learning technique; but instead of learning a single layer of dictionary, we learn multiple layers.

The chapter is organized as follows. Section 6.2 discusses the existing work. Proposed algorithm is presented in section 1.2. Experiments and

results are discussed in section 6.4.

6.1 Introduction

In this work we focus on four classical inversion problems in imaging - denoising, deblurring, super-resolution, and reconstruction. In classical signal processing literature, all of the aforesaid three can be expressed as a linear inverse problem of the following form -

$$y = Ax + n \tag{6.1}$$

Here y is the acquired (corrupted) image, x is the clean image (to be estimated), A is the corruption operator and n is the noise. For denoising, A is identity; for deblurring it is a convolution with the blurring kernel, for super-resolution it is a sampling operator and for reconstruction it is a transformation (usually with sub-sampling). Classical signal processing techniques formulate 6.1 as an inverse problem, usually with sparsity constraints. The image is assumed to be sparse in some basis (wavelet, DCT, etc.); this prior information is exploited in order to estimate it. More modern techniques adopt a learning based approach, i.e. instead of assuming the basis to be fixed, they learn it from the image itself. The advantage of such signal processing based methods is that they are adaptive, and can restore any image without any prior knowledge.

The limitation of such an approach is that they are slow, requires solv-

ing complex optimization problems. This precludes their application for real-time problems. These days, on-chip processing is required for various modern day tasks like autonomous driving, unmanned aerial vehicles, etc. In such cases all of the aforesaid problems arise, but the processors on board are not powerful enough to carry out complex signal processing based techniques on the fly (in real-time). They require solutions based on edge computing (data processing as close to the source as possible). This precludes the use of signal processing based applications.

Most recent techniques on image restoration are based on deep learning, i.e., the corrupted image is at the input and the clean image is at the output of the deep neural network. During training, the model (stacked autoencoder / deep convolutional neural network) 'learns' the inversion operation from many examples. The training is time consuming, but can be done on powerful computers. But the trained models are light-weight and can be easily ported on board. During the operational stage, when the corrupted image is input to the learnt model, one only requires a few matrix vector multiplications to produce the clean output.

Our work is based on the same learning based philosophy, i.e., the training period is time consuming requiring powerful computers but the trained model is portable and can operate on the edge. However, it is not based on the straight-forward deep learning based approach. Rather we formulate inversion as a domain adaptation problem - a transformation from the corrupted space to the clean space. Our work is based on the coupled represen-

tation learning paradigm. Here, two models are learnt - one for the source (corrupted domain) and one for the target (clean domain), along with the two models a coupling map is learnt to transform from the representation space of the corrupted domain to that of the clean domain. During training, the models and the coupling map are learnt. In the operational stage, the representation of the corrupted image (source domain) is obtained from the corresponding model; this is next transferred to the target domain representation by the coupling map; finally, the target (clean) model is used to express the clean representation into the image space.

6.2 Literature Review

Broadly there are two approaches towards inversion - transductive and inductive. The former directly solve the inversion problem without any learning stage; this is more of a signal processing approach. The later, 'trains' on data to 'learn' the inversion operation; it has more of a machine learning flavor to it. These techniques have been discussed in chapter 2 in detail. Proposed work is a deeper version of the coupled representation learning technique, which is briefly discussed in the next subsection.

6.2.1 Coupled Representation Learning

The main concept of coupled representation learning is schematically shown in Figure 6.1. There are two domains - source and target; a model learns representations from each domain and a linear map is learnt from the rep-

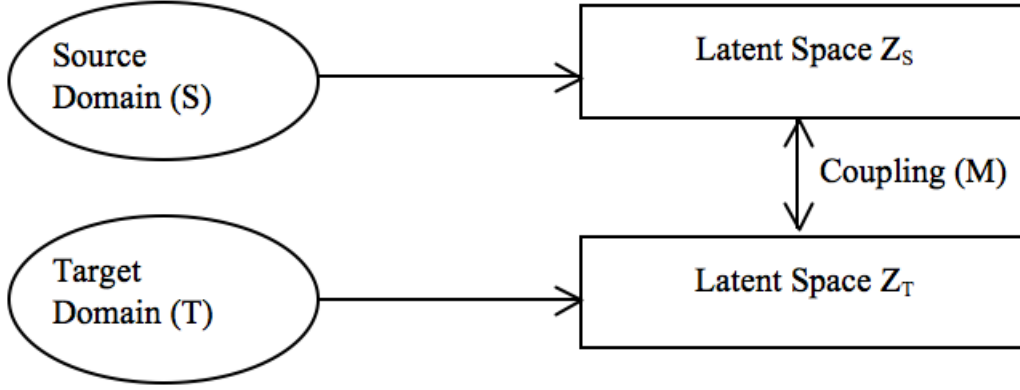


Figure 6.1: Coupled representation Learning

resentation of the source to that of the target. It has been used in several problems arising in domain adaptation.

So far there are two popular models for coupled representation learning; based on - dictionary learning and autoencoder. A relatively newer approach is the coupled transform learning formulation. The formulation for coupled dictionary learning [51, 116] is given by -

$$\begin{aligned}
 \min_{D_S, D_T, Z_S, Z_T} & \|X - D_S Z_S\|_F^2 + \|Y - D_T Z_T\|_F^2 + \gamma \|Z_T - M Z_S\|_F^2 + \\
 & \lambda_S \|Z_S\|_1 + \lambda_T \|Z_T\|_1
 \end{aligned} \tag{6.2}$$

Here X and Y are the data for the source and target domains respectively. D_S and the D_T are models (dictionaries) learnt for each of the two domains; their corresponding representations are Z_S and Z_T . M is the linear map coupling the representations. The dictionaries learn a representation in each domain and there is a map between the representations of the two

domains.

For synthesis tasks, once the training is complete, the dictionaries for the source and target are preserved along with the learnt linear map; the coefficients are not of any further use. Once a new sample in the source domain arrives, its representation is obtained by the learnt dictionary of the source domain. This is further mapped onto the target domain by the learnt linear map. The thus obtained representation in the target domain is synthesized into the corresponding signal by the target domain dictionary. For inverse problems, the source domain is usually the corrupted image, and the target domain is the clean image. This approach has been used for super-resolution [51], denoising [116], deblurring [117] and, reconstruction [118].

A deeper version of coupled dictionary learning has been proposed in [119]. But this is a greedy (hence suboptimal) approach. The shallower layer uses standard dictionary learning for feature generation. The generated features are fed into a coupled dictionary learning for domain adaptation. The model is not learnt in a joint fashion. This technique has been used for image super-resolution in a supervised fashion.

Following coupled dictionaries, coupled autoencoder based formulations for domain adaptation were introduced [120]. The general deep version is

given by,

$$\min_{f_T, f_S, g_T, g_S, M} \|Y - f_T(g_T(Y))\|_F^2 + \|X - f_S(g_S(X))\|_F^2 + \lambda \|g_T(Y) - M g_S(X)\|_F^2 \quad (6.3)$$

The symbols carry their usual meanings. The subscript T refers to the target domain and S refers to the source domain. Coupled autoencoders have been used for super-resolution [120] and deblurring [121]. The way it is used is similar to that of dictionary learning. Once the training is over, the encoders and decoders need to be preserved along with the learnt linear map. The source domain encoder is applied on new data to obtain its deep representation; this is then adapted to the target domain by the linear map. The decoder from the target domain is applied to the thus obtained representation to generate the signal.

6.3 Proposed Formulation

This work proposes coupled deep dictionary learning. The concept has been used in the past [119], but that was a greedy sub-optimal approach. This is because the output (subspaces) of the shallow layer encompasses the output (subspaces) of the deeper layers, hence the coupling at a deeper layer would produce the same transformation in space as a coupling in the shallow layer. This is the reason, why it is theoretically impossible to improve by going deeper using the greedy technique. In our proposed work, we learn all the layers and the coupling at the deepest layer, simultaneously.

We develop the concept for three layers as an example; extension to more or fewer layers will be self explanatory. Two deep dictionaries are learnt for each domain (say source and target) and there is a coupling map from the deepest representation of one to the other.

For the source and target domains we learn deep dictionaries (shown for three levels). This is expressed as follows,

$$X_S = D_{S1}\phi(D_{S2}\phi(D_{S3}Z_S)) \quad (6.4a)$$

$$X_T = D_{T1}\phi(D_{T2}\phi(D_{T3}Z_T)) \quad (6.4b)$$

Here S denotes the source domain and T the target domains; 1, 2 and 3 denotes the levels. Coupling maps are learnt from the deepest layers of the source and target domains.

$$Z_T = M_{ST}Z_S \quad (6.5)$$

The joint optimization is formulated as follows:

$$\begin{aligned} \min_{D_{S'1}, D_{T'1}, Z_S, Z_T, M_{ST}} & \|X_S - D_{S1}\phi(D_{S2}\phi(D_{S3}Z_S))\|_F^2 \\ & + \|X_T - D_{T1}\phi(D_{T2}\phi(D_{T3}Z_T))\|_F^2 + \lambda \|Z_T - M_{ST}Z_S\|_F^2 \end{aligned} \quad (6.6)$$

6.3.1 Training

In this work we follow the variable splitting augmented Lagrangian approach; this is a popular approach in signal processing, e.g. [14]. The first

level proxy variables are defined as,

$$Z_{S1} = \phi(D_{S2}\phi(D_{S3}Z_S)) \quad (6.7a)$$

$$Z_{T1} = \phi(D_{T2}\phi(D_{T3}Z_T)) \quad (6.7b)$$

With these proxies, the augmented Lagrangian for 6.6 is,

$$\begin{aligned} \min_{D_{S' s}, D_{T' s}, Z_{S' s}, Z_{T' s}, M_{ST}} & \|X_S - D_{S1}Z_{S1}\|_F^2 + \|X_T - D_{T1}Z_{T1}\|_F^2 + \lambda\|Z_T - M_{ST}Z_S\|_F^2 \\ & + \mu(\|Z_{S1} - \phi(D_{S2}\phi(D_{S3}Z_S))\|_F^2 + \|Z_{T1} - \phi(D_{T2}\phi(D_{T3}Z_T))\|_F^2) \end{aligned} \quad (6.8)$$

The second (and final level) proxies are defined as,

$$Z_{S2} = \phi(D_{S3}Z_S) \quad (6.9a)$$

$$Z_{T2} = \phi(D_{T3}Z_T) \quad (6.9b)$$

The final formulation takes the form,

$$\begin{aligned} \min_{D_{S' s}, D_{T' s}, Z_{S' s}, Z_{T' s}, M_{ST}} & \|X_S - D_{S1}Z_{S1}\|_F^2 + \|X_T - D_{T1}Z_{T1}\|_F^2 + \lambda\|Z_T - M_{ST}Z_S\|_F^2 \\ & + \mu(\|\phi^{-1}(Z_{S1}) - D_{S2}Z_{S2}\|_F^2 + \|\phi^{-1}(Z_{T1}) - D_{T2}Z_{T2}\|_F^2) + \\ & \mu(\|\phi^{-1}(Z_{S2}) - D_{S3}Z_S\|_F^2 + \|\phi^{-1}(Z_{T2}) - D_{T3}Z_T\|_F^2) \end{aligned} \quad (6.10)$$

Note that, it is trivial to invert the activation function since it acts element-wise. This 6.10 can be solved efficiently using the alternating direction method of multipliers (ADMM) [47]. ADMM segregates 6.10 into the fol-

lowing (easier) sub-problems.

$$P_1 : \min_{D_{S1}} \|X_S - D_{S1}Z_{S1}\|_F^2$$

$$P_2 : \min_{D_{S2}} \|\phi^{-1}(Z_{S1}) - D_{S2}Z_{S2}\|_F^2$$

$$P_3 : \min_{D_{S3}} \|\phi^{-1}(Z_{S2}) - D_{S3}Z_S\|_F^2$$

$$P_4 : \min_{D_{T1}} \|X_T - D_{T1}Z_{T1}\|_F^2$$

$$P_5 : \min_{D_{T2}} \|\phi^{-1}(Z_{T1}) - D_{T2}Z_{T2}\|_F^2$$

$$P_6 : \min_{D_{T3}} \|\phi^{-1}(Z_{T2}) - D_{T3}Z_T\|_F^2$$

$$P_7 : \min_{M_{ST}} \|Z_T - M_{ST}Z_S\|_F^2$$

$$P_8 : \min_{Z_{S1}} \|X_S - D_{S1}Z_{S1}\|_F^2 + \|Z_{S1} - \phi(D_{S2}Z_{S2})\|_F^2$$

$$P_9 : \min_{Z_{S2}} \|\phi^{-1}(Z_{S1}) - D_{S2}Z_{S2}\|_F^2 + \|Z_{S2} - \phi(D_{S3}Z_S)\|_F^2$$

$$P_{10} : \min_{Z_S} \|\phi^{-1}(Z_{S2}) - D_{S3}Z_S\|_F^2 + \|Z_T - M_{ST}Z_S\|_F^2$$

$$P_{11} : \min_{Z_{T1}} \|X_T - D_{T1}Z_{T1}\|_F^2 + \|Z_{T1} - \phi(D_{T2}Z_{T2})\|_F^2$$

$$P_{12} : \min_{Z_{T2}} \|\phi^{-1}(Z_{T1}) - D_{T2}Z_{T2}\|_F^2 + \|Z_{T2} - \phi(D_{T3}Z_T)\|_F^2$$

$$P_{13} : \min_{Z_T} \|\phi^{-1}(Z_{T2}) - D_{T3}Z_T\|_F^2 + \|Z_T - M_{ST}Z_S\|_F^2$$

Since the activation function is trivial to invert, we have used the forward operation and inverse operation in a fashion to keep the variable (being solved) linear. With ADMM approach (and proper use of the forward and inverse of the activation function) we can divide the entire problem 6.10 in each iteration as a series of simple least squares problems that can be updated in closed form via the pseudo-inverse. We have shown the derivation for three levels of dictionaries. But the approach used here can be extended to more.

6.3.2 Testing

In the testing phase, let us assume that the sample (x_s) is in the source domain. This is the corrupted sample (blurred / low resolution / noisy). The problem is to generate the clean sample (deblurred / high resolution / noiseless). The first step is to generate the coefficients (z_s) in the source domain. This requires solving the following,

$$\min_{z_s} \|x_s - D_{S1}\phi(D_{S2}\phi(D_{S3}z_s))\|_F^2 \quad (6.11)$$

Just like the training algorithm, the solution to 6.11 needs to be done via proxies / substitution. For the first level, the substitution is: $z_{S1} = \phi(D_{S2}\phi(D_{S3}z_s))$. This allows recasting 6.11 as,

$$\min_{z_s} \|x_s - D_{S1}z_{S1}\|_F^2 + \mu \|z_{S1} - \phi(D_{S2}\phi(D_{S3}z_s))\|_F^2 \quad (6.12)$$

For the second level the substitution is: $z_{S2} = \phi(D_{S3}z_s)$. This leads to,

$$\min_{z_s} \|x_s - D_{S1}z_{S1}\|_F^2 + \mu (\|z_{S1} - \phi(D_{S2}z_{S2})\|_F^2 + \|z_{S2} - \phi(D_{S3}z_s)\|_F^2) \quad (6.13)$$

This augmented Lagrangian can be solved using ADMM. One can segregate 6.13 into the following sub-problems.

$$\begin{aligned} & \min_{z_{S1}} \|x_s - D_{S1}z_{S1}\|_F^2 + \mu \|z_{S1} - \phi(D_{S2}z_{S2})\|_F^2 \\ & \min_{z_{S2}} \|\phi^{-1}(z_{S1}) - D_{S2}z_{S2}\|_F^2 + \|z_{S2} - \phi(D_{S3}z_s)\|_F^2 \\ & \min_{z_s} \|\phi^{-1}(z_{S2}) - D_{S3}z_s\|_F^2 \end{aligned}$$

As in the training algorithm, all the sub-problems have a closed form so-

lution in the form of pseudoinverse. Once the representation of the source signal is obtained, it needs to be mapped to the target domain. This is achieved by multiplication with the coupling map.

$$\hat{z}_T = M_{ST} * z_S \quad (6.14)$$

For synthesis problems the goal is to generate the sample in the other domain, for example in generating sketches from photos or super-resolving a low resolution image. In such scenarios, once the representation in the target domain is obtained, one needs to generate the corresponding sample. This is easily done by applying the dictionaries (with appropriate activation function) from the target domain.

$$\hat{x}_T = D_{T1}\phi(D_{T2}\phi(D_{T3}\hat{z}_T)) \quad (6.15)$$

6.4 Experiments and Results

For the first three experiments (denoising, deblurring and, super-resolution) we have followed the protocol outlined in [122]. The training image set consists of 91 images for all the problems. For the first three problems (Denoising, Deblurring and, Super-resolution) testing dataset comprised of Set5 and Set14 consisting of 5 and 14 images respectively. For the last problem (Reconstruction) the test data consists of 11 standard test images.

Images	PSNR					SSIM				
	Noisy	Transform	BM3D	DnCNN	Proposed	Noisy	Transform	BM3D	DnCNN	Proposed
Set 14										
baboon	20.14	21.82	23.93	23.17	23.95	0.54	0.50	0.68	0.62	0.72
barbara	20.12	24.31	25.57	25.42	25.56	0.40	0.62	0.65	0.66	0.66
bridge	20.15	23.80	25.07	24.90	25.84	0.50	0.59	0.71	0.70	0.71
coastguard	20.15	24.81	25.32	25.78	25.72	0.40	0.55	0.68	0.66	0.76
comic	20.28	22.06	24.49	23.72	23.74	0.59	0.67	0.77	0.76	0.76
face	20.79	27.86	27.86	27.04	27.96	0.27	0.60	0.62	0.61	0.71
flower	20.33	25.21	26.30	26.53	26.52	0.40	0.68	0.68	0.71	0.71
foreman	20.54	26.46	27.19	27.20	27.67	0.29	0.60	0.65	0.68	0.68
lena	20.07	27.03	26.96	27.08	27.03	0.27	0.61	0.59	0.63	0.63
man	20.48	24.73	25.96	26.37	26.35	0.40	0.59	0.66	0.64	0.68
monarch	20.05	26.48	26.73	27.06	27.94	0.28	0.68	0.59	0.66	0.68
pepper	20.17	27.34	27.12	27.23	27.55	0.26	0.62	0.58	0.62	0.66
ppt3	20.54	23.00	27.03	24.26	27.07	0.38	0.56	0.77	0.71	0.77
zebra	20.31	24.50	26.12	26.91	25.87	0.46	0.66	0.72	0.74	0.74
Set 5										
baby	20.42	27.28	27.69	27.85	28.05	0.27	0.60	0.65	0.66	0.68
bird	20.41	27.38	27.23	27.10	27.33	0.32	0.72	0.69	0.70	0.72
butterfly	20.17	23.05	25.00	24.18	25.16	0.52	0.70	0.73	0.74	0.76
head	20.75	27.66	27.93	27.82	28.05	0.27	0.60	0.62	0.60	0.63
woman	20.45	25.80	26.84	27.31	27.32	0.36	0.70	0.70	0.72	0.72

Table 6.1: Denoising Results

6.4.1 Denoising

We compare with all the standard denoising approaches -transform learning [123] (yields better results than dictionary learning), BM3D (gold standard) and DnCNN [124]; the first two are the best known transductive learning techniques while the last one is the state-of-the-art inductive method. DnCNN yields better results than sparse stacked autoencoder (SSDA) based denoising techniques; hence SSDA solutions have been compared against in this work.

For our proposed technique, we need to specify certain parameters. We have used overlapping patches of size 16×16 . The best results were obtained from a three layer architecture; the number of atoms are $128 - 64 - 32$. Our algorithm requires specification of one parameter λ and one hyper-



Figure 6.2: Barbara- Denoising Results. a)Ground Truth b)Noisy c)Transform d)BM3D e)DnCNN f) Proposed

parameter μ . Note that the hyper-parameter refers to the relative importance of each level; since there is no reason to give more weightage to one layer over the other, we keep $\mu = 1$. For the parameter λ , we use leave one cross validation in the training set for tuning it. The value we arrive at is $\lambda = .8$.

The results for denoising are shown in Table 6.1. To generate the noisy images, moderate Gaussian noise ($\sigma = 25$ and $\mu = 0$) was used. The best results are shown in bold; but for cases where the difference in PSNR is less than 0.1 dB, both have been considered. This is because such minute variation in PSNR is imperceptible to the human eye. One can see that our method yields the best denoising performance for the majority of the cases. In other cases, the results are close to the best. For visual evaluation, we show the image of Barbara is shown in Figure 6.2.

6.4.2 Deblurring

Images	PSNR					SSIM				
	Blurry	SPB	VCNN	CDAE	Proposed	Blurry	SPB	VCNN	CDAE	Proposed
Set 14										
baboon	20.29	20.21	21.30	23.54	23.91	0.37	0.37	0.49	0.73	0.76
barbara	23.51	22.86	24.13	26.49	26.96	0.67	0.64	0.70	0.82	0.85
bridge	21.06	20.93	22.97	24.51	25.05	0.44	0.44	0.59	0.75	0.76
coastguard	22.25	22.19	23.83	25.70	25.70	0.38	0.38	0.52	0.73	0.73
comic	18.59	18.41	20.44	22.26	22.28	0.45	0.44	0.60	0.70	0.74
face	28.02	27.79	29.53	31.07	31.65	0.66	0.65	0.72	0.82	0.87
flower	21.43	21.25	23.74	25.21	25.73	0.60	0.59	0.70	0.73	0.79
foreman	23.56	23.10	26.53	26.98	27.40	0.72	0.70	0.78	0.80	0.85
lena	26.62	26.36	28.81	30.22	30.65	0.76	0.75	0.80	0.83	0.87
man	22.58	22.39	24.28	26.04	26.49	0.58	0.56	0.66	0.68	0.78
monarch	21.83	21.66	24.80	25.39	25.91	0.77	0.76	0.82	0.84	0.86
pepper	26.10	25.70	28.44	29.42	29.96	0.75	0.73	0.79	0.80	0.84
ppt3	18.97	18.69	21.49	23.00	23.00	0.72	0.70	0.77	0.78	0.78
zebra	19.61	19.40	22.52	23.45	23.45	0.48	0.47	0.67	0.74	0.77
Set 5										
baby	27.17	26.71	29.58	31.06	31.69	0.78	0.76	0.85	0.86	0.89
bird	25.13	24.69	28.01	29.06	29.63	0.76	0.74	0.83	0.84	0.85
butterfly	16.32	16.12	19.51	20.04	20.88	0.48	0.47	0.62	0.63	0.65
head	28.09	27.82	29.60	31.77	31.43	0.66	0.65	0.72	0.84	0.84
woman	22.32	21.98	25.36	26.80	27.24	0.73	0.71	0.80	0.80	0.82

Table 6.2: Deblurring Results

For deblurring, the experimental design remains the same as before. The blurred images were synthesized with known motion blur kernels. We have compared our proposed formulation with deep version of coupled autoencoder (CDAE) [120], Vectorized CNN (VCNN) [125] and sparse prior deblurring (SPB) [126]. The first two are the most recent inductive (non-adaptive approaches), and the last one is the state-of-the-art adaptive (blind) approach for image deblurring. In the paper [119], it has been shown that CDAE outperforms coupled dictionary learning, so we do not show those results here. Coupled transform learning was also used, but the results were not much better than coupled dictionary learning. The numerical results for



Figure 6.3: Man - Deblurring Results. a) Ground Truth b) Blurred image c) SPB d) VCNN e) CDAE f) Proposed

Deblurring (with CDAE, VCNN and SPB) are shown in Table 6.2.

For our proposed method everything remains the same as in section 6.4.1 apart from the value of λ . As before, we arrive at its final value ($\lambda = .6$) for deblurring using cross validation on the training set.

The results show that our method consistently yields the best results. Coupled deep autoencoder comes a distant second (more than 0.5 dB difference on an average). Other techniques like VCNN and SPB are much worse. The visual results corroborate the numerical metrics.

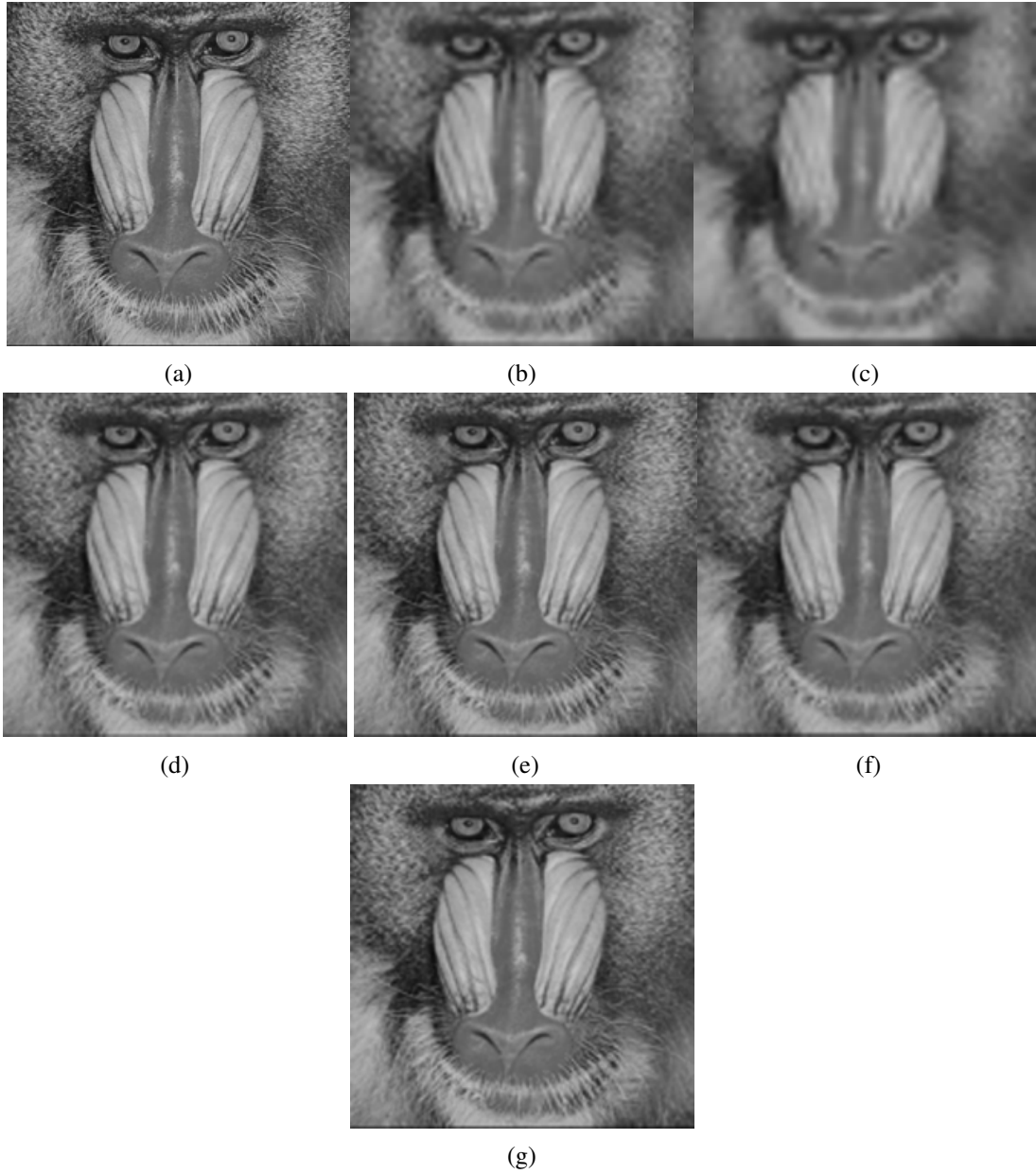


Figure 6.4: Baboon - Super-resolution Results. a) Ground Truth b) Upsampled c) CTL d) SRCNN e) SRN f) CDAE g) Proposed

Images	PSNR						SSIM					
	Upsampling*	CTL	SRCNN	SRN	CDAE	Proposed	Upsampling*	CTL	SRCNN	SRN	CDAE	Proposed
Set 14												
baboon	24.66	24.28	24.37	25.18	25.24	25.24	0.69	0.67	0.76	0.76	0.75	0.76
barbara	27.94	27.19	28.52	28.28	28.47	28.53	0.84	0.79	0.88	0.87	0.87	0.88
bridge	26.50	25.84	27.60	27.66	27.30	27.66	0.79	0.75	0.84	0.84	0.84	0.85
coastguard	29.14	28.30	30.49	30.57	30.23	30.69	0.79	0.75	0.85	0.85	0.85	0.86
comic	26.06	25.33	28.31	28.33	27.33	28.47	0.85	0.80	0.91	0.92	0.89	0.92
face	34.87	33.04	35.65	35.76	35.59	35.79	0.86	0.81	0.88	0.88	0.88	0.88
flower	30.43	29.41	33.10	33.31	31.75	33.78	0.90	0.85	0.93	0.94	0.92	0.95
foreman	32.68	30.71	33.84	34.59	33.60	34.58	0.95	0.81	0.95	0.96	0.95	0.96
lena	34.75	32.47	36.55	36.57	35.96	36.99	0.91	0.82	0.93	0.93	0.93	0.94
man	29.27	28.50	30.84	30.93	30.08	31.25	0.84	0.79	0.88	0.88	0.87	0.89
monarch	32.98	31.40	37.20	37.61	34.56	37.90	0.96	0.87	0.97	0.97	0.97	0.98
pepper	33.08	31.37	34.00	35.57	33.78	35.74	0.90	0.81	0.91	0.92	0.91	0.93
ppt3	26.92	26.08	30.44	30.40	28.30	30.26	0.95	0.78	0.97	0.98	0.95	0.97
zebra	30.68	28.74	33.32	33.42	32.45	33.44	0.91	0.83	0.94	0.94	0.92	0.94

Table 6.3: Super-resolution Results

6.4.3 Super-resolution

For super-resolution we have compared with coupled transform learning (CTL), coupled deep autoencoder (CDAE), CNN super-resolution (SRCNN) [45] and sparse super-resolution network SRN [127]. We have not compared with the recent generative adversarial network based techniques like [128, 129] because, they are known to yield photo-realistic images but do not beat CNN based techniques [45, 127] in terms of any known metrics. We also have not compared against coupled dictionary learning (CDL) because, it has been shown in [120, 121] that CDL yields worse results than CAE and, CDAE.

For the previous experiments, the dimensions of the source and the target were same. This is not the case for super-resolution; therefore we need to specify the configuration for the source and target separately. We found that a two layer architecture yields the best results. In the experiments shown here, a magnification of $(2 \times 2 = 4)$ has been implemented. The

number of dictionary atoms for the source is 8×8 and for the target at 16×16 ; both use overlapping patches. The number of atoms for the two layer source dictionaries is $64 - 16$ and for the two layer target dictionaries are $256 - 64$. The value of λ used here (via cross validation on the training set) is $\lambda = .8$.

The results show that our method yields the best results on an aggregate; especially in terms of SSIM. We mentioned before that, we are not showing results for GAN based methods and CDL in the main paper since they are known to be worse. For the sake of completeness, we have shown them in the appendix. In there, one can find a comparison with a popular signal processing based super-resolution technique [31].

6.4.4 Reconstruction

For image reconstruction, the protocol outlined in [46] is as follows -

- The image is first compressively sampled (using Bernoulli projection matrix).
- From the compressed data, a noisy image is recovered by applying the transpose of the compressing matrix.
- This is given as input for reconstruction; the output being the clean image.
- The network 'learns' to remove the artifacts and produce a clean image.

Images	PSNR				SSIM			
	RODEO	ReConNet	Transform	Proposed	RODEO	ReConNet	Transform	Proposed
Monarch	17.28	18.19	19.94	20.80	0.46	0.52	0.62	0.64
Parrots	20.17	20.37	23.08	23.84	0.56	0.63	0.76	0.78
Barbara	19.86	20.38	22.28	22.85	0.40	0.48	0.57	0.59
boats	19.63	21.35	23.08	23.90	0.44	0.53	0.62	0.63
cameraman	19.10	19.28	21.14	21.98	0.45	0.53	0.63	0.65
fingerprint	17.12	16.93	18.47	19.02	0.33	0.42	0.49	0.52
flinstones	15.89	16.37	17.92	18.30	0.33	0.36	0.46	0.49
foreman	22.01	23.69	27.52	27.93	0.55	0.66	0.80	0.81
house	21.31	22.57	24.99	25.67	0.52	0.61	0.72	0.74
lena	20.30	21.26	23.45	24.31	0.49	0.56	0.67	0.69
peppers	18.60	19.56	21.71	22.57	0.47	0.51	0.65	0.67

Table 6.4: Reconstruction Results

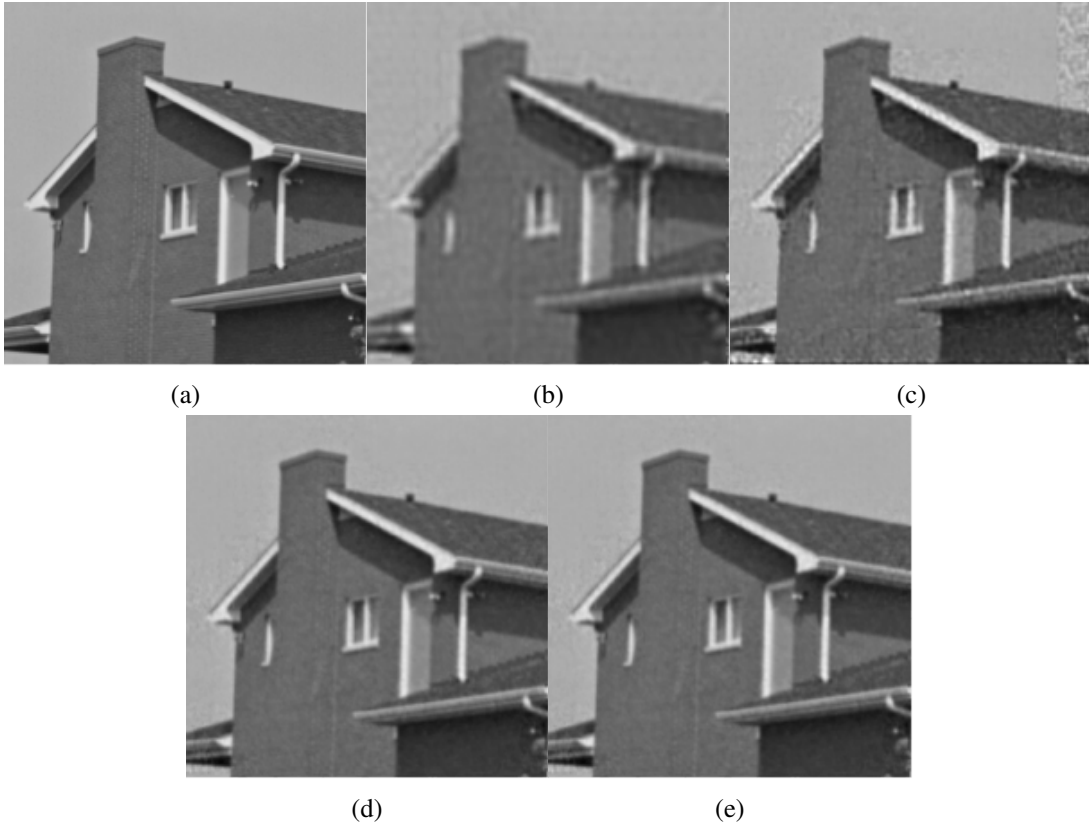


Figure 6.5: Foreman - Reconstruction Results. a) Ground Truth b) RODEO c) ReconNet d) Transform e) Proposed

In our framework, we do not need to apply the transpose of the compressing matrix and produce the image as the source. We directly feed the compressed data as the source and the corresponding clean image as the target. The coupled network learns to reconstruct the image directly from the compressed domain.

The best known technique for recovering images from such compressive samples is based on the transform learning approach [38]; this involves no learning and is based on the signal processing approach of solving inverse problems. There are two deep learning based techniques for inductive reconstruction. The first one is the ReconNet [46] (CNN based), the second one is RODEO [43] (stacked autoencoder based). To the best of our knowledge coupled dictionary and transform learning has not been used for reconstruction; hence we have not compared against them. For the experiments we have used 16 : 1 compression. Here we follow the experimental protocol outlined in [46].

In proposed technique we use a two layer architecture for the source and the target. The number of dictionary atoms is halved in every successive layer. The value of $\lambda = 0.1$ has been found by cross validation on the training set. The results are shown in Table 6.4.

The values for prior studies are commensurate with published results. The results conclusively show that our proposed deep method outperforms shallow coupled dictionary learning based techniques. Visual evaluation corroborate the numerical metrics.

6.5 Discussion

Here, a deeper version of existing coupled dictionary learning is proposed. Coupled dictionary learning framework is extended by incorporating multiple layers of dictionaries in the source and target models. In this work we focus on solving four standard inverse problems - denoising, deblurring, super-resolution and reconstruction. We have seen empirically that this leads to considerable improvements in results.

Learning a mapping between the source and target directly, is known to give poor results. This is because there is huge variability in images, and trying to learn a coupling map that encompasses all possible variabilities is difficult. When one layer of dictionary is learned, the coupling is no longer between the source and the target but on their representations. These representations are directly linked to the basis elements learned by the model. When we propose to go deeper, we learn more abstract basis. The coupling learned on the representation of such a basis will be even more robust and generic; since it needs to capture even less variability than a shallow model.

Chapter 7

Conclusions

This thesis focuses on developing a new framework for deep learning called deep dictionary learning (DDL). By developing this new framework we try to overcome some shortcomings of existing deep learning methods. Currently there are three main approaches to deep learning: convolutional neural networks (CNN), deep belief networks (DBN) and stacked autoencoder (SAE). CNNs are known to give best results in image analysis tasks, but their applicability is limited largely to the images in visible range. CNNs require large amount of labeled data to train; which preclude their applicability in the unsupervised tasks like segmentation and clustering.

Restricted Boltzmann machine forms a basic building block for DBN. Mathematically its cost function is cumbersome and the input to an RBM can only be binary or between the range of 0 to 1; which is highly restrictive. SAE overcomes the limitations of DBN but it requires learning of twice the number of parameters than the other neural networks. Which makes it sensitive to overfitting.

We are developing this new framework considering three key points: 1. It's cost function should be amenable to modification over time. 2. Should be amenable to semi-supervised and supervised learning 3. Should learn optimal number of connections.

Rest of the chapter is organized as follows. Section 7.1 presents chapter wise contribution of the thesis. In Section 7.2 future work is discussed.

7.1 Summary of Contribution

7.1.1 Chapter 1 : Deep Dictionary Learning - Classification and Clustering Problem

DDL is a deeper version of dictionary learning where, dictionary acts as a basic building block in the same way as autoencoder is a basic building block for stacked autoencoder. DDL is formed by stacking multiple layers of dictionaries one after the other. The existing approach to solve DDL is greedy, where one layer is learned at a time. In the greedy approach the input to the first layer is the training data. Then the output coefficients learned in the previous layer forms the input to the next layer. The problem with this approach is that the data flows in only forward direction. The deeper layers have no influence on the learning of shallower layers. Hence no feedback. In this chapter, we are addressing this shortcoming by proposing a solution where all the dictionaries and coefficients are learned simultaneously in a single optimization problem. This is unsupervised method, and any classifier can be used to make predictions.

Experiments are performed on standard classification and clustering datasets. It is seen that in both the cases proposed approach performed better than the fundamental generic methods.

7.1.2 Chapter 2 : Deep Dictionary Learning - Inverse Problem

In this chapter four imaging problems: denoising, deblurring, super-resolution and reconstruction are solved using deep dictionary learning. These problem can be modeled as linear inverse problems. In dictionary based inversion, the dictionary is learned adaptively from the image during the inversion process. We propose to adaptively learn multiple levels of dictionaries during inversion. This gives a deep dictionary learning based inversion formulation. Experiments are performed on benchmark datasets and proposed method outperforms the shallow (adaptive) and CNN based non-adaptive solutions.

7.1.3 Chapter 3 : Supervised Deep Dictionary Learning

This work proposes to incorporate supervision into the deep dictionary learning framework. Supervision is introduced by adding label consistency penalty to the previous unsupervised formulation. The derivation for solving the ensuing formulation is based on the state-of-the-art optimization paradigm that includes proximal variable splitting, augmented Lagrangians and alternating direction method of multipliers. This formulation is flexible enough to handle single label and multi-label classification problems.

The experiments have been carried out on benchmark datasets for both single label and multi-label problems. In every case, our formulation excels over the rest.

7.1.4 Chapter 4 : Semi-Supervised Deep Blind Compressed Sensing

This work proposes a semi-supervised method for classifying biomedical signals from their compressive measurements. This requirement arises in wireless body area networks, where it is essential to compress the signal at the sensor nodes with minimal computation. Previously it was done in two stages. In the first stage, the signal was reconstructed by some compressed sensing technique. In the second stage, some classifier was used to analyze the reconstructed samples. The main issue with this approach is that the compressed sensing reconstruction introduced certain artifacts, which hampered the classification performance. We propose to combine both the stages into a single formulation. Results are shown on EEG and ACG signals and the proposed method excels over the other generic methods in both reconstruction and classification.

7.1.5 Chapter 5 : Siamese Deep Dictionary Learning

This chapter proposes an architecture of Siamese deep dictionary learning. In a siamese network, there are two parallel processing pipelines for two inputs for example face verification problem with the objective of classifying two inputs as match or mismatch. The experimental effectiveness is

demonstrated on five different image databases pertaining to two classification problems: face verification and kinship verification. The experiments show that the proposed Siamese Deep Dictionary Learning yields comparable results compared to popular benchmark algorithms on all five databases. Other than these verification problems proposed architecture can be applied to generic fusion tasks. For example prediction of BP from ECG and PPG data. The ECG and PPG signals will form the input to the two pipelines of the network.

7.1.6 Chapter 6 : Coupled Deep Dictionary Learning

In this chapter we focus on solving four standard inverse problems in imaging-denoising, deblurring, super-resolution, and reconstruction. The inverse problems are modeled as a domain adaptation problem a transfer from corrupted space (source) to clean space (target). Existing approaches learn a linear map between source and target domains. Deep architecture can help in learning complex mappings. We propose a deep version of existing coupled dictionary learning technique, but instead of learning a single dictionary, multiple layer of dictionaries are learned. Experimental results on standard datasets show that our method yields the best results.

7.2 Future Work

7.2.1 Residual Deep Dictionary Learning

It is difficult to decide the optimal number of layers in a deep network. It is known that "Deeper is better" since it gives more expressiveness and better generalization. But, if the number of layers is too large the training becomes difficult because of the vanishing/exploding gradient problem. In theory, the training error should reduce with an increase in the number of layers in the network. In practice, however, the training error starts to increase after a certain number of layers. Residual Neural Networks (ResNet) do not suffer from this problem.

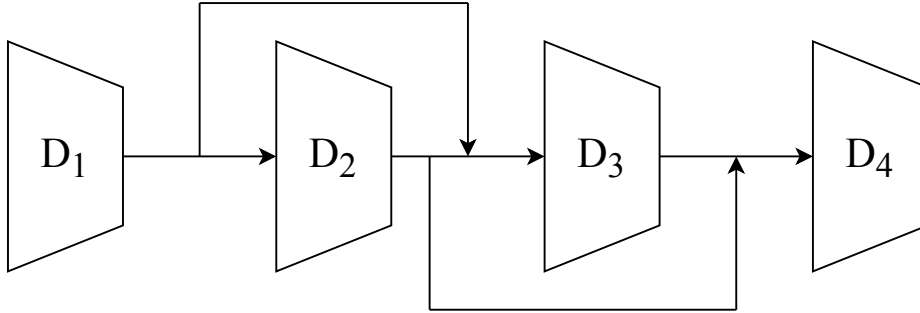


Figure 7.1: Residual Dictionary Learning

ResNet's are formed by stacking multiple residual blocks together. In traditional neural network, each layer fed into the next layer. On the other hand, in residual blocks activation unit from a layer is fed directly to a deeper layer, which is termed a skip connection or short cut.

We propose to develop ResNet architecture with dictionary learning. Figure 7.3 shows an example of residual deep dictionary learning. Here, the

output coefficients of the first layer are fed directly to the input of the third layer. Similarly, the output activations of the second layer have a direct connection to the input of the fourth layer.

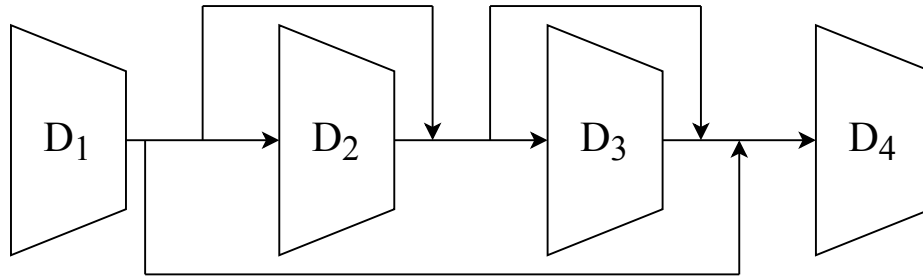


Figure 7.2: Dense Dictionary Learning

DenseNet's are logical extension of ResNets. In DenseNet every two layers are connected; every deeper layer receives input from all preceding layers. The inputs from the previous layers are combined by concatenation, which makes output representation a combination of simple and complex features. This can help in better generalization. Similar architecture can be formed using dictionary learning. Figure 7.2 shows an example of a deep dictionary based denseNet. As can be seen the input of the third layer is the concatenation of output of the first and second layer. Similarly, fourth layer receives input from all the preceding layers. In future we would like to explore dictionary learning based ResNet and DenseNet architectures.

7.2.2 Recurrent Deep Dictionary Learning

Feed forward neural networks can only work with fixed size input and fixed size output. They are designed to handle sequences or time series data. Recurrent neural networks can work with variable size input and outputs

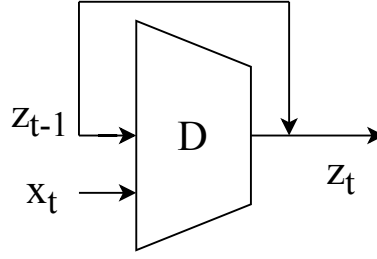


Figure 7.3: Recurrent Dictionary Learning

and are capable of handling time series data. They have shown significant performance in applications like image captioning, generating music and machine translation. In recurrent neural networks output of a layer in the network is fed back to the input of that layer. Unlike feed forward networks recurrent neural network can receive a sequence of values as input, and it can also produce a sequence of values as output, this ability opens up this network to a wide variety of applications for example document classification and image captioning. Dictionary learning can be used to generate an RNN like architecture. As shown in figure 7.3, the output of the network z_{t-1} at time $t - 1$ is fed back to the network along with input x_t as the input at time t . In future, we would like to explore Recurrent neural network deep architecture using dictionary learning as basic building block.

References

- [1] T. I. Dhamecha, R. Singh, M. Vatsa, and A. Kumar, “Recognizing disguised faces: Human and machine evaluation,” *PloS one*, vol. 9, no. 7, p. e99212, 2014.
- [2] T. I. Dhamecha, A. Nigam, R. Singh, and M. Vatsa, “Disguise detection and face recognition in visible and thermal spectrums,” in *Biometrics (ICB), 2013 International Conference on*. IEEE, 2013, pp. 1–8.
- [3] T. I. Dhamecha, M. Shah, P. Verma, M. Vatsa, and R. Singh, “Crowd-facedb: Database and benchmarking for face verification in crowd,” *Pattern Recognition Letters*, vol. 107, pp. 17–24, 2018.
- [4] Z. Zhang, T.-P. Jung, S. Makeig, and B. D. Rao, “Compressed sensing for energy-efficient wireless telemonitoring of noninvasive fetal ecg via block sparse bayesian learning,” *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 2, pp. 300–309, 2013.
- [5] P. Baldi, “Autoencoders, unsupervised learning, and deep architectures,” in *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, 2012, pp. 37–49.

- [6] I. Sutskever, G. E. Hinton, and G. W. Taylor, “The recurrent temporal restricted boltzmann machine,” in *Advances in Neural Information Processing Systems*, 2009, pp. 1601–1608.
- [7] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, “Recurrent neural network based language model,” in *Eleventh annual conference of the international speech communication association*, 2010.
- [8] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, p. 436, 2015.
- [9] G. E. Hinton, “Deep belief networks,” *Scholarpedia*, vol. 4, no. 5, p. 5947, 2009.
- [10] Y. LeCun *et al.*, “Lenet-5, convolutional neural networks,” *URL: <http://yann.lecun.com/exdb/lenet>*, p. 20, 2015.
- [11] S. Tariyal, A. Majumdar, R. Singh, and M. Vatsa, “Deep dictionary learning,” *IEEE Access*, vol. 4, pp. 10 096–10 109, 2016.
- [12] K. Engan, S. O. Aase, and J. H. Husoy, “Method of optimal directions for frame design,” in *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, vol. 5. IEEE, 1999, pp. 2443–2446.
- [13] I. Daubechies, M. Defrise, and C. De Mol, “An iterative thresholding algorithm for linear inverse problems with a sparsity constraint,” *Communications on pure and applied mathematics*, vol. 57, no. 11, pp. 1413–1457, 2004.

- [14] P. L. Combettes and J.-C. Pesquet, “Proximal splitting methods in signal processing,” in *Fixed-point algorithms for inverse problems in science and engineering*. Springer, 2011, pp. 185–212.
- [15] E. G. Birgin and J. M. Martínez, *Practical augmented Lagrangian methods for constrained optimization*. SIAM, 2014, vol. 10.
- [16] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [17] A. Gogna, A. Majumdar, and R. Ward, “Semi-supervised stacked label consistent autoencoder for reconstruction and analysis of biomedical signals,” *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 9, pp. 2196–2205, 2017.
- [18] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [19] K. Lang, “Newsweeder: Learning to filter netnews,” in *Proceedings of the Twelfth International Conference on Machine Learning*, 1995, pp. 331–339.
- [20] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio, “An empirical evaluation of deep architectures on problems with many factors of variation,” in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 473–480.

- [21] G. Tzanetakis and P. Cook, “Musical genre classification of audio signals,” *IEEE Transactions on speech and audio processing*, vol. 10, no. 5, pp. 293–302, 2002.
- [22] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion,” *Journal of machine learning research*, vol. 11, no. Dec, pp. 3371–3408, 2010.
- [23] A. Makhzani and B. Frey, “K-sparse autoencoders,” *arXiv preprint arXiv:1312.5663*, 2013.
- [24] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, “Contractive auto-encoders: Explicit invariance during feature extraction,” in *Proceedings of the 28th International Conference on International Conference on Machine Learning*. Omnipress, 2011, pp. 833–840.
- [25] G. E. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.
- [26] F. Tian, B. Gao, Q. Cui, E. Chen, and T.-Y. Liu, “Learning deep representations for graph clustering.” in *AAAI*, 2014, pp. 1293–1299.
- [27] X. Peng, S. Xiao, J. Feng, W.-Y. Yau, and Z. Yi, “Deep subspace clustering with sparsity prior.” in *IJCAI*, 2016, pp. 1925–1931.
- [28] P. Rodríguez and B. Wohlberg, “Efficient minimization method for a generalized total variation functional,” *IEEE Transactions on Image Processing*, vol. 18, no. 2, pp. 322–332, 2009.

- [29] H. Fang, Y. Chang, G. Zhou, and L. Deng, “Iteratively reweighted blind deconvolution with adaptive regularization parameter estimation,” *IEEE Access*, vol. 5, pp. 11 959–11 973, 2017.
- [30] L. He and Y. Wang, “Iterative support detection-based split bregman method for wavelet frame-based image inpainting,” *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5470–5485, 2014.
- [31] J. Yang, J. Wright, T. S. Huang, and Y. Ma, “Image super-resolution via sparse representation,” *IEEE transactions on image processing*, vol. 19, no. 11, pp. 2861–2873, 2010.
- [32] D. L. Donoho and J. Tanner, “Precise undersampling theorems,” *Proceedings of the IEEE*, vol. 98, no. 6, pp. 913–924, 2010.
- [33] E. J. Candès and M. B. Wakin, “An introduction to compressive sampling,” *IEEE signal processing magazine*, vol. 25, no. 2, pp. 21–30, 2008.
- [34] M. Aharon, M. Elad, and A. Bruckstein, “*rmk*-svd: An algorithm for designing overcomplete dictionaries for sparse representation,” *IEEE Transactions on signal processing*, vol. 54, no. 11, pp. 4311–4322, 2006.
- [35] M. Elad and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries,” *IEEE Transactions on Image processing*, vol. 15, no. 12, pp. 3736–3745, 2006.
- [36] H. S. Mousavi and V. Monga, “Sparsity-based color image super resolution via exploiting cross channel constraints,” *IEEE Transactions*

on Image Processing, vol. 26, no. 11, pp. 5094–5106, 2017.

- [37] J. Shi, X. Ren, G. Dai, J. Wang, and Z. Zhang, “A non-convex relaxation approach to sparse dictionary learning,” in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 1809–1816.
- [38] S. Ravishankar and Y. Bresler, “Mr image reconstruction from highly undersampled k-space data by dictionary learning,” *IEEE transactions on medical imaging*, vol. 30, no. 5, p. 1028, 2011.
- [39] K. Cho, “Simple sparsification improves sparse denoising autoencoders in denoising highly corrupted images,” in *International Conference on Machine Learning*, 2013, pp. 432–440.
- [40] F. Agostinelli, M. R. Anderson, and H. Lee, “Adaptive multi-column deep neural networks with application to robust image denoising,” in *Advances in Neural Information Processing Systems*, 2013, pp. 1493–1501.
- [41] J. Xie, L. Xu, and E. Chen, “Image denoising and inpainting with deep neural networks,” in *Advances in neural information processing systems*, 2012, pp. 341–349.
- [42] H. C. Burger, C. J. Schuler, and S. Harmeling, “Image denoising: Can plain neural networks compete with bm3d?” in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2392–2399.

- [43] J. Mehta and A. Majumdar, “Rodeo: robust de-aliasing autoencoder for real-time medical image reconstruction,” *Pattern Recognition*, vol. 63, pp. 499–510, 2017.
- [44] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, “Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising,” *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.
- [45] C. Dong, C. C. Loy, K. He, and X. Tang, “Image super-resolution using deep convolutional networks,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2016.
- [46] K. Kulkarni, S. Lohit, P. Turaga, R. Kerviche, and A. Ashok, “Reconnet: Non-iterative reconstruction of images from compressively sensed measurements,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 449–458.
- [47] C. Wu and X.-C. Tai, “Augmented lagrangian method, dual methods, and split bregman iteration for rof, vectorial tv, and high order models,” *SIAM Journal on Imaging Sciences*, vol. 3, no. 3, pp. 300–339, 2010.
- [48] Q. Zhang and B. Li, “Discriminative k-svd for dictionary learning in face recognition,” in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2691–2698.
- [49] S. Ravishankar and Y. Bresler, “Learning sparsifying transforms,” *IEEE Transactions on Signal Processing*, vol. 61, no. 5, pp. 1072–

1086, 2013.

- [50] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Bm3d image denoising with shape-adaptive principal component analysis,” 2009.
- [51] S. Wang, L. Zhang, Y. Liang, and Q. Pan, “Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis,” in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2216–2223.
- [52] S. Ravishankar and Y. Bresler, “Efficient blind compressed sensing using sparsifying transforms with convergence guarantees and application to magnetic resonance imaging,” *SIAM Journal on Imaging Sciences*, vol. 8, no. 4, pp. 2519–2557, 2015.
- [53] M. J. Gangeh, A. K. Farahat, A. Ghodsi, and M. S. Kamel, “Supervised dictionary learning and sparse representation-a review,” *arXiv preprint arXiv:1502.05928*, 2015.
- [54] O. Bryt and M. Elad, “Compression of facial images using the k-svd algorithm,” *Journal of Visual Communication and Image Representation*, vol. 19, no. 4, pp. 270–282, 2008.
- [55] M. J. Gangeh, L. Sørensen, S. B. Shaker, M. S. Kamel, M. De Bruijne, and M. Loog, “A texton-based approach for the classification of lung parenchyma in ct images,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2010, pp. 595–602.

- [56] J. A. Bagnell and D. M. Bradley, “Differentiable sparse coding,” in *Advances in neural information processing systems*, 2009, pp. 113–120.
- [57] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. R. Bach, “Supervised dictionary learning,” in *Advances in neural information processing systems*, 2009, pp. 1033–1040.
- [58] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, “Robust face recognition via sparse representation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 2, pp. 210–227, 2009.
- [59] M. Yang, L. Zhang, J. Yang, and D. Zhang, “Metaface learning for sparse representation based face recognition,” in *Image Processing (ICIP), 2010 17th IEEE International Conference on*. IEEE, 2010, pp. 1601–1604.
- [60] Z. Jiang, Z. Lin, and L. S. Davis, “Learning a discriminative dictionary for sparse coding via label consistent k-svd,” in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 1697–1704.
- [61] —, “Label consistent k-svd: Learning a discriminative dictionary for recognition,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 11, pp. 2651–2664, 2013.
- [62] Y. Yankelevsky and M. Elad, “Graph-constrained supervised dictionary learning for multi-label classification,” in *Science of Electrical*

- Engineering (ICSEE), IEEE International Conference on the.* IEEE, 2016, pp. 1–5.
- [63] J. M. Bioucas-Dias, “A variable splitting augmented lagrangian approach to linear spectral unmixing,” in *Hyperspectral Image and Signal Processing: Evolution in Remote Sensing, 2009. WHISPERS’09. First Workshop on.* IEEE, 2009, pp. 1–4.
- [64] A. Martinez and R. Benavente, “The ar face database. cvc technical report 24,” Tech. Rep., june 1998.
- [65] Yale, “The extended yale face database b,” 2001. [Online]. Available: <http://vision.ucsd.edu/~iskwak/ExtYaleDatabase/ExtYaleB.html>
- [66] I. Manjani, S. Tariyal, M. Vatsa, R. Singh, and A. Majumdar, “Detecting silicone mask-based presentation attack via deep dictionary learning,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 7, pp. 1713–1723, 2017.
- [67] A. Sankaran, M. Vatsa, R. Singh, and A. Majumdar, “Group sparse autoencoder,” *Image and Vision Computing*, vol. 60, pp. 64–74, 2017.
- [68] Y. Liu, S. Zhou, and Q. Chen, “Discriminative deep belief networks for visual data classification,” *Pattern Recognition*, vol. 44, no. 10-11, pp. 2287–2296, 2011.
- [69] M. Mehdipour Ghazi and H. Kemal Ekenel, “A comprehensive analysis of deep learning based representation for face recognition,” in

- Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 34–41.
- [70] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, “Learning multi-label scene classification,” *Pattern recognition*, vol. 37, no. 9, pp. 1757–1771, 2004.
 - [71] G. Tsoumakas, I. Katakis, and I. Vlahavas, “Random k-labelsets for multilabel classification,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 7, pp. 1079–1089, 2011.
 - [72] M.-L. Zhang and Z.-H. Zhou, “A k-nearest neighbor based algorithm for multi-label classification,” in *Granular Computing, 2005 IEEE International Conference on*, vol. 2. IEEE, 2005, pp. 718–721.
 - [73] C.-K. Yeh, W.-C. Wu, W.-J. Ko, and Y.-C. F. Wang, “Learning deep latent space for multi-label classification.” in *AAAI*, 2017, pp. 2838–2844.
 - [74] G. W. Hart, “Nonintrusive appliance load monitoring,” *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.
 - [75] G. Tsoumakas and I. Vlahavas, “Random k-labelsets: An ensemble method for multilabel classification,” in *European conference on machine learning*. Springer, 2007, pp. 406–417.
 - [76] G. Tsoumakas and I. Katakis, “Multi-label classification: An overview,” *International Journal of Data Warehousing and Mining (IJDWM)*, vol. 3, no. 3, pp. 1–13, 2007.

- [77] J. Read, B. Pfahringer, G. Holmes, and E. Frank, “Classifier chains for multi-label classification,” *Machine learning*, vol. 85, no. 3, p. 333, 2011.
- [78] G. Nasierding and A. Z. Kouzani, “Comparative evaluation of multi-label classification methods,” in *Fuzzy Systems and Knowledge Discovery (FSKD), 2012 9th International Conference on*. IEEE, 2012, pp. 679–683.
- [79] K. Basu, V. Debusschere, S. Bacha, U. Maulik, and S. Bondyopadhyay, “Nonintrusive load monitoring: A temporal multilabel classification approach,” *IEEE Transactions on industrial informatics*, vol. 11, no. 1, pp. 262–270, 2015.
- [80] S. M. Tabatabaei, S. Dick, and W. Xu, “Toward non-intrusive load monitoring via multi-label classification,” *IEEE Transactions on Smart Grid*, vol. 8, no. 1, pp. 26–40, 2017.
- [81] J. Z. Kolter and M. J. Johnson, “Redd: A public data set for energy disaggregation research,” in *Workshop on Data Mining Applications in Sustainability (SIGKDD)*, San Diego, CA, vol. 25, no. Citeseer. Citeseer, 2011, pp. 59–62.
- [82] O. Parson, G. Fisher, A. Hersey, N. Batra, J. Kelly, A. Singh, W. Knottenbelt, and A. Rogers, “Dataport and nilmtk: A building data set designed for non-intrusive load monitoring,” in *2015 IEEE Global Conference on Signal and Information Processing (Global-SIP)*, Dec 2015, pp. 210–214.

- [83] J. Z. Kolter and T. Jaakkola, “Approximate inference in additive factorial hmms with application to energy disaggregation,” in *Artificial Intelligence and Statistics*, 2012, pp. 1472–1482.
- [84] R. Bonfigli, M. Severini, S. Squartini, M. Fagiani, and F. Piazza, “Improving the performance of the afamap algorithm for non-intrusive load monitoring,” in *2016 IEEE Congress on Evolutionary Computation (CEC)*, July 2016, pp. 303–310.
- [85] J. Kelly and W. Knottenbelt, “Neural nilm: Deep neural networks applied to energy disaggregation,” in *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*. ACM, 2015, pp. 55–64.
- [86] R. Rubinstein, A. M. Bruckstein, and M. Elad, “Dictionaries for sparse representation modeling,” *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1045–1057, 2010.
- [87] S. Li and H. Qi, “Compressed dictionary learning for detecting activations in fmri using double sparsity,” in *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2014, pp. 434–437.
- [88] G. Trigeorgis, K. Bousmalis, S. Zafeiriou, and B. W. Schuller, “A deep matrix factorization method for learning attribute representations,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 3, pp. 417–429, 2017.

- [89] M. Saar-Tsechansky and F. Provost, “Handling missing values when applying classification models,” *Journal of machine learning research*, vol. 8, no. Jul, pp. 1623–1657, 2007.
- [90] D. Jakovetić, J. M. Moura, and J. Xavier, “Linear convergence rate of a class of distributed augmented lagrangian algorithms,” *IEEE Transactions on Automatic Control*, vol. 60, no. 4, pp. 922–936, 2015.
- [91] R. G. Andrzejak, K. Lehnertz, F. Mormann, C. Rieke, P. David, and C. E. Elger, “Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state,” *Physical Review E*, vol. 64, no. 6, p. 061907, 2001.
- [92] A. Page, J. Turner, T. Mohsenin, and T. Oates, “Comparing raw data and feature extraction for seizure detection with deep learning methods,” pp. 284–287, 01 2014.
- [93] D. Wulsin, J. Gupta, R. Mani, J. Blanco, and B. Litt, “Modeling electroencephalography waveforms with semi-supervised deep belief nets: fast classification and anomaly measurement,” *Journal of neural engineering*, vol. 8, no. 3, p. 036015, 2011.
- [94] P. Thodoroff, J. Pineau, and A. Lim, “Learning robust features using deep learning for automatic seizure detection,” in *Machine learning for healthcare conference*, 2016, pp. 178–190.
- [95] A. Supratak, L. Li, and Y. Guo, “Feature extraction with stacked autoencoders for epileptic seizure detection,” in *2014 36th Annual*

International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE, 2014, pp. 4184–4187.

- [96] Y. Zhang, Y. Li, J. Sun, and J. Ji, “Estimates on compressed neural networks regression,” *Neural Networks*, vol. 63, pp. 10–17, 2015.
- [97] A. Fabisch, Y. Kassahun, H. Wöhrle, and F. Kirchner, “Learning in compressed space,” *Neural Networks*, vol. 42, pp. 83–93, 2013.
- [98] L. Vidya, V. Vivekanand, U. Shyamkumar, and D. Mishra, “Rbf-network based sparse signal recovery algorithm for compressed sensing reconstruction,” *Neural Networks*, vol. 63, pp. 66–78, 2015.
- [99] J. Zhao, Y. Lv, Z. Zhou, and F. Cao, “A novel deep learning algorithm for incomplete face recognition: Low-rank-recovery network,” *Neural Networks*, vol. 94, pp. 115–124, 2017.
- [100] S. Singh, V. Singhal, and A. Majumdar, “Deep blind compressed sensing,” *arXiv preprint arXiv:1612.07453*, 2016.
- [101] E. J. D. S. Luz, T. M. Nunes, V. H. C. De Albuquerque, J. P. Papa, and D. Menotti, “Ecg arrhythmia classification based on optimum-path forest,” *Expert Systems with Applications*, vol. 40, no. 9, pp. 3561–3573, 2013.
- [102] M. M. Al Rahhal, Y. Bazi, H. AlHichri, N. Alajlan, F. Melgani, and R. R. Yager, “Deep learning approach for active classification of electrocardiogram signals,” *Information Sciences*, vol. 345, pp. 340–354, 2016.

- [103] G. Koch, R. Zemel, and R. Salakhutdinov, “Siamese neural networks for one-shot image recognition,” in *ICML Deep Learning Workshop*, vol. 2, 2015.
- [104] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, “Fully-convolutional siamese networks for object tracking,” in *European conference on computer vision*. Springer, 2016, pp. 850–865.
- [105] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, “Deepface: Closing the gap to human-level performance in face verification,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1701–1708.
- [106] R. Fang, K. D. Tang, N. Snavely, and T. Chen, “Towards computational models of kinship verification,” in *Image Processing (ICIP), 2010 17th IEEE International Conference on*. IEEE, 2010, pp. 1577–1580.
- [107] S. Xia, M. Shao, and Y. Fu, “Kinship verification through transfer learning,” in *IJCAI Proceedings-international joint conference on artificial intelligence*, vol. 22, no. 3, 2011, p. 2539.
- [108] J. Lu, X. Zhou, Y.-P. Tan, Y. Shang, and J. Zhou, “Neighborhood repulsed metric learning for kinship verification,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 2, pp. 331–345, 2014.

- [109] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, “Multi-pie,” *Image and Vision Computing*, vol. 28, no. 5, pp. 807–813, 2010.
- [110] O. M. Parkhi, A. Vedaldi, A. Zisserman *et al.*, “Deep face recognition.” in *BMVC*, vol. 1, no. 3, 2015, p. 6.
- [111] H. Yan, J. Lu, and X. Zhou, “Prototype-based discriminative feature learning for kinship verification,” *IEEE transactions on cybernetics*, vol. 45, no. 11, pp. 2535–2545, 2015.
- [112] J. Lu, J. Hu, and Y.-P. Tan, “Discriminative deep metric learning for face and kinship verification,” *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4269–4282, 2017.
- [113] H. Dibeklioglu, “Visual transformation aided contrastive learning for video-based kinship verification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2459–2468.
- [114] N. Kohli, M. Vatsa, R. Singh, A. Noore, and A. Majumdar, “Hierarchical representation learning for kinship verification,” *IEEE Transactions on Image Processing*, vol. 26, no. 1, pp. 289–302, 2017.
- [115] J. Hu, J. Lu, and Y.-P. Tan, “Sharable and individual multi-view metric learning,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 9, pp. 2281–2288, 2018.
- [116] D.-A. Huang and Y.-C. Frank Wang, “Coupled dictionary and feature space learning with applications to cross-domain image synthe-

- sis and recognition,” in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 2496–2503.
- [117] S. Xiang, G. Meng, Y. Wang, C. Pan, and C. Zhang, “Image deblurring with coupled dictionary learning,” *International Journal of Computer Vision*, vol. 114, no. 2-3, pp. 248–271, 2015.
- [118] P. Dou, Y. Wu, S. K. Shah, and I. A. Kakadiaris, “Monocular 3d facial shape reconstruction from a single 2d image with coupled-dictionary learning and sparse coding,” *Pattern Recognition*, vol. 81, pp. 515–527, 2018.
- [119] M. Singh, S. Nagpal, M. Vatsa, R. Singh, and A. Majumdar, “Identity aware synthesis for cross resolution face recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 479–488.
- [120] K. Zeng, J. Yu, R. Wang, C. Li, and D. Tao, “Coupled deep autoencoder for single image super-resolution,” *IEEE transactions on cybernetics*, vol. 47, no. 1, pp. 27–37, 2017.
- [121] K. Gupta, B. Bhowmick, and A. Majumdar, “Motion blur removal via coupled autoencoder,” in *Image Processing (ICIP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 480–484.
- [122] Z. Li and J. Tang, “Weakly supervised deep matrix factorization for social image understanding,” *IEEE Transactions on Image Processing*, vol. 26, no. 1, pp. 276–288, 2017.

- [123] B. Wen, S. Ravishankar, and Y. Bresler, “Video denoising by on-line 3d sparsifying transform learning,” in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 118–122.
- [124] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, “Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1874–1883.
- [125] J. S. Ren and L. Xu, “On vectorization of deep convolutional neural networks for vision tasks.” in *AAAI*, 2015, pp. 1840–1846.
- [126] W.-Z. Shao, H.-S. Deng, Q. Ge, H.-B. Li, and Z.-H. Wei, “Regularized motion blur-kernel estimation with adaptive sparse image prior learning,” *Pattern Recognition*, vol. 51, pp. 402–424, 2016.
- [127] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang, “Deep networks for image super-resolution with sparse prior,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 370–378.
- [128] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. P. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, “Photo-realistic single image super-resolution using a generative adversarial network.” in *CVPR*, vol. 2, no. 3, 2017, p. 4.
- [129] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,”

arXiv preprint, 2017.