

School of Electrical Engineering, Computing and Mathematical Sciences

**Novel Deep Learning Techniques For Computer Vision and Structure
Health Monitoring**

Chathurdara Sri Nadith Pathirage

**This thesis is presented for the Degree of
Doctor of Philosophy
of
Curtin University**

August 2018

To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgement has been made. This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.



Nadith Pathirage

03.08.2018

Date

Abstract

Complexity theory of circuits strongly suggests that deep architectures can be much more efficient (sometimes exponentially) than shallow architectures (i.e. Kernel), in terms of computational elements required to represent complicated functions. Deep multi-layer neural networks have many levels of non-linearities allowing them to compactly represent highly non-linear and highly-varying functions. Much research has been devoted to learning algorithms for deep architectures such as Deep Belief Networks and stacks of auto-encoder variants, with impressive results obtained in several areas, mostly on vision and language datasets. The best results obtained on supervised learning tasks involve an unsupervised learning component, usually in an unsupervised pre-training phase. Even though these new algorithms have enabled training deep models, many questions remain as to the nature of this challenging learning problem. Despite the vast development of deep learning approaches, most of them emphasize on the applications in the computer vision domain such as fields recognition, face recognition, face verification, etc. Applications in other engineering domain are scarce. Typically, a deep model is focused on one specific task but not many tasks in vastly different domains. There is almost no work done on how a deep learning model could be built as a generic framework for both computer vision and civil engineering applications. The aim of this thesis is to build a generic deep learning framework utilizing the autoencoders and enhance the performance of all its applications in both the computer vision and civil engineering domains. We proposed novel deep learning techniques based on the simplest deep learning building block, autoencoder, to address machine learning problems in widely different domains.

Firstly, we propose a basic deep learning framework that is generic to both the computer vision and civil engineering domains. We emphasize on the necessity for the division of components to address non-linear dimensionality reduction and relationship learning tasks efficiently. The basic framework can improve the performance of typical machine learning problems in the respective domains when there are no irregularities (noise) in data. Secondly, we propose an extension to the basic deep learning framework introduced above to enhance the performance under various types of noise. A carefully analyzed regularization scheme is proposed and more freedom in the framework is introduced for deeper layers and more hidden nodes. The experiments reveal the high performance on both the visual data and numerical data with various types of noise. Thirdly, we propose novel cost formulations to perform discriminant analysis to overcome the typical issues that exist in linear discriminant analysis techniques such as LDA. Furthermore, we embed the novel cost formulation to a deep architecture to perform deep discriminant analysis on classification tasks. The effects of outliers and mitigating strategies with respect to the novel

cost formulations are discussed in detail to build a robust deep learning framework. Lastly, we propose a few generic strategies to design complex deep learning systems to address complicated problems such as face recognition under arbitrary occlusions. We show that a typical deep learning model would fail to address all the non-linearities in a complex problem as a whole. However, a methodical approach to divide the complex objective into many sub-objectives would ease the task of learning highly complex non-linear mappings. Therefore, a pool of deep learning models where each model is utilized on such sub-objective can work together in a system towards the complex objective to solve the problem successfully. The strategies to design such complex deep learning system perform a crucial role in this regard. The experiments outperform the state-of-the-art methods demonstrating the necessity of the study on complex deep learning system designs.

Building a generic framework for both the regression and classification tasks in the respective fields is the focus of this work. Combinedly with complex deep learning system designs, the proposed generic framework can be effectively utilized to solve complicated problems without the need of engineering the model to be problem specific in widely different domains. The experiment results demonstrate significant improvements of all the proposed techniques towards accuracy and efficiency.

Acknowledgments

I would like to express my sincere gratitude to the following people for the great guidance and support extended to me during the long journey of my PhD study. This thesis would not be possible without them.

- First and foremost, I owe my deepest gratitude to Prof. Kevin Fynn (HOS) for providing me with an opportunity to follow the PhD programme at Curtin University after my first degree in Information Technology with 1st class Honours.
- I wish to extend a special acknowledgment and sincere appreciation and gratitude to my supervisor, Associate Professor Ling Li and co-supervisor Associate Professor Wanquan Liu for their constant encouragement, guidance, and support throughout my PhD study period. I am deeply grateful for their patience and compassion in teaching me how to be independent and reliable not only as a researcher but also as a person in society. This study has been a very valuable learning experience in my life.
- I would like to express a special thanks to the School of Electrical Engineering and Computing and Prof. Hong Hao and Dr. Jun Li for providing a sponsorship for my PhD Study programme. I would also like to thank Prof. Hao and Dr. Li for their guidance in Civil Engineering field.
- I wish to thank Dr. Antoni Liang for all the support given to me throughout my PhD study including this thesis.
- I also greatly appreciate the support and encouragement of my brother Mr. Damith Pathirage, sister Anushika Pathirage and my uncle Mr. Dantha Peris.
- I also would like to thank all my fellow research students (Master and PhD) and researchers (Ke Fan, Xin Zhang, Qilin Li, Mustafa M. M. Alrjebi, Ruhua Wang) for all the inspirations and support.
- Lastly, I would like to thank our administrative officers Mary Simpson, Mary Mulligan, Cindy Wong and Sucy Leong for the assistance provided on the paper works and scholarships.

I wish to dedicate this study to my parents Mr. Thilak Pathirage and Mrs. Dharshani Pathirage who always guided me in my life and inspired my academic career and stood by me as pillars of strength.

Published Work

Several academic publications included in this thesis over the course of this PhD study are listed in chapter order:

- Pathirage, C. S. N., Li, L., Liu, W., and Zhang, M. (2016) Stacked Face De-Noising Autoencoders for Expression-Robust Face Recognition. *International Conference on Digital Image Computing: Techniques and Applications, (DICTA)*. (Chapter 3)
- Pathirage, C. S. N., Jun Li, Ling Li, Hong Hao, Wanquan Liu. (2017) Deep Autoencoder Model for Pattern Recognition in Civil Structural Health Monitoring. *International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering (QR2MSE)* (Chapter 3).
- Pathirage, C. S. N., Jun Li, Ling Li, Hong Hao, Wanquan Liu. (Accepted) Application of deep autoencoder model for structural condition monitoring. *Journal of Systems Engineering and Electronics*. (Chapter 3)
- Antoni Liang, Pathirage, C. S. N., Chenyu Wang, Wanquan Liu, Ling Li, and Jinming Duan. (2015) Face Recognition Despite Wearing Glasses. *International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. pages 1-8. (Chapter 4)
- Pathirage, C. S. N., Jun Li, Ling Li, Wanquan Liu et al. (2018) Autoencoder based Framework for Structural Damage Identification. *Engineering Structures*. (Chapter 4)
- Li, L., Hao, H., Li, J., Pathirage, C. S. N., Liu, W., Wang, R. (2018) Numerical Studies on Using Deep Sparse Autoencoders for Damage Identification of Structures. Invited talk in *The 7th World Conference on Structural Control and Monitoring (7WCSCM)* (Chapter 4).
- Pathirage, C. S. N. et al. (Under Review) Deep Neural Networks with Sparse Autoencoders for Structural Health Monitoring. *International Journal Of Structural Health Monitoring*. (Chapter 4)
- Pathirage, C. S. N., Ling Li, Wanquan Liu. (2017) Discriminant auto encoders for face recognition with expression and pose variations. *International Conference on Pattern Recognition*. pages 35123517. (Chapter 5)

- Pathirage C. S. N., Ling Li, Wanquan Liu. (Under Review) Discriminant Feature Extraction based on Correntropy Criterion. *Journal of Selected Topics in Signal Processing*. (Chapter 5)
- Alrjebi, M. M., Pathirage, C. S. N., Wanquan Liu, and Ling Li. (2017) Face recognition against occlusions via colour fusion using 2d-mcf model and src. *Pattern Recognition Letters*. (Chapter 6)
- Pathirage C. S. N., Ling Li, Wanquan Liu. (Under Review) Stacked Autoencoder Framework for Face Recognition Against Occlusions. *Pattern Recognition*. (Chapter 6)

Contents

1	Introduction	1
1.1	Research Gaps and Aims	2
1.2	Thesis Structure and Contributions	4
2	Background	7
2.1	Machine Learning Domains	9
2.1.1	Supervised Learning	9
2.1.2	Unsupervised Learning	9
2.1.3	Transfer Learning	10
2.2	Linear Techniques	10
2.2.1	Principal Component Analysis (PCA)	10
2.2.2	Data Whitening	11
2.2.3	Linear Discriminant Analysis (LDA)	12
2.2.4	Sparse Representation Classifier (SRC)	14
2.3	Neural Networks	15
2.4	Deep Learning	16
2.4.1	Autoencoder	17
2.4.2	Denoising Autoencoder	18
2.4.3	Sparse Autoencoder	18
2.4.4	Deep Autoencoder	20
2.4.5	Activation Functions	22
2.5	Optimization Methods	25
2.5.1	Stochastic Gradient Descent	25
2.5.2	Gradient Descent with Momentum	27
2.5.3	Conjugate Gradient Algorithms	27
2.6	Regularization	28
2.6.1	Bias vs Variance	28
2.6.2	Norm Penalties	29
2.6.3	Cross validation	30
2.6.4	Feature Normalization	31
2.7	Face Recognition	31
2.7.1	Databases	32
2.8	Structure Health Monitoring	37
2.8.1	Numerical Studies	38
2.8.2	Experimental Studies	44

2.9	Summary	52
3	Basic Framework For Pattern Recognition	53
3.1	Deep Learning on Non-Linear Problem Domain	54
3.2	Autoencoders as Generic Building Blocks	55
3.3	Proposed Framework (AutoNet)	56
3.3.1	Dimensionality Reduction	57
3.3.2	Relationship Learning	59
3.3.3	Fine-Tuning	60
3.4	Applications	60
3.4.1	Computer Vision Application - Face Recognition	60
3.4.2	Civil Engineering Application - Structure Health Monitoring	73
3.5	Summary	80
4	Robust Framework For Pattern Recognition	82
4.1	Introduction	82
4.1.1	Generalized Machine Learning	82
4.1.2	Regularization	86
4.2	Proposed Extended Framework	87
4.2.1	Extended Dimensionality Reduction Component	88
4.2.2	Extended Relationship Learning Component	89
4.2.3	Effective Enforcement Of Sparsity	90
4.2.4	Constraining the Complexity of Framework	91
4.3	Applications	92
4.3.1	Computer Vision Application - Glass Removal For Face Recognition and Verification	93
4.3.2	Civil Engineering Application - Structure Health Monitoring	100
4.4	Summary	114
5	Non-linear Discriminant Analysis	117
5.1	Introduction	117
5.1.1	Why Discrimination?	117
5.2	Outliers in Discriminant Analysis	119
5.3	Proposed Error Criteria	120
5.3.1	Non-Linear Discriminant Error Criterion	121
5.3.2	Discriminant Co-Entropy Criterion (DCC)	121
5.4	Deep Discriminant Analysis (DDA) Framework	128
5.5	Applications	132
5.5.1	DDA Framework For Face Recognition	133
5.5.2	Discriminant Coentropy in Classification Context	143

5.6	Summary	151
6	Complex Deep Learning Systems	154
6.1	Introduction	154
6.2	Proposed System Design	156
6.2.1	Deep Model Fusion System	158
6.3	Applications	163
6.3.1	Deep Model Fusion System Design For Occlusion Removal	163
6.4	Summary	171
7	Conclusions and Future Directions	173
7.1	Future Study	175

List of Figures

2.1	Machine is pipeline for regression or classification problems.	8
2.2	A visualization of data whitening process.	11
2.3	Typical shallow autoencoder with encoder and decoder components.	17
2.4	Typical deep autoencoder with multiple encoder and decoder components.	20
2.5	Example of k-fold cross validation.	31
2.6	13 image variations on 2 sessions for each participant on AR database.	33
2.7	Examples of all the facial poses and expressions on CMU multiPIE database. . .	33
2.8	range data (depth) and 2D images from CurtinFaces database.	34
2.9	Images with variety on (from top) pose, accessories, expression, and illumination from CAS-PEAL-R1.	35
2.10	Images of ORL database.	36
2.11	Images of MNIST database with many variations.	37
2.12	Laboratory model and dimensions of the steel frame structure.	40
2.13	Finite element model of the steel frame structure.	41
2.14	The first seven measured frequencies and mode shapes of the frame structure. . .	42
2.15	A steel frame model in the laboratory.	45
2.16	Mode shapes before and after updating.	46
2.17	The experimental testing model.	48
2.18	Dimensions of the testing model and the sensor placement.	49
2.19	Finite element model of the testing bridge.	50
2.20	Updated frequencies from the finite element model.	51
2.21	Introduced cracks in the web elements of the tested bridge.	51
3.1	Highlevel view of the proposed framework.	56
3.2	Architecture of the proposed framework.	58
3.3	Transformation functions that are learned by the proposed AutoNet framework via its hidden layers and the formation of the low dimensional feature space.	61
3.4	Recovered neutral expressions faces against the noisy version of it. The first row of faces with expressions are neutralized and shown in the second row.	62
3.5	The proposed framework where $f_1 \in \mathbb{R}^{50}$ denotes low dimensional noisy feature learnt at layer 1, while $f_2 \in \mathbb{R}^{50}$ denotes the noiseless feature learnt at layer 2 in the observed low dimensional space. We halves the image space by 50% to constraint the framework to learn an effective low dimensional feature.	63

3.6	Progressive pre-training of respected layers to achieve better initial weights prior to the training phase. Left figure denotes the non-linear dimension reduction layer pre-training while de-noising layer pre-training is shown in the right figure. . . .	64
3.7	Images with different expressions and their corresponding indices.	65
3.8	Data splitting in Test Case 3. Images 1, 2, 4-6 from 75 identities were taken for training. Images of the same indices (1, 2, 4-6) from the remaining 25 identities were used for validation. Images 3 of the respective 75 identities were used for testing.	66
3.9	Results of the tests performed with 75 identities in AR database.	67
3.10	The data splitting of a cross identity test case. All images (1-6) from 50 identities were taken for training whereas the other 50 identities were split (25 identities each) for validation and testing. Each test case concerns one expression for recognition accuracies. Shown in the figure is when Images 3 were used for testing. . .	68
3.11	Results of the experiments performed on cross subject arrangements.	69
3.12	Data splitting in Test Case 3. Images 1, 2, 4-6 from 125 identities were taken for training. Images of the same indices (1, 2, 4-6) from the remaining 25 identities were used for validation. Images 3 of the respective 125 identities were used for testing.	70
3.13	Results of the experiments performed on the combined database.	71
3.14	Results of the experiments performed on the cross-database arrangement.	72
3.15	Proposed autoencoder based framework.	74
3.16	Pre-training the proposed autoencoder based framework.	75
3.17	Dimensionality reduction component with decoder.	77
3.18	An example of single damage identification.	79
3.19	An example of multi-damage identification.	80
4.1	Images with different expressions and their corresponding indices.	85
4.2	Highlevel view of the proposed extended framework.	87
4.3	Architecture of the proposed extended framework.	88
4.4	Complete system pipeline for glasses detection and removal.	95
4.5	Facial recognition with classification approaches PCA, LDA, and SRC. This result proves that removing presence of glasses improves the facial recognition rate. . .	97
4.6	ROC curves on thin glasses with PCA, LDA, SRC respectively.	98
4.7	ROC curves on thick glasses with PCA, LDA, SRC respectively.	99
4.8	Proposed extended autoencoder based framework.	102
4.9	Pretraining on proposed extended autoencoder based framework.	103
4.10	Damage identification results of a single damage case from AutoDNet and SAF for Scenario 1.	106

4.11	Damage identification results of a multiple damage case from AutoDNet and SAF for Scenario 1.	106
4.12	Damage identification results of a continuously distrubted multiple damage case from AutoDNet and SAF for Scenario 1.	107
4.13	Damage identification results of a multiple damage case from AutoDNet and SAF for Scenario 2.	108
4.14	Damage identification results of another multiple damage case from AutoDNet and SAF for Scenario 2.	109
4.15	Damage identification results of a single damage case from AutoDNet and SAF for Scenario 3.	110
4.16	Damage identification results of a multiple damage case from AutoDNet and SAF for Scenario 3.	110
4.17	Damage identification results of a single damage case from AutoDNet and SAF for Scenario 4.	112
4.18	Damage identification results of a multiple damage case from AutoDNet and SAF for Scenario 4.	112
4.19	Damage identification results from AutoDNet and SAF in the experimental study.	114
5.1	The set of outliers that can potentially affect the estimation of class means. Black line indicates the robust projection (W) that is learnt via DCC. Triangles and Squares indicate the samples that belong to two different classes.	123
5.2	Highlevel view of the proposed discriminant framework.	129
5.3	Architecture of the proposed discriminant framework.	130
5.4	Progressive pre-training of respective layers to achieve better initial weights prior to the training phase. Layers 1 and 2 perform the progressive non-linear dimension reduction while de-noising happens at Layer 3. The non-linear discriminant criteria is utilized on the reconstructed neutral face (DDA layer). The mean face of class i is denoted by m_i	133
5.5	DDA framework where $\{h_i^r\}_1, \{h_i^r\}_2$ is the low dimensional noisy feature learned at Layer 1 and 2 for the r_{th} input image of class i , which is c_i^r . $\{h_i^r\}_3$ denotes the noise-less feature learned at Layer 3 (de-noising layer) in the observed low dimensional space. $g_3(\cdot)$ represents the decoder function. Hence the discriminant layer where $\{h_i^r\}_d \in \mathbb{R}^{class\ count-1}$ is shown as the right most layer.	134
5.6	Different expressions and the corresponding indices.	135
5.7	Results of the same identity experiments on AR database.	137
5.8	Results of the cross identity experiments on AR database.	138
5.9	Results of the cross database experiments on AR and Curtin database.	139
5.10	Images with different poses and the indices.	141

5.11	(a) Samples in toy set and 1D subspaces. Solid lines indicate, Red: LDA subspace, Amber: LDA-L1 subspace, Black: DCC subspace. (b) Results of LDA-L2 projection. (c) Results of LDA-L1 projection. (d) Results of DCC projection. . .	145
5.12	Different expressions and occlusions	146
5.13	Training sets with corruptions	147
5.14	Testing set	148
5.15	Images with different poses.	149
5.16	Visualizing MNIST digits {3, 8, 9} with TSNE. (a) Digits in image space. (b) Representations in subspace.	150
5.17	Testing set	151
5.18	Rotation of the data in the image space. Rotations are performed within the sphere denoted by L2 norm. Hence the rotations are performed with respect to class centers.	151
6.1	Illustration of a patch-based deep learning system.	155
6.2	Illustration of system designs for complex problems.	157
6.3	Illustration of sub-system design with autoencoder based models.	157
6.4	Single autoencoder recognition performance on occluded images of different magnitude.	157
6.5	Illustration of a deep model fusion system.	159
6.6	Illustration of the occluded face manifold, and the limited but tractable goals that are set during the training of each layer of ANF.	160
6.7	(a) Examples of reconstructed faces of MultiPIE database with simulated occlusion; (b) Examples of reconstructed faces of AR database with real occlusion. The top row shows the faces with no occlusion and the bottom row shows the reconstructed faces.	160
6.8	Illustration of simulated occlusion in various directions. The alias for each occlusion is shown at the bottom of each column.	161
6.9	Illustration of simulated occlusion in various directions and magnitudes for a frontal face. Rows from top shows the bottom, left, top, right, middle (horizontal), middle (vertical) occlusions with varying ratios respectively. The occlusion ratios are shown at the bottom of each column.	162
6.10	Frontal face images with their corresponding indices.	166
6.11	Face image subset of the AR database including real occlusion.	168
6.12	Face image subset of Curtin database including real occlusion.	169

List of Tables

2.1	Measured and updated frequencies before and after updating.	43
2.2	MAC Values before and after updating.	43
2.3	Measured and updated frequencies before and after updating.	47
3.1	Results of the tests performed with 75 identities in AR database.	66
3.2	Results of the experiments performed on cross subject arrangements.	68
3.3	Results of the experiments performed on the combined database.	71
3.4	Results of the experiments performed on the cross-database arrangement.	73
3.5	Evaluation results of the proposed framework with the decoder for reconstruction of original feature.	78
3.6	Evaluation results of the proposed framework.	79
4.1	Face verification rate at 0.1% False Acceptance Rate (FAR) before and after glasses removal.	100
4.2	Performance evaluation results for Scenario 1 in the numerical study.	105
4.3	Performance evaluation results for Scenario 2 in the numerical study.	108
4.4	Performance evaluation results for Scenario 3 in the numerical study.	109
4.5	Performance evaluation results for Scenario 4 in the numerical study.	111
4.6	Performance evaluation results in the experimental study.	114
5.1	Results of the same identity experiments on AR database.	136
5.2	Results of the cross identity experiments on AR database.	139
5.3	Results of the cross database experiments on AR and Curtin database.	140
5.4	Results of the cross identity experiments on MultiPIE.	142
5.5	Results of the cross database experiments on MultiPIE and Curtin database.	142
5.6	Recognition rates on AR dataset.	146
5.7	Recognition rates on AR dataset.	147
5.8	Recognition rates of Experiment 1 on ORL dataset.	148
5.9	Recognition rates of Experiment 2 on ORL dataset.	148
5.10	Recognition rates of Experiment 3 on ORL dataset.	148
5.11	Recognition rates on Multi-PIE dataset.	149
5.12	Recognition rates on MNIST dataset.	150
6.1	Accuracy on Cross Identity.	167
6.2	Accuracy on Cross Identity with various occlusion orientations.	167
6.3	Accuracy on the AR Database.	169
6.4	Accuracy on the Curtin Database.	169

6.5 Accuracy on Cross Database.	171
---	-----

Chapter 1

Introduction

Deep learning is a subset of machine learning in Artificial Intelligence (AI) that has networks capable of learning feature hierarchies with features from higher levels of the hierarchy formed by the composition of lower level features. Deep learning includes methods for a wide array of deep architectures, including neural networks with many hidden layers, and graphical models with many levels of hidden variables. Theoretical results state that in order to learn the kind of complicated functions that can represent high-level abstractions (e.g., in vision, language processing, and other AI-level tasks), one may need deep architectures.

Irrespective of the serious challenges in training deep models with many layers of adaptive parameters, these models significantly out-perform the shallow competitors and often match or beat the state-of-the-art. Accumulating evidence in challenging AI-related tasks such as computer vision, natural language processing, robotics, information retrieval, etc. show the tremendous power in utilizing deep learning in various problem domains.

Despite the usefulness of deep architectures, the objective function of almost all instances of deep learning is a highly non-convex function of the parameters. It leads to the potential for many distinct local minima in the model parameter space. The primary difficulty with many distinct local minimas is that these minimas provide equivalent generalization errors. For deep architectures, the standard training schemes (based on random initialization) tend to place the parameters in regions of the parameters space that generalize poorly (Bengio *et al.*, 2007b). Ranzato *et al.* (2006) and Bengio *et al.* (2007a) proposed effective training strategies for deep architectures with autoencoders which is greedy layer-wise unsupervised pre-training, followed by fine-tuning where the unsupervised pre-training sets the stage for a final training phase (fine-tuning).

Although the improvement in performance of trained deep models offered by the pre-training strategy is impressive, not much work has been done on how autoencoders can be used to build a generic framework for both computer vision and engineering domains. Many deep learning models based on autoencoders are focused on one specific problem definition, not applicable in widely different domains. The concepts such as the curse of dimensionality and non-linear manifold learning are not carefully investigated in depth with respect to the autoencoder based models, especially when both the visual data in computer vision domain and numerical data in engineering

domain are concerned. For labeled data, an investigation is required in learning a discriminant deep latent space where with-in class distances and in-between class distances are critical considerations. Moreover, the inherent drawbacks of most commonly used squared error cost function in deep learning models need to be reinvestigated regarding its properties and efficiency toward dealing with large outliers. Lastly, for complex problems where a single stack deep autoencoder model is not capable of tackling the non-linearities in the problem domain, an investigation into an elegant and flexible deep learning system designs must be drawn attention.

1.1 Research Gaps and Aims

The research gaps presented below were observed after conducting in-depth investigations on the literature review. We aim our research to specifically address each of these gaps.

1. Many machine learning approaches perform quite well on a wide variety of problems. However, the non-linear characteristics in important real-life problems restrict those approaches due to their fundamental limitations of capacity. The non-linearity in the learning domain of the problem has always remained a challenge in machine learning for decades. Feature learning linear techniques and shallow models such as kernel methods with convex loss functions were able to address various aspects of different problem domains. However, it says nothing about the efficiency of the representation. For example, there is empirical and theoretical evidence that shallow architectures cannot implement invariant visual recognition tasks efficiently in Bengio *et al.* (2007b). Hence they have not succeeded in solving the central problems in AI. The development of deep learning was motivated in part by the failure of traditional algorithms to generalize well in such non-linear problem domains. Deep architectures that can incorporate the non-linearity observed in the problem domain, promote efficient feature learning while being consistent on performance, addressing challenging dilemmas in the respective field. Deep learning is more efficient for representing classes of functions in general, particularly those involved in visual recognition. Most deep learning architectures are developed in a problem specific manner hence may not necessarily be utilized in different tasks, especially when the application domain vastly differ such as computer vision and civil engineering. We aim to develop a generic framework based on the simplest learning module in deep learning to overcome this issue while not compromising the beneficial properties of deep learning.
2. Challenge of generalizing to new examples becomes exponentially more difficult when working with high-dimensional data due to the freedom of dimensionality that can fit noise. Thus the mechanisms used to achieve generalization in typical machine learning approaches

are insufficient to learn complicated functions in high-dimensional spaces. Such spaces also often impose high computational costs. Many deep learning models perform well on the training data due to its immense capacity. However, it is difficult to make reasonable predictions for unseen data due to the effects of overfitting where the models learn details of training data too well including the noise. We aim to develop a generic framework improved on generalization aspects for better performance and stability in both the computer vision and civil engineering domains. The proposed generic framework is favorable to have mechanisms in place to discourage complex or extreme explanations of the world even if they fit what has been observed better (training data) while benefitting from the high dimensional data at the same time. The intuition is that such explanations are unlikely to generalize well. They may happen to explain a few samples from the past, but this may be due to the effect of noise in the samples. We need an efficient parameterization of the class of functions that we need to build intelligent machines. Effective means of generalization need to be developed to enhance the stability of the proposed generic framework to perform better in noisy data encounters.

3. Machine learning algorithms discover patterns in data that lead to actionable insights. At a high level, these algorithms can be mainly classified into two groups based on the way they "learn" about data to make predictions: unsupervised and supervised learning. As mostly seen in deep learning, unsupervised learning is where the machines can learn to identify complex processes and patterns without any explicit guidance from a human. It lacks the power of discrimination for classification tasks such as face recognition. Furthermore, unsupervised learning may also be affected by large outliers that exist in data due to the most commonly used squared error loss function. In contrast, supervised learning assumes that the algorithm's possible outputs are already known, and the training data for the algorithm is already labeled with correct answers. While a supervised classification algorithm learns to ascribe inputted labels to the corresponding samples, its unsupervised counterpart will look at the similarities between the samples and separate them into groups accordingly, assigning its new label to each group. It is essential to consider the structure and volume of the data along with the nature of the problem to choose either a supervised or unsupervised machine learning algorithm. A well-rounded, deep learning model will use both types of algorithms to build predictive data models that help stakeholders make decisions. Many of the deep models in the literature do not follow a semi-supervised learning approach to utilize the class information that is already available in the labeled dataset. We aim to develop novel robust cost formulations extending the generic deep learning framework to act as a hybrid framework that performs both unsupervised and supervised training along the way. Furthermore, these cost formulations promote deep discriminative latent space that is highly suitable for classification tasks.

4. Many deep learning models can be directly utilized in a wide variety of tasks to perform efficient feature learning. However, certain problems exist that need careful attention to be utilized by a deep learning model. Since the complexities of a problem can change from simple to extreme, it is necessary to sub-divide the complex objectives into simpler and tractable sub-objectives. For example, consider the face recognition against occlusion problem. Occlusion can occur in various directions in different magnitudes thus introduce additional complexities that a typical single deep learning network would not be able to tackle successfully. It is favorable to divide the problem into manageable sub-objectives to achieve the global non-linearity involved in the problem domain. These sub-objectives can be addressed via a set of trained deep networks working together towards one complex global objective. As far as we know, in the case of certain complex problem such as face recognition against occlusion, there is no methodical approach to break the complex objective into such sub-objectives to ease the task of learning highly complex non-linear mappings. We aim to develop a few generic strategies for the design of complex deep learning systems.

1.2 Thesis Structure and Contributions

The list below briefly describes the content of each chapter in this thesis along with its contributions.

- Chapter 2: We introduce some preliminary knowledge related to our proposed approaches. We begin with pattern recognition and its evolution to machine learning. Three main types of learning are discussed in machine learning while providing useful insights on popular linear and non-linear techniques. The history of the neural network and the reasons behind its failure are discussed in detail to step towards deep learning which is currently a popular topic in the machine learning research field. The anatomy of the autoencoder model and its advancements are explained in great depths to outline the building block of the novel deep learning techniques introduced in this thesis. The other relevant key aspects of deep learning such as optimization and regularization strategies are exploited to further support claims made throughout the thesis. Moreover, a general introduction on face recognition and structural health monitoring problem domains is provided to make aware of the contextual information that is essential for the experiments performed in the respective chapters.
- Chapter 3: We propose a carefully designed novel deep learning framework based on the autoencoder model that can be utilized both in the computer vision and the civil engineering domains. The framework is comprised of two components focused on two different objectives. The first component performs the dimensionality reduction of the input features

while the second does the relationship learning from the learned low dimensional representation to the output. The generic framework can be utilized to model classification problems such as face recognition under various expressions and regression tasks like mapping inputs to outputs. Experimental results reveal the significant improvement on both accuracy and easy adaptivity of the framework against some state-of-the-art in the respective fields. The effectiveness of the dimensionality reduction and the relationship learning component are then evaluated individually in civil engineering domain to show the importance of both the components of the proposed framework.

- Chapter 4: We propose an extension to the basic framework presented in Chapter 3 to perform robust feature learning against the varying degree of noise in data on which the basic framework would fail. The principal advantages of the basic framework such as the generic nature of easy adaptability and flexibility in different application domains are preserved in the proposed extended framework. Also, efficient regularization strategies are introduced to perform better with data acquired in challenging environments. Experiments are performed in both the computer vision and civil engineering domain to validate the effectiveness of the proposed extended framework. The results reveal a significant improvement on both accuracy and effectiveness against some state-of-the-art in the respective fields. The consideration of sparse constraints in the learning domain is also exploited to showcase the improvements that the proposed framework could achieve thus making it a complete, robust framework towards machine learning problems in computer vision and civil engineering domains.
- Chapter 5: We propose a novel framework (DDA) based on the basic framework introduced above to perform non-linear discriminant analysis for labeled data. The proposed DDA framework utilizes the class information of the data to learn a latent discriminant space in the newly introduced discriminant analysis component. It is a generic framework to perform non-linear discriminant analysis effectively, especially in classification contexts. Experiments are conducted on various face related problems such as face recognition across expression and pose variations to evaluate the effectiveness of the non-linear discriminant analysis performed by the proposed DDA framework. Furthermore, we introduce an alternative cost formulation named as Discriminant Co-entropy Criterion (DCC) to perform robust discriminant analysis when there are large outliers in the data. An extensive set of experiments are performed to validate the robustness of the proposed alternative cost formulation against various outliers.
- Chapter 6: We exploit an effective design choice for complex deep learning systems where the problem complexity could be further divided into a set of low complex objectives. We investigate the feasibility of combining the frameworks proposed in previous chapters to

design a complete integrated deep learning system to solve complex problems in computer vision domain. A stepwise process is discussed to divide the complexity involved in a task generally and two major design concepts are introduced for complex deep learning systems. Face recognition against occlusion is considered a complex problem since occlusion can occur in many different forms. The proposed system design shows its high effectiveness towards dealing with occlusion in various forms. The experiments performed with both the genuine and simulated occlusions show its superiority against the other state-of-the-art methods.

- Chapter 7: The whole thesis is concluded and some potential future directions are addressed.

Chapter 2

Background

A highly sophisticated skill set has been developed by humans to sense the environment and take actions according to what is observed such as recognizing an object or face, understanding spoken words, reading handwriting, etc. Pattern recognition algorithms are often influenced by the knowledge of how patterns are modeled and recognized in nature. During the past decade, research on machine perception helps us to gain a deeper understanding and appreciation for pattern recognition systems in nature. Many techniques exist that are purely numerical and do not have any correspondence in natural systems.

A pattern is an abstract object that can be described by a set of features or measurements. These feature descriptions can be effectively utilized in a classification or regression problem domain. Furthermore, there exist many types of patterns such as visual, temporal, sonic, and logical, etc. Pattern Recognition (PR) is the study of how machines can observe the environment, learn to distinguish patterns of interest, make sound and reasonable decisions about the categories of the patterns Girshick *et al.* (2014a). It is focused on how to make computer programs perform intelligent (human-like) tasks, such as an identifying object in an image. Initially, these tasks were not investigated in-depth as to utilize the machines to achieve this intelligence, as long as it works correctly. Technologies such as filters, boundary detection, and morphological processing have shown to be effective when applied to an image detection algorithms. With a surge in interest in classic PR techniques, researchers in the pattern recognition community spawned the field of optical character recognition while applying the concepts in many other fields. PR was the most innovative and "intelligent" signal processing of the 1970s, the 1980s, and even the early 1990s. Concepts such as decision trees, a heuristic method, and discriminatory quadratic analysis were all introduced during this period. PR slowly shifted from being a topic in electrical engineering to a topic of interest in computer science.

In the early 1990s, many effective ways to create PR algorithms were discovered, particularly replacing probability and statistics researches with machines. This evolution led to the creation of machine learning (ML). The goal of machine learning is to give to computer a collection of data and let the computer make its conclusion with minimal human intervention. It was about generating a probabilistic model to determine the possible outcome of the collected statistics from the data via the machine. Machine learning has origins in Computer Science while pattern recognition

has origins in Engineering yet they are different facets of the same field Girshick *et al.* (2014a).

Machine learning techniques are concerned with the theory and algorithms for categorizing abstract objects (features/measurements made on physical objects) into classes. Typically the classes are assumed to be known in advance, although there are techniques to learn the classes (i.e., clustering). However, Machine learning is a more general problem that encompasses other types of output as well. Other examples are regression, which assigns a real-valued output to each input, e.g. predicting housing prices, normalizing a face image, structural damage estimation tasks, etc; sequence labeling, which categories each member of a sequence of values (for example, in speech tagging, each word in an input sentence is assigned a part of speech); and parsing, which an input sentence is assigned a parse tree, describing the syntactic structure of the sentence. Methods of machine learning are useful in many applications such as information retrieval, data mining, document image analysis and recognition, computational linguistics, forensics, biometrics, and bioinformatics. Statistical classification methods are mostly considered in most of the topics. Methods based on Bayes decision theory and related techniques of parameter estimation and density estimation falls into this category. They are also known as generative methods. Next come discriminative methods such as nearest-neighbor classification, support vector machines. Artificial neural networks, classifier combination, and clustering are other main components of machine learning. A typical machine learning pipeline is shown in Figure 2.1.

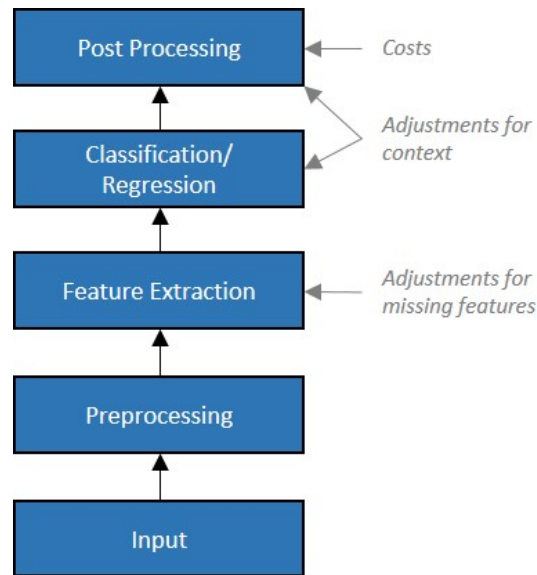


Figure 2.1: Machine is pipeline for regression or classification problems.

The following sections present the background knowledge and related techniques while introducing the problem domains that are mainly focused on this thesis. Furthermore, the databases involved in all our experiments are also described. The summary of the contents is presented at the at the end of this chapter.

2.1 Machine Learning Domains

Machine Learning (ML) has origins in computer science that utilizes algorithms that are funneled by data. In contrast to programs that follow explicit instructions (hard coding), machine learning algorithms use training sets of real-world data to infer models that are more accurate and sophisticated. For example, a hypothetical non-machine learning algorithm for face recognition in images would try to define what unique features that could be seen in a face such as roundness, skin color, eyes, nose line, etc. A machine learning algorithm would not have such coded definitions but would learn by examples. Hence a good algorithm will eventually learn and be able to predict whether or not a new unseen face image belongs to a particular identity. There is different kind of learnings in machine learning.

2.1.1 Supervised Learning

Supervised learning relies on data where the true class of the data is revealed. It learns the task of inferring a function from labeled training data. The training data consist of a set of training examples. In supervised learning, each example is a pair consisting of an input object and the desired output value. A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. Supervised learning mainly divides into classification and regression where the former means to predict the output value using training data and the latter means to group the output into a class.

2.1.2 Unsupervised Learning

Unsupervised learning means that the learning algorithm does not have any labels attached to supervise the learning. The training data is typically unlabelled unlike in supervised learning. In such cases, the algorithm itself cannot reveal structure in the observed data, but it can divide the data into groups. The goal is to discover interesting facts about the measurements such as looking for an informative way to visualize the data or discovering subgroups among the variables or the observations, etc. For example, principal components analysis, a tool used for data visualization or data pre-processing before supervised techniques are applied while clustering, is a broad class of methods for discovering unknown subgroups in data.

2.1.3 Transfer Learning

The intuition behind transfer learning is to use knowledge learned from tasks for which a lot of labeled data is available in settings where there exists a limited amount of labeled data. Creating labeled data is expensive, so optimally leveraging existing datasets is the key. Transfer learning involves using the solution to an existing problem and adapting it to a new related target problem. Torrey and Shavlik (2009); Girshick *et al.* (2014b) provide good intuition about why transfer learning works and how to ensure it can produce good performance.

The primary goal in a traditional machine learning model is to generalize to unseen data based on patterns learned from the training data. With transfer learning, this generalization process is boosted by starting from patterns that have been learned for a similar task. Essentially, instead of starting the learning process from an (often randomly initialized) blank sheet, the learning starts from patterns that have been learned to solve a similar task.

2.2 Linear Techniques

The commonly used linear approaches in machine learning that are relevant to this thesis are detailed in the following sections.

2.2.1 Principal Component Analysis (PCA)

The PCA Turk and Pentland (1991); Belhumeur *et al.* (1997) is an unsupervised linear dimension reduction approach which projects the feature vector into a lower dimensional subspace while maximizing the scatter of the projected data. The intuitive idea here is to preserve a smaller portion of the data while still represents the majority information of the original data. This reduction improves the computation efficiency and removes undesirable noises in the data.

Consider a set of N face images (input) $\{x_1, x_2, \dots, x_N\}$ represented by a d -dimension feature vector for each image, the aim is to define a mapping W to project all input feature vectors into lower m -dimension ($m < d$) output feature vectors $\{y_1, y_2, \dots, y_N\}$. Each output is calculated as:

$$y_i = W^T x_i \text{ for } i = 1, 2, \dots, N \quad (2.1)$$

The mapping W can be calculated by maximizing the determinant of total scatter matrix S_t in the following equations:

$$S_t = \sum_{i=1}^N (x_i - \mu)(x_i - \mu)^T \quad (2.2)$$

$$W^* = \arg \max_W |W^T S_t W| \quad (2.3)$$

where μ is the mean image of all input images. W^* contains m eigenvectors (also known as Eigen-faces) of S_t with the largest eigenvalues. Other eigenvectors with smaller eigenvalues are usually associated with unwanted noise which would decrease the performance of facial recognition rate and therefore should be discarded.

After learning the optimal mapping W^* , all the gallery images are projected into the new subspace. In the testing stage, the query image is also projected into the corresponding subspace and the distance to each gallery image is measured. The Nearest Neighbour (NN) classifier is then applied on this newly defined subspace to determine the identity of the query face image.

2.2.2 Data Whitening

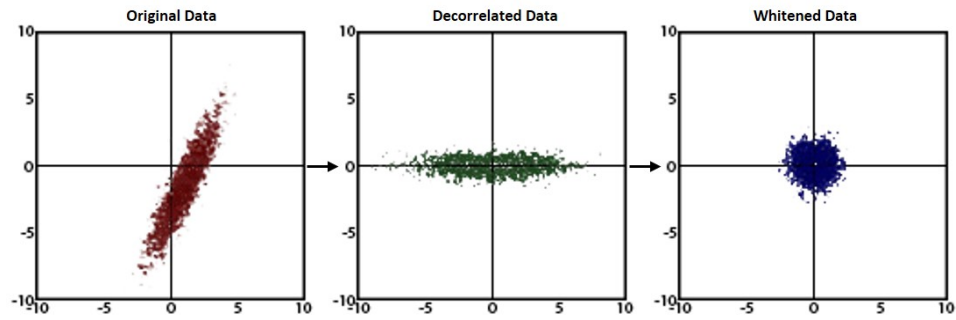


Figure 2.2: A visualization of data whitening process.

The purpose of data whitening is to make the input data less redundant and uncorrelated with each other, and all the features have the same identity variance. The transformation is called "whitening" because the input vector is transferred as a white noise vector. The input features are uncorrelated via the observation of an orthogonal projection matrix U by performing PCA on the original input data:

$$x_{rot} = U^T x^i \quad (2.4)$$

where x^i is the i^{th} sample. To make each input feature have a unit variance, a simple rescale is applied as:

$$x_{whiten,j}^i = \frac{x_{rot,j}^i}{\sqrt{\lambda_j}} \quad (2.5)$$

where λ_j is the eigenvalue corresponding to the j^{th} eigenvector obtained from PCA, $x_{whiten,j}^i$ is the j^{th} component of the whitened data sample. Figure 2.2 shows the schematic process of an example on the data whitening. The first step is to decorrelate the data and the second step is to apply the whitening transformation.

2.2.3 Linear Discriminant Analysis (LDA)

The LDA Belhumeur *et al.* (1997) is another subspace projection technique. It is different from the PCA in the context of the objective of projection. The PCA attempts to maximize the data variance in the new subspace, while the LDA maximize the separability/discrimination of the data classes (e.g. face identity).

The LDA is a supervised approach where it utilizes the class information from the training dataset to train a discriminative classifier. To be more specific, the LDA projects the feature vectors into a new subspace with a requirement that the samples belong to the same class are clustered together and the clusters of samples in different classes are far away to each other. In summary, the LDA minimizes the intra-class (within-class) distance and maximizes the inter-class (between-class) distance.

The subspace projection is similar to 2.1. However, the mapping W is now defined based on the inter-class scatter matrix and intra-class scatter matrix. Assuming there are c classes (unique face identities) in the dataset, the between-class scatter matrix (S_b) and within-class scatter matrix (S_w) are defined as:

$$S_b = \sum_{i=1}^c N_i (\mu_i - \mu)(\mu_i - \mu)^T \quad (2.6)$$

$$S_w = \sum_{i=1}^c \sum_{x_k \in X_i} (x_k - \mu_i)(x_k - \mu_i)^T \quad (2.7)$$

where μ_i and N_i are the mean of the samples and amount of samples respectively in class X_i . $W_{optimal}$ is then computed by maximizing the ratio between determinants of S_b and S_w :

$$W^* = \arg \max_W \frac{|W^T S_b W|}{|W^T S_w W|} \quad (2.8)$$

where the solution can be derived from a generalized eigenvectors and eigenvalues formulation:

$$S_b w_i = \lambda_i S_w w_i, \quad i = 1, 2, \dots, m \quad (m \leq c - 1) \quad (2.9)$$

The computation in 2.7 is only feasible if S_w is a nonsingular (invertible) matrix. Unfortunately, this assumption is most likely not satisfied due to the fact that S_w cannot reach full rank since the number of samples N is usually much smaller than the number of pixels (d dimension). This problem has been solved by incorporating the PCA at the early stage to project the feature vectors into a lower dimensional subspace to ensure that S_w is nonsingular. This approach (also known as Fisherfaces) computes W^* derived from two projections W_{pca} and W_{lda} :

$$(W^*)^T = W_{lda}^T W_{pca}^T \quad (2.10)$$

$$W_{pca}^* = \arg \max_W |W^T S_t W| \quad (2.11)$$

$$W_{lda}^* = \arg \max_W \frac{|W^T W_{pca}^T S_b W_{pca} W|}{|W^T W_{pca}^T S_w W_{pca} W|} \quad (2.12)$$

The Nearest Neighbour (NN) classifier is also applied on the projected feature vectors in order to recognize the class of the query sample.

2.2.4 Sparse Representation Classifier (SRC)

The SRC Wright *et al.* (2009) is one of the state-of-the-art approaches to perform classification. This approach has an assumption that the training samples for each class contain sufficient variations (e.g. facial expressions) spanning the whole sample space for a robust classification task. However, if some prior knowledge of the query sample is known, then fewer variations of the training samples can be tolerated. SRC approximates a given query sample by a linear combination (with coefficients c) of the whole training samples. The intuitive expectation is that coefficient C will be sparse (contain mostly zero-valued elements) with the exception on the training samples of the same class. This sparse representation will immediately expose the class of the query samples since it is easy to notice which training sample is dominant in coefficient c .

Consider a sufficiently large set of N d -dimensional training samples $X = \{x_1, x_2, \dots, x_N\} \in \mathbb{R}^{d \times N}$, the expectation is that a query sample y can be represented as a linear combination of X and sparse coefficient c :

$$y = Xc, \quad c \in \mathbb{R}^N \quad (2.13)$$

In order to ensure that coefficient c is as sparse as possible while satisfying 2.13, one needs to solve the following problem:

$$c_0 = \min \|c\|_0 \text{ s.t. } y = Xc \quad (2.14)$$

which minimizes the amount of non-zero elements in c through ℓ^0 -norm. Unfortunately, 2.14 is considered a NP-hard problem which implies that it is difficult to solve it efficiently. However, it has been discovered Donoho (2006); Candès *et al.* (2006); Candes and Tao (2006); Sharon *et al.* (2009) that the same solution can be obtained with ℓ^1 -norm with the condition that c is sufficiently sparse:

$$c_1 = \min \|c\|_1 \text{ s.t. } y = Xc \quad (2.15)$$

The ideal scenario is that the query sample is "clean" (no unwanted noise). However, the real-life scenario will not always satisfy this condition. In order to improve the robustness to noise, Wright *et al.* (2009) extend 2.13 and 2.15 respectively into:

$$y = Xc + z \quad (2.16)$$

$$c_1 = \min \|c\|_1 \text{ s.t. } \|Xc - y\|_2 \leq \varepsilon \quad (2.17)$$

which incorporates noise vector z and noise level ε in the equations to anticipate noise in the sample. However, because ε is difficult to predict beforehand, one approach to solve this is by employing Lasso Tibshirani (1996) with sparsity regularization parameter λ :

$$\min_{c,z} \|y - Xc + z\|_2^2 + \lambda(\|c\|_1 + \|z\|_1) \quad (2.18)$$

After the coefficient c is computed, recognition can be done by choosing the class i which produces the smallest sample reconstruction residue on its corresponding coefficients:

$$\min_i r_i(y) = \|y - X\delta_i(c)\|_2 \quad (2.19)$$

where $\delta_i : \mathbb{R}^N \rightarrow \mathbb{R}^N$ is a function to choose only the elements in coefficient c corresponding to class i .

2.3 Neural Networks

A standard artificial neural network (ANN) Padil *et al.* (2017) consists of many simple and connected processors called neurons, with each producing a sequence of real-valued activations. Neurons in the input layer get activated through sensors perceiving the environment and hidden neurons get activated through weighted connections from previously active neurons. Learning is about finding weights that make the neural network exhibit the desired behavior, such as a prediction of a certain set of elements Schmidhuber (2015). Depending on the problem and how the neurons are connected, such behavior may require long causal chains of computational stages, where each stage performs non-linear transformation of the aggregate activation of the network. Back Propagation (BP) based on gradient descent method is one of the most popular shallow learning algorithms used for training the neural network. The fundamental problem in utilizing BP based training in deep neural networks Schmidhuber (2015); Hochreiter *et al.* (2001) is the problem of

vanishing or exploding gradient that occurs when back propagating the gradient to the first layers of the network. With standard activation functions, cumulative backpropagated error signals either shrink rapidly or grow out of bounds. In fact, they decay exponentially in the number of layers, or they explode. It is also known as the longtime lag problem. Later, Bengio *et al.* (1994) demonstrate the basins of attraction and their stability under noise from a dynamical systems point of view: either the dynamics are not robust to noise, or the gradients vanish.

Neural Networks which usually consist of 1 or 2 hidden layers were used for supervised prediction and classification in the 1980's. There have been attempts to train multi-layer neural networks with depth for approximately 20 years using backpropagation and other algorithms. However, they always suffered from various degrees of overfitting and vanishing gradient effect: as the error is back propagated through layers, the entangled, non-linear interactions make it very difficult for the lowest layers to adjust its activations according to the error gradient because of the gradient being close to zero.

2.4 Deep Learning

Deep learning is about learning a hierarchy of features or constructing multiple levels of representation for better abstraction. It is motivated by intuition, theoretical arguments from empirical results, circuit theory and recent knowledge of neuroscience. Most modern learning algorithms (naive Bayes, decision trees, SVMs, kernel methods) are "shallow" whereas deep algorithms involve learning useful representations of input which is passed through several non-linearities before being output. Shallow machine learning approaches would involve much duplication of effort to express things that a deep architecture could more abstractly by gracefully reusing computations that were carried out in the previous layers.

The discovery of the possibilities in pre-training make deep learning possible where the lower layers are trained in a greedy (ignoring, the higher layers), problem-agnostic manner. The pre-training looks for regularities in the data to build an efficient, disentangled input representation that just about any higher layer will find more useful than the original input. Hence each layer acts as a regularizer of the data, producing a representation which is mostly invariant to the details of the original input but essential for the task being performed via the network. The next layer can utilize this learned representation to extract more interesting patterns to yield a better abstraction of the original data.

2.4.1 Autoencoder

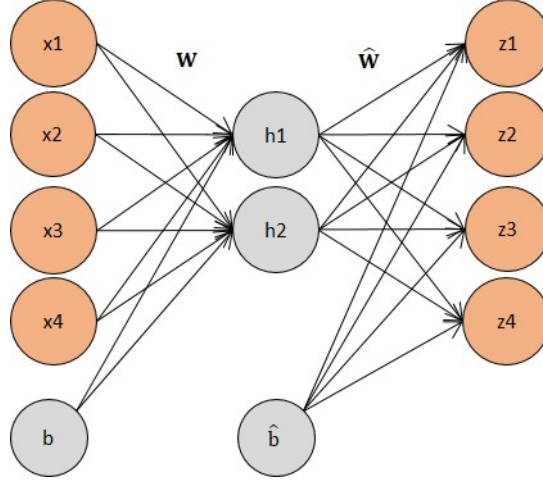


Figure 2.3: Typical shallow autoencoder with encoder and decoder components.

A traditional unsupervised autoencoder DL network Bengio (2009) consists of two core segments: encoder and decoder Vincent *et al.* (2010) - with a single hidden layer as shown in Figure 2.3

Encoder: The deterministic mapping $f(\mathbf{x})$ that transforms a d -dimensional input vector $\mathbf{x} \in \mathbb{R}^d$ into a r -dimensional hidden representation $\mathbf{h} \in \mathbb{R}^r$ is called an encoder. Its typical form is an affine mapping followed by a non-linearity as follows:

$$\mathbf{h} = f(\mathbf{x}) = \Phi(W\mathbf{x} + \mathbf{b}) \quad (2.20)$$

where $W \in \mathbb{R}^{r \times d}$ denotes the affine mapping, $\mathbf{b} \in \mathbb{R}^r$ is the bias and $\Phi(\mathbf{x}) = \text{sigmoid}(\mathbf{x}) = \frac{1}{1+e^{-x}}$ is the activation function of each element which is usually a squashing non-linear function.

Decoder: The mapping $g(\mathbf{h})$ that transforms the hidden representation \mathbf{h} (observed in the above step) back into a reconstructed d -dimensional vector \mathbf{z} in the input space is called a decoder. The typical form of a decoder also has an affine mapping optionally followed by a squashing non-linearity.

$$\mathbf{z} = g(\mathbf{h}) = \Phi(\hat{W}\mathbf{h} + \hat{\mathbf{b}}) \quad (2.21)$$

where $\hat{W} \in \mathbb{R}^{d \times r}$ is the affine mapping, $\hat{\mathbf{b}} \in \mathbb{R}^d$ is the bias and $\Phi(\cdot)$ is the activation function which is described above. \mathbf{z} can be interpreted as the exact reconstruction of \mathbf{x} as well as in

probabilistic terms as the mean of a distribution $p(\mathbf{X} = \mathbf{x} | \mathbf{Z} = \mathbf{z})$ that may generate \mathbf{x} with high probability Vincent *et al.* (2010) where $loss(\mathbf{x}, \mathbf{z}) \propto -\log(p(\mathbf{x} | \mathbf{z}))$. For a real value x , the squared error can be derived as the loss incurred if a Gaussian is chosen as $p(\mathbf{x} | \mathbf{z})$. Hence training an autoencoder to minimize the reconstruction error amounts to maximizing the lower bound on the mutual information between input \mathbf{x} and the representation \mathbf{h} .

2.4.2 Denoising Autoencoder

The denoising autoencoder Vincent *et al.* (2008) is a variation of a typical autoencoder where the input is stochastically corrupted, but the uncorrupted input is still used as the target for the reconstruction. The intuition is to encode the input by preserving the information about the input and try to undo the effect of a corruption process stochastically applied to the input of the autoencoder. It can only be done by capturing the statistical dependencies between the inputs. The stochastic corruption process is to set some of the inputs (as many as half of them) to zero in the random setting. Hence the denoising autoencoder tries to predict the missing values from the non-missing values in the input, for randomly selected subsets of missing patterns. Its loss function is expressed as follows:

$$loss(\mathbf{x}, c(\tilde{\mathbf{x}})) \propto -\log(p(\mathbf{x} | c(\tilde{\mathbf{x}}))) \quad (2.22)$$

where \mathbf{x} is the uncorrupted input, $\tilde{\mathbf{x}}$ is the stochastically corrupted input, and $c(\tilde{\mathbf{x}}) = f(g(\tilde{\mathbf{x}}))$ (Eq. 2.20, Eq. 2.21), is the decoded code obtained from $\tilde{\mathbf{x}}$. Hence the output of the decoder is viewed as the parameter for the distribution over the uncorrupted input. The denoising autoencoder can be shown to be in correspondence to a generative model. Also, another interesting aspect of the denoising auto-encoder is that it naturally lends itself to data with missing values. This is because it is trained with stochastically corrupted inputs in the random setting. In Vincent *et al.* (2008), an extensive series of experimental comparisons over 8 vision tasks were performed with respect to a supervised criterion and shown that stacking denoising autoencoders into a deep architecture fine-tuned yielded generalization performance that was better than stacking typical autoencoders.

2.4.3 Sparse Autoencoder

Informally, think of a neuron as being "active" (or "firing") if its output value is close to 1, or as being "inactive" if its output value is close to 0. The main objective of a sparse autoencoder is to constrain the neurons to be inactive most of the time. This discussion assumes a sigmoid activation

function. If you are using the tanh activation function, the "inactive" state of the neuron would be close to -1 .

Assume that $a_j^2(\mathbf{x})$ denotes the activation of hidden unit j of the 2^{nd} layer in an autoencoder when the network is given a specific input x . Then the average activation of hidden unit j (averaged over the training set) is as below:

$$\hat{\rho}_j = \frac{1}{N} \sum_{i=1}^N [a_j^2(\mathbf{x}^i)] \quad (2.23)$$

We would like to (approximately) enforce the constraint:

$$\hat{\rho}_j = \rho \quad (2.24)$$

where ρ is a sparsity parameter, typically a small value close to zero (i.e $\rho = 0.05$). To satisfy this constraint, the average activation of each hidden neuron j must mostly be near 0. In order to constrain the learning, an extra penalty term is introduced to the optimization objective $J(W, b)$ that penalizes $\hat{\rho}_j$ deviating significantly from ρ . Hence the penalty term is shown below:

$$\sum_{j=1}^{s_2} \rho \log \frac{\rho}{\hat{\rho}_j} + (1 - \rho) \log \frac{1 - \rho}{1 - \hat{\rho}_j} \quad (2.25)$$

where s_2 is the number of neurons in the hidden layer, and the index j is summing over the hidden units in the network. Hence this penalty term is based on the concept of KL divergence and alternatively could be expressed as below:

$$\sum_{j=1}^{s_2} KL(\rho || \hat{\rho}_j) \quad (2.26)$$

where $KL(\rho || \hat{\rho}_j) = \rho \log \frac{\rho}{\hat{\rho}_j} + (1 - \rho) \log \frac{1 - \rho}{1 - \hat{\rho}_j}$ is the Kullback-Leibler (KL) divergence between a Bernoulli random variable with mean ρ and a Bernoulli random variable with mean $\hat{\rho}_j$. KL divergence is a standard function for measuring how different two different distributions are. This penalty function has the property that $KL(\rho || \hat{\rho}_j) = 0$ if $\hat{\rho}_j = \rho$ and otherwise it increases monotonically as $\hat{\rho}_j$ diverges from ρ . KL divergence reaches its minimum of 0 at $\hat{\rho}_j = \rho$, and blows

up (it actually approaches ∞) as $\hat{\rho}_j$ approaches 0 or 1. Thus, minimizing this penalty term has the effect of causing $\hat{\rho}_j$ to be close to ρ . The overall cost function is shown below:

$$J_{sparse}(W, b) = J(W, b) + \beta \sum_{j=1}^{s_2} KL(\rho || \hat{\rho}_j) \quad (2.27)$$

where $J(W, b)$ is as defined previously, and β controls the weight of the sparsity penalty term. The term $\hat{\rho}_j$ (implicitly) depends on W, b also, because it is the average activation of hidden unit j , and the activation of a hidden unit depends on the parameters W, b .

2.4.4 Deep Autoencoder

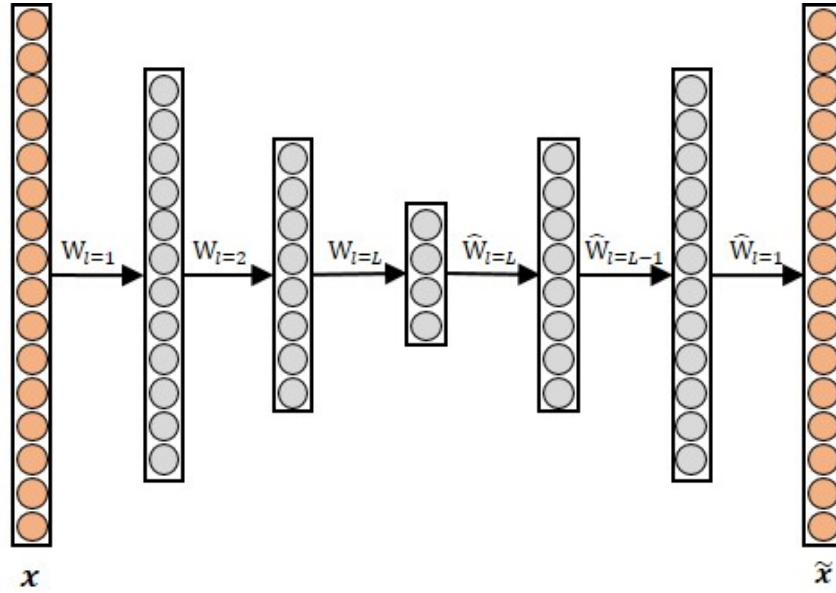


Figure 2.4: Typical deep autoencoder with multiple encoder and decoder components.

Deep Autoencoder is a deep multi-layer neural network that utilizes autoencoders as building blocks Bengio *et al.* (2007a); Ranzato (2007); Vincent *et al.* (2008). Typically a deep autoencoder network is used to perform hierarchical dimensionality change via multiple layers of encoding and decoding. One of the main advantages of utilizing autoencoders in building a deep autoencoder model is the weight initialization. As shown in Figure 2.4 deep autoencoder consists of a series of encoding and decoding layers. Hence the training procedure for encoding layers of a deep autoencoder can be summarized as follows:

- The first layer is trained as an auto-encoder to minimize some form of reconstruction error

of the raw input. In this step, training is purely unsupervised.

- The hidden units' outputs (i.e. the hidden feature representation) of the auto-encoder are then fed as the input to another layer, also trained to be an autoencoder.
- Iterate the second step to initialize the desired number of additional encoding layers.

Training is carried out in an unsupervised fashion and requires no labeled samples. Furthermore, encoding layers are utilized to perform progressive dimension reduction of the input feature \mathbf{x} . The decoding layers facilitate the progressive dimensionality gain as opposed to the gradual dimension reduction via encoding layers. We choose the nodes for each hidden layer as shown in Figure 2.4. The pre-training for layer 1 and layer 2 is performed as described in Section 3.1. Once pre-trained weights are obtained (layer 1, layer 2 encoding weights), these encoding weights ($W_{l=1}^*, W_{l=2}^*$) are used to tie the weights Hinton *et al.* (2006) for the corresponding decoders as shown below:

$$\widehat{W}_{l=1} = (W_{l=1}^*)^T, \widehat{W}_{l=2} = (W_{l=2}^*)^T, \dots, \widehat{W}_{l=L} = (W_{l=L}^*)^T \quad (2.28)$$

where W_l is the weight for encoding layer l and \widehat{W}_l is the corresponding tied weight for decoding layer l . Then the whole network is fine-tuned again to optimize all layers (L) jointly as below

$$\begin{aligned} & \left[W_l^*|_{l=1}^L, b_l^*|_{l=1}^L, \widehat{W}_L^*, \widehat{b}_L^* \right] = \\ & \arg \min_{W_l|_{l=1}^L, b_l|_{l=1}^L, \widehat{W}_L, \widehat{b}_L} \sum_{i=1}^S \sum_{r=1}^{N_i} \|\mathbf{x} - \kappa(\mathbf{x})\|_2^2 \end{aligned} \quad (2.29)$$

where $\kappa(\mathbf{x}) = g_1 \dots g_{L-1}(g_L(f_L(f_{L-1} \dots f_1(\mathbf{c}^r))))$ with $L = 2$ and $W_l|_{l=1}^L$ denotes the encoder's weights while $\widehat{W}_l|_{l=1}^L$ denotes the decoder's weights as shown in Figure 2.4. By jointly optimizing the objective function as shown in Eq. 2.29 towards reconstructing the original feature \mathbf{x} , the decoder weights will be further optimized to perform non-linear dimensionality gain to produce the original feature at the output.

2.4.5 Activation Functions

Activation functions play a key role in deep learning networks to learn non-linear complex functional mappings between the inputs and response variable. They introduce non-linear properties to the network thus converting an input signal of a node to an output signal is the main purpose. The output signal now can be used as an input in the next layer in the stack. Specifically in a deep learning network activation function $a(x)$ is applied to the sum of products of inputs (x) and their corresponding Weights (W) to get the output of that layer and feed it as an input to the next layer.

Another important feature of an activation function is that it should be differentiable. We need it to be this way so as to perform backpropagation optimization strategy while propagating backwards in the network to compute gradients of error (loss) with respect to weights and then accordingly optimize weights using gradient descend or any other optimization technique to reduce the error.

2.4.5.1 Linear Function

A linear function is just a polynomial of one degree. A linear equation is easy to solve but they are limited in their complexity and have less power to learn complex functional mappings from data. A neural network without an activation function would simply be a linear regression model, which has limited power and does not performs good most of the times. Given the node input z , the linear activation function can be mathematically expressed as:

$$lin(z) = z \quad (2.30)$$

2.4.5.2 Piecewise Linear Function

Due to the undefined bounds of the linear activation function shown in Eq. 2.30, the output of a hidden node can reach infinity eventually. Therefore in most applications, piecewise linear activation (variant of linear activation) is utilized to saturate the linear activation function as shown below:

$$satlin(z) = \begin{cases} 0 & \text{if } z < x_{min} \\ mz + b & \text{if } x_{min} \leq z \leq x_{max} \\ 1 & \text{if } z > x_{max} \end{cases} \quad (2.31)$$

where $m = \frac{1}{x_{max} - x_{min}}$ and $b = -mx_{min} = 1 - mx_{max}$ for some x_{min} and x_{max} , which is the "range" for the output of the hidden node. Everything less than this range will be 0, and everything greater than this range will be 1. Anything else is linearly-interpolated between. Hence z in Eq.2.31 is the input to the hidden node as described above.

2.4.5.3 Sigmoid Function

The sigmoid non-linearity has the mathematical form:

$$sigmoid(z) = \frac{1}{1 + e^{-z}} \quad (2.32)$$

and also known as the logistic function. As alluded to in the previous section, it takes a real-valued number and squashes it into range between 0 and 1. In particular, large negative numbers become 0 and large positive numbers become 1. The sigmoid function has seen frequent use since it has a nice interpretation as the firing rate of a neuron: from not firing at all (0) to fully-saturated firing at an assumed maximum frequency (1).

Furthermore, sigmoid outputs are not zero-centered. This is undesirable since neurons in later layers of processing in a deep learning network would be receiving data that is not zero-centered. This has implications on the dynamics during gradient descent, because if the data coming into a neuron is always positive (e.g. $x > 0$ elementwise in $f = w^T x + b$), then the gradient on the weights w will during backpropagation become either all be positive, or all negative (depending on the gradient of the whole expression f). This could introduce undesirable zig-zagging dynamics in the gradient updates for the weights. It may make the gradient updates go too far in different directions ($0 < output < 1$), thus making the optimization harder. However, notice that once these gradients are added up across a batch of data the final update for the weights can have variable signs, somewhat mitigating this issue.

2.4.5.4 Tanh - Hyperbolic Tangent Function

The tanh non-linearity has the mathematical form:

$$\tanh(z) = \frac{1 - e^{-2z}}{1 + e^{-2z}} \quad (2.33)$$

It squashes a real-valued number to the range $[-1, 1]$. Like the sigmoid neuron, its activations saturate, but unlike the sigmoid neuron its output is zero-centered. Therefore, in practice the tanh non-linearity may be preferable over the sigmoid non-linearity in most of the problem domains. A tanh neuron can be also seen as a scaled sigmoid neuron, in particular the following holds:

$$\tanh(z) = 2 \times \text{sigmoid}(2z) - 1 \quad (2.34)$$

2.4.5.5 ReLU - Rectified Linear Unit

The Rectified Linear Unit has become very popular in the last few years. It computes the function:

$$\text{relu}(z) = \max(0, z) \quad (2.35)$$

where it simply thresholds the linear activation function at zero. It is very simple and efficient activation function that can be used to achieve sparsity among the hidden nodes. Even though it lacks some potential of modeling non-linearity it has few pros and cons compared to the sigmoid and tanh activation functions as stated below:

- It was found to greatly accelerate (e.g. a factor of 6 in Krizhevsky *et al.* (2012)) the convergence of stochastic gradient descent compared to the sigmoid/tanh functions. It is argued that this is due to its linear, non-saturating form.
- Compared to tanh/sigmoid neurons that involve expensive operations (exponentials, etc.), the ReLU can be implemented by simply thresholding a matrix of activations at zero.
- Unfortunately, ReLU units can be fragile during training and can "die". For example, a large gradient flowing through a ReLU neuron could cause the weights to update in such a

way that the neuron will never activate on any data-point again. If this happens, then the gradient flowing through the unit will forever be zero from that point on. That is, the ReLU units can irreversibly die during training since they can get knocked off the data manifold. For example, it may be found that as much as 40% of a neural network can be "dead" (i.e. neurons that never activate across the entire training dataset) if the learning rate is set too high. With a proper setting of the learning rate this is less frequently an issue.

- Another limitation of ReLu is that it should only be used within hidden layers of a neural network. The output layers should be assigned the softmax function (generalized logistic function) to compute the probabilities for the classes in a classification problem while using a linear function for a regression problem where the inputs are regressed against the real-valued outputs.

2.5 Optimization Methods

Optimization algorithms are utilized to minimize (or maximize) an Objective Function $J(\theta)$ which is simply a mathematical function dependent on the Model's internal learnable parameters which are used in computing the target values Y from the set of predictors X used in the model, i.e. minimizing the loss by the network's training process and also play a major role in the training process of the deep learning model.

The internal parameters of a model play a very important role in efficiently and effectively training a model and produce accurate results. Therefore various optimization strategies and algorithms are utilized to update and calculate appropriate and optimum values of such model's parameters which influence models learning process and the output of a model. Optimization strategies that are considered in this thesis are discussed in the following sections.

2.5.1 Stochastic Gradient Descent

Stochastic gradient descent (SGD) is utilized as the optimization method for nearly all of the deep learning model for its simplicity and versatile ability. Stochastic gradient descent is an extension of the gradient descent algorithm. Gradient descent, in general, has often been regarded as slow or unreliable. In the past, the application of gradient descent to non-convex optimization problems was regarded as foolhardy or unprincipled. The optimization algorithm may not be guaranteed to arrive at even a local minimum in a reasonable amount of time, but it often finds a very low

value of the cost function quickly enough to be useful. A good generalization of machine learning algorithm comes at the cost of having a large training set, but large training sets are more computationally expensive. The cost function used by a machine learning algorithm often decomposes as a sum over training samples of some per-sample loss function. For example, the negative conditional log-likelihood of the training data can be written as:

$$J(\theta) = (1/N) \sum_{i=1}^N L(\mathbf{x}^i, y^i, \theta) \quad (2.36)$$

where L is the per-sample loss $L(\mathbf{x}^i, y^i, \theta) = -\log p(y|\mathbf{x}; \theta)$. For these additive cost functions, gradient descent requires computing the gradient of $J(\theta)$ (shown below) where the computational cost of the operation is $O(N)$.

$$\nabla_{\theta} J(\theta) = (1/N) \sum_{i=1}^N \nabla_{\theta} L(\mathbf{x}^i, y^i, \theta) \quad (2.37)$$

As the training set size grows to billions of samples, the time to take a single gradient step becomes considerably long. The intuition behind the stochastic gradient descent is that the gradient is an expectation. The expectation may be approximately estimated using a small set of samples which is also known as a mini-batch. Specifically, on each step of the algorithm, a minibatch of samples $B = \{\mathbf{x}^1, \dots, \mathbf{x}^{N'}\}$ is drawn uniformly from the training set. The minibatch size N' is typically chosen to be a relatively small number of samples, ranging from 1 to a few hundred. Crucially, N' is usually held fixed as the training set size N grows. We may fit a training set with billions of samples using updates computed on only a hundred samples. Hence the estimate of the gradient for mini-batch of samples is formed as:

$$g = (1/N') \sum_{i=1}^{N'} \nabla_{\theta} L(\mathbf{x}^i, y^i, \theta) \quad (2.38)$$

utilizing the samples from the minibatch B . The stochastic gradient descent algorithm then follows the estimated gradient downhill:

$$\theta \leftarrow \theta - \lambda g \quad (2.39)$$

where the learning rate is λ .

2.5.2 Gradient Descent with Momentum

Gradient descent with momentum, allows a network to respond not only to the local gradient, but also to recent trends in the error surface. Acting like a lowpass filter, momentum allows the network to ignore small features in the error surface. Without momentum a network can get stuck in a shallow local minimum. With momentum a network can slide through such a minimum.

Gradient descent with momentum depends on two training parameters. The parameter α indicates the learning rate, similar to the simple gradient descent. The parameter β is the momentum constant that defines the amount of momentum. β is set between 0 (no momentum) and values close to 1 (lots of momentum). A momentum constant of 1 results in a network that is completely insensitive to the local gradient and, therefore, does not learn properly.

$$\begin{aligned} V_t &= \beta V_{t-1} + (1 - \beta)g \\ W &= W - \alpha V_t \end{aligned} \tag{2.40}$$

where t is the current step and g is the gradient of the cost function $J(\theta)$ defined above.

2.5.3 Conjugate Gradient Algorithms

Stochastic gradient descent adjusts the weights in the steepest descent direction (negative of the gradient). This is the direction in which the performance function is decreasing most rapidly. Although the function decreases most rapidly along the negative of the gradient, this does not necessarily produce the fastest convergence. In the conjugate gradient algorithms, a search is performed along conjugate directions, which produces generally faster convergence than steepest descent directions. Unlike the first order methods (SGD), conjugate gradient algorithm utilizes the second order information to derive the conjugate directions.

In most of the training algorithms that are based on first order derivative (i.e SGD) the learning rate is used to determine the length of the weight update (step size). In most of the conjugate gradient algorithms, the step size is adjusted at each iteration. A search is made along the conjugate gradient direction to determine the step size, which minimizes the performance function along that line.

Conjugate gradient algorithms Hagan (2007) generally require a line search at each iteration. This line search is computationally expensive since it requires that the network response to all training inputs be computed several times for each search. The scaled conjugate gradient algorithm (SCG),

developed by Moller Møller (1993), was designed to avoid the time-consuming line search. The intuition is to combine the model-trust region approach (used in the Levenberg-Marquardt algorithm Roweis (1996)), with the conjugate gradient approach. SCG may require more iterations to converge than the other conjugate gradient algorithms, but the number of computations in each iteration is significantly reduced because no line search is performed. The storage requirements for the scaled conjugate gradient algorithm are about the same as those of Fletcher-Reeves Fletcher and Reeves (1964).

2.6 Regularization

Regularization refers to the act of modifying a learning algorithm to favor "simpler" prediction rules to avoid overfitting. Few important concepts of regularization that were considered in this thesis are briefly discussed in the following sections.

2.6.1 Bias vs Variance

The goal of machine learning is to learn a function that can correctly predict all data it might hypothetically encounter in the world. This is due to the inaccessibility to all possible data. Therefore by doing well on the training data, it can be expected to make better predictions on such datasets that are unseen in the training stage. Generally, training data sample refers to a sample of the true data. When a parameter is estimated from a sample, the estimate is biased if the expected value of the parameter is different from the true value. The expected value of the parameter is the theoretical average value of all the different parameters that could be acquired from different samples. Example: random sampling (e.g. in a poll) is unbiased. Hence if sampling is repeated over and over, on an average better approximation of the true answer could be expected (even though each individual sample might give a wrong answer).

Regularization adds a bias because it systematically pushes the estimates in a certain direction (weights close to 0). If the true weight for a feature should actually be large, consistent mistakes would take place by underestimating it, so on average the estimate will be wrong (therefore biased). The variance of an estimate refers to how much the estimate will vary from sample to sample. If parameter estimate is consistently the same regardless of what training sample being used, this parameter has low variance.

Bias and variance both contribute to the error of a chosen classifier. Variance is the error due to

randomness in how the training data is selected. Bias is the error due to something systematic, not random. High bias will learn similar functions even if given different training examples while being prone to underfitting. The high variance will add a dependency to the learned function towards the specific data used to train while being prone to overfitting. Some amount of bias is needed to avoid overfitting in this situation. Too much bias is bad, but too much variance is usually worse.

2.6.2 Norm Penalties

The weight penalty is a standard way for regularization, widely used in training other model types. The penalties try to keep the weights small (specifically, weights that are large) or non-existent (zero) unless there are big gradients to counteract it, which makes models more interpretable. An alternative name in literature for weight penalties is "weight decay" since it forces the weights to decay towards zero. This helps with generalization because it will not give large weight to features unless there is sufficient evidence that they are useful. The usefulness of a feature toward improving the loss has to outweigh the cost of having large feature weights. The common form of applying norm penalty is shown below:

$$L(\mathbf{w}; X) + \lambda R(w) \tag{2.41}$$

where $L(\mathbf{w}; X)$ is the loss function and $R(\mathbf{w})$ is called the regularization term or regularizer or penalty and λ is called the regularization strength. A discussion on most commonly used norm penalty terms are shown below:

2.6.2.1 L2 Penalty

This method penalizes the square value of the weight (which explains also the "2" from the name) and tends to drive all the weights to smaller values. It measures the vector's length and also known as the Euclidean norm. When the regularizer $R(\mathbf{w})$ is the squared L2 norm $\|\mathbf{w}\|_2^2$, it is called L2 regularization. This is the most common type of regularization and can be added to many machine learning algorithms. The function $R(\mathbf{w}) = \|\mathbf{w}\|^2$ is convex. Therefore if it is added to a convex loss function, the combined function will still be convex.

$$||\mathbf{w}||_2 = \sqrt{\sum_{j=1}^k (w_j)^2} \quad (2.42)$$

2.6.2.2 L1 Penalty

This method penalizing the absolute value of the weight (v- shape function) tends to drive some weights to exactly zero (introducing sparsity in the model) while allowing some weights to be large. This is another common regularizer which is also known as L1 norm. It is convex but not differentiable when $w_j = 0$ (But 0 is a valid subgradient for gradient descent). Hence the results are often in many weights being exactly 0 (sparse) while L2 norm just makes them small but nonzero.

$$||\mathbf{w}||_1 = \sum_{j=1}^k |w_j| \quad (2.43)$$

2.6.3 Cross validation

In order to choose the optimal value for λ to minimize test-set error, we split the training set into training and validation datasets Bengio (2012); Li and Hao (2014). The validation set is assumed to be representing the test dataset and should not be utilized in training stage. The process is to try different values of λ , learn $(J^*(\theta))^\lambda$ on rest of the data, test $(J^*(\theta))^\lambda$ on the validation set and pick the best λ that minimizes the validation set error. This is also known as the "train-on-all" method. The key factor of this method is the choice of the validation set size. A small validation set will lead to a large error in the estimated loss while a large validation set (a small training set) will lead to poor performance in training (poor $(J^*(\theta))^\lambda$).

2.6.3.1 K-fold Cross Validation

In this method, data is divided into K blocks and each block is trained except the k^{th} block. Test on the k^{th} block, average the results and choose the best λ . A high computation cost is involved in this method since there will be K folds and many choices of model or λ associated with each of the folds. This method is illustrated in Figure 2.5 when $K = 5$.

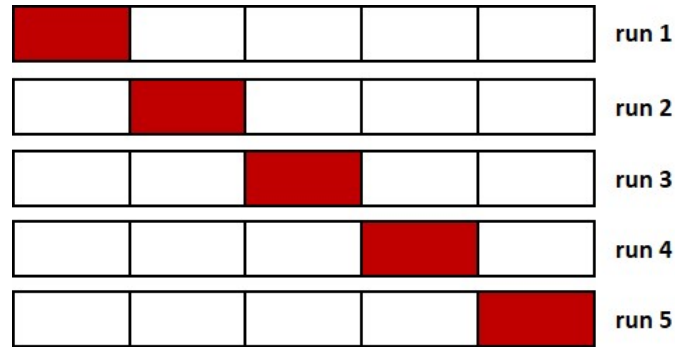


Figure 2.5: Example of k-fold cross validation.

2.6.4 Feature Normalization

The scale of the feature values matters when using regularization. If one feature has values between $[0, 1]$ and another between $[0, 10000]$, the learned weights might be on very different scales. Typically regularizers penalize the weights that are naturally larger. The features that have some important information whose scale is comparatively smaller than the rest may end up learning large weights thus being penalized due to the regularization. This will negatively influence the machine learning task. Therefore feature normalization (or standardization) is essential to perform feature learning efficiently. We perform feature normalization by converting the values to a standard range before being utilized in the deep learning models.

2.7 Face Recognition

Face recognition is an important research problem spanning numerous fields and disciplines due to having numerous practical applications such as bankcard identification, access control, Mug shots searching, security monitoring, surveillance system and etc. Hence it is a fundamental human behaviour that is essential for effective communications and interactions among people.

In the literatures, face recognition problem can be formulated as: given an image (static or video image of a scene), identify or verify one or more persons in the scene by comparing with faces stored in a database. Face recognition in comparison to face verification differ in several aspects. Face verification is concerned with validating a claimed identity based on the image of a face, and either accepting or rejecting the identity claim (one-to-one matching) while the goal of face recognition is to identify a person based on the image of a face. This face image has to be compared with all the registered persons (one-to-many matching). Assume a client (an authorized user of a personal identification system) to be co-operative and makes an identity claim. Automatic

authentication systems that perform verification must operate in near-real time to be acceptable to users. Furthermore, in face recognition experiments, only images of people from the training database are presented to the system, whereas the case of an imposter (most likely a previously unseen person) is of utmost importance for authentication systems.

A formal method of classifying faces was first proposed in Galton (1889). The author proposed collecting facial profiles as curves, finding their norm, and then classifying other profiles by their deviations from the norm. This classification is multi-modal, i.e. resulting in a vector of independent measures that could be compared with other vectors in a database. Progress has advanced to the point that face recognition systems are being demonstrated in real-world settings (Ou *et al.*, 2014). The rapid development of face recognition is due to a combination of factors: active development of algorithms, the availability of a large databases of facial images, and a method for evaluating the performance of face recognition algorithms. The following section presents some comprehensive databases that are utilized in contemporary research field.

2.7.1 Databases

All the proposed deep learning techniques are evaluated on various publicly available image databases. They are generally limited to research purpose only. In this section, a brief description on each database is provided. The details of experiment configuration (e.g. amount of chosen images, how to define training/testing set) will be described separately on experiment section in each chapter.

2.7.1.1 AR Dataset

There are over 3000 colour face images in the AR database Martínez and Benavente (1998) Martínez (1998) captured from 136 people (76 males and 60 females). However, only photographs from 116 people (63 males and 53 females) were obtained properly in all sessions. Each participant was required to attend two sessions (2 weeks apart). Although all the images are only frontal faces, it has 13 variations on facial expressions (neutral, smile, anger, and scream), illumination (lighting from left, right, and both), and occlusions (sunglasses and scarf). Some examples are show in Figure (2.6).

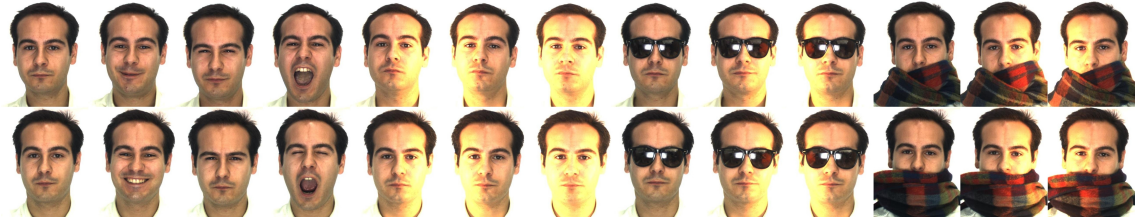


Figure 2.6: 13 image variations on 2 sessions for each participant on AR database.

2.7.1.2 CMU multiPIE

CMU multiPIE Gross *et al.* (2010) Gross (2010) is a massive face database extended from the Pose, Illumination, and Expression (PIE) database Sim *et al.* (2002). It contains more than 750,000 face images from 337 participants with a variation on 6 facial expressions (neutral, smile, surprise, squint, disgust, scream), 15 camera viewpoints (pose), and 19 illumination conditions in 4 sessions over the span of 5 months. Some examples are shown in Figure 2.7. These faces are recorded on high resolution, thus is suitable for face-related applications (e.g. detection or recognition).



Figure 2.7: Examples of all the facial poses and expressions on CMU multiPIE database.

2.7.1.3 CurtinFaces

CurtinFaces Li *et al.* (2013a) Mian (2013) has both 2D colour images and "depth" information which provides basic information of 3D geometric features of the faces. A standard digital camera (Lumix-DMC-FT1) (high resolution 4000x3000 colour images) and a Kinect sensor (Microsoft) (640x480 colour + depth images) were used in the photography session. Various facial expressions, illuminations, poses, and occlusions were captured from 52 participants along with some

combinations (e.g. expression + pose, expression + illumination) leading to 97 images per subject. Some examples are shown in Figure 2.8.



Figure 2.8: range data (depth) and 2D images from CurtinFaces database.

2.7.1.4 CAS-PEAL-R1

CAS-PEAL-R1 Gao *et al.* (2008) Shan (2008) is massive database collected under the sponsor of the Chinese National Hi-Tech Program and ISVISION Tech. Co. Ltd. The variations in this database are enormous, particularly in Pose, Expression, Accessories, and Lighting (PEAL). The whole database consists of 99,594 photographs from 1,040 participants taken from 9 camera angles, 5 facial expressions (closed eyes, frown, open mouth, smile, surprise), 6 accessories (3 hats, 3 glasses), and 15 illumination directions. However, only a part of this dataset is made available to the public which is called CAS-PEAL-R1 containing 30,900 grey-scale photographs with fewer variations. Some examples are shown in Figure 2.9.

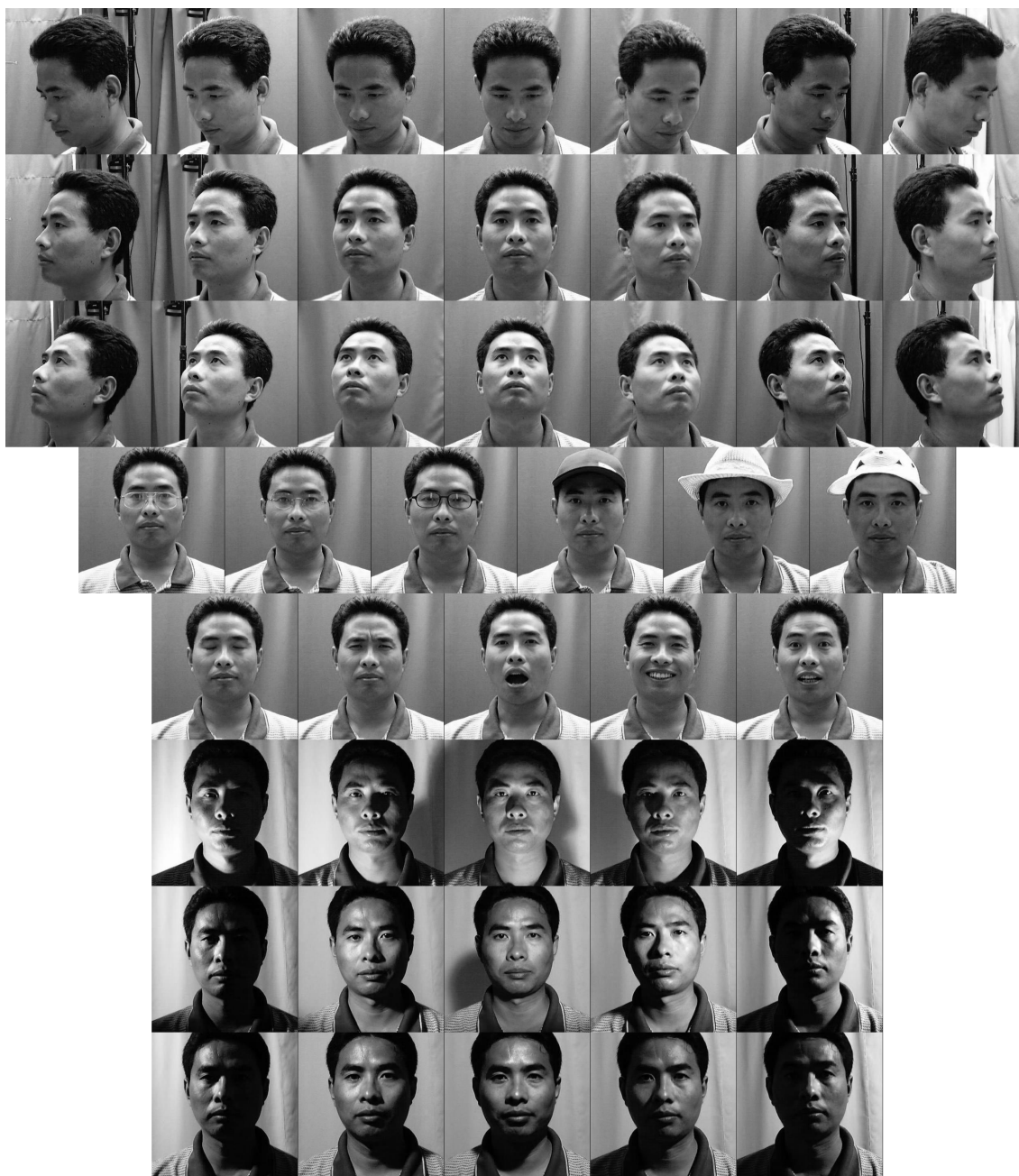


Figure 2.9: Images with variety on (from top) pose, accessories, expression, and illumination from CAS-PEAL-R1.

2.7.1.5 ORL

ORL Samaria and Harter (1994) is a database of faces that was used in the context of a face recognition project carried out in collaboration with the Speech, Vision and Robotics Group of the Cambridge University Engineering Department. It consists of ten different images for each of 40 distinct subjects. For some subjects, the images were taken at different times, varying in lighting, facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement). Some examples are shown in Figure 2.10.



Figure 2.10: Images of ORL database.

2.7.1.6 MNIST

The MNIST (LeCun *et al.*, 1998) is massive database of handwritten digits which composed with a training set of 60,000 examples, and a test set of 10,000 examples. It is a subset of a larger set available from NIST. The digits have been size-normalized and centered in a fixed-size image in MNIST revision. It is a good database on real-world data that could be easily utilized in pattern recognition problems while spending minimal efforts on preprocessing and formatting.

The MNIST database was constructed from NIST's Special Database 3 and Special Database 1 which contain binary images of handwritten digits. NIST originally designated SD-3 as their training set and SD-1 as their test set. However, SD-3 is much cleaner and easier to recognize than SD-1. The reason for this can be found on the fact that SD-3 was collected among Census Bureau employees, while SD-1 was collected among high-school students. Drawing sensible conclusions from learning experiments requires that the result be independent of the choice of training set and

test among the complete set of samples. Therefore MNIST was built by mixing NIST's datasets. The MNIST training set is composed of 30,000 patterns from SD-3 and 30,000 patterns from SD-1. The test set was composed of 5,000 patterns from SD-3 and 5,000 patterns from SD-1. The 60,000 pattern training set contained examples from approximately 250 writers. The sets of writers of the training set and test set are disjoint. Some examples are shown in Figure 2.11.



Figure 2.11: Images of MNIST database with many variations.

2.8 Structure Health Monitoring

Civil infrastructure including bridges and buildings etc., are crucial for a society to well function. They may deteriorate progressively and accumulate damage during their service life due to fatigue, overloading and extreme events, such as strong earthquake and cyclones. Structural Health Monitoring (SHM) provides practical means to assess and predict the structural performance under operational conditions. It is usually referred as the measurement of the critical responses of a structure to track and evaluate the symptoms of operational incidents, anomalies, and deterioration that may affect the serviceability and safety Brownjohn (2007). Numerous efforts have been devoted to develop vibration based structural damage identification methods by using vibration characteristics of structures Li and Hao (2016). These methods are based on the fact that changes in the structural physical parameters, such as stiffness and mass, will alter the structural vibration characteristics as well, i.e. natural frequencies and mode shapes. Structural damage identification based on changes in vibration characteristics of structures can be formulated as a

pattern-recognition problem.

One of the most significant challenges associated with the vibration based methods is that they are susceptible to uncertainties in the damage identification process, such as, finite element modelling errors, noises in the measured vibration data and environmental effect etc. Artificial intelligence techniques, such as Artificial Neural networks (ANN) Padil *et al.* (2017) and Genetic Algorithms (GA) Hao and Xia (2002), are computational approaches based on machine learning to learn and make predictions based on data, and have been applied successfully in diverse applications including SHM in civil engineering. Yun et al. Xu *et al.* (2015) estimated the structural joint damage from modal data via an ANN model. Noise injection learning with a realistic noise level for each input component was found to be effective in better understanding the noise effect in this work. Later, the mode shape differences or the mode shape ratios before and after damage were used as the input to the neural networks to reduce the effect of the modelling errors in the baseline finite element model Ding *et al.* (2017). Measured frequency response functions (FRF) were analyzed by using Principal Component Analysis (PCA) for data reduction, and the compressed FRFs represented by the most significant components were then used as the input to ANN for structural damage detection Yun *et al.* (2001). Ni et al. Lee *et al.* (2005) investigated the construction of appropriate input vectors to neural networks for hierarchical identification of structural damage location and extent from measured modal properties. The neural network is first trained to locate the damage, and then re-trained to evaluate the damage extent with several natural frequencies and modal shapes. Yeung and Smith Zang and Imregun (2001) generated the vibration feature vectors from the response spectra of a bridge under moving traffic as the input to neural networks for examination. It was shown that the sensitivity of the neural networks may be adjusted so that a satisfactory rate of damage detection is achieved even in the presence of noisy signals. Bakhary et al. Ni *et al.* (2002) proposed a statistical approach to account for the effect of uncertainties in developing an ANN model. Li et al. Yeung and Smith (2005) used pattern changes in frequency response functions and ANN to identify structural damage. Later, Bandara et al. Bakhary *et al.* (2007) used PCA to reduce the dimension of the measured FRF data and transformed it as new damage indices. ANN was then employed for the damage localization and quantification. Dackermann et al. Li *et al.* (2011) utilized cepstrum based operational modal analysis and ANN for damage identification of civil engineering structures. The damages in the joints of a multi-storey structure can be identified effectively.

2.8.1 Numerical Studies

The accuracy and efficiency of utilizing the novel deep learning techniques for structural damage identification is another important novel contribution of this thesis. These approaches are first evaluated with simulation data generated from a numerical finite element model. The details of

the experiments (e.g. data generation and pre-processing, etc.) will be described separately on experiment section in each chapter.

2.8.1.1 Numerical Model

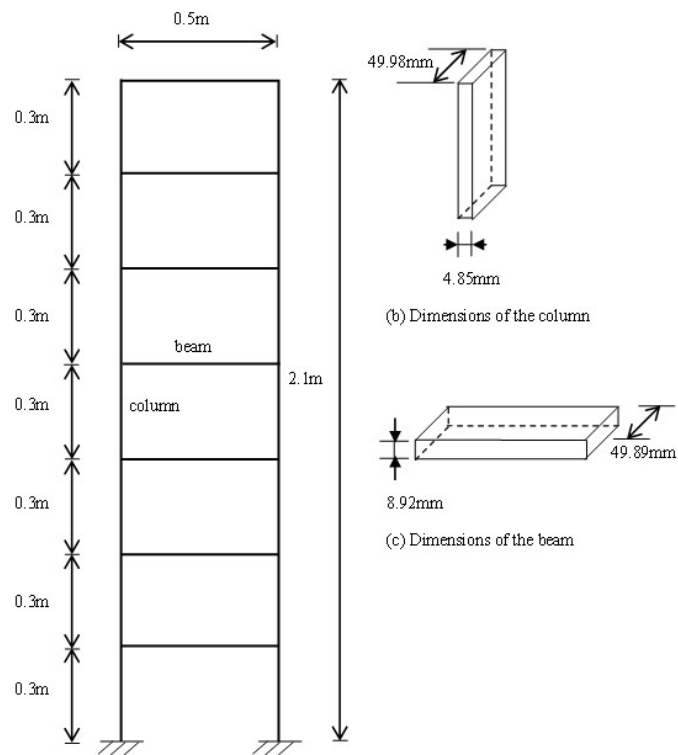
A seven-storey steel plane frame is fabricated in the laboratory and the dimensions of the frame are shown in Figure 2.12(b). The column of the frame has a total height of $2.1m$ with $0.3m$ for each storey. The length of the beam is $0.5m$. The cross-sections of the column and beam elements are measured as $49.98mm \times 4.85mm$ and $49.89mm \times 8.92mm$, respectively. The measured mass densities of the column and beam elements are $7850kg/m^3$ and $7734.2kg/m^3$, respectively. The initial Young's modulus is taken as $210GPa$ for all members. The connections between column and beam elements are continuously welded at the top and bottom of the beam section. Two pairs of mass blocks with approximately $4kg$ weight each, are fixed at the quarter and three-quarter length of the beam in each storey to simulate the mass from the floor of a building structure. The laboratory frame is shown in Figure 2.12(a). The bottoms of the two columns of the frame are welded onto a thick and solid steel plate which is fixed to the ground.

Figure 2.13 shows the finite element model of the whole frame structure. It consists of 65 nodes and 70 planar frame elements. The weights of steel blocks are added at the corresponding nodes of the finite element model as concentrated masses. Each node has three DOFs (two translational displacements x, y and a rotational displacement), and the system has 195 DOFs in total. The translational and rotational restraints at the supports, which are Nodes 1 and 65, are represented initially by a large stiffness of $3 \times 10^9 N/m$ and $3 \times 10^9 N.m/rad$, respectively.

Finite element model updating of the initial finite element model is conducted to minimize the discrepancies between the analytical finite element model and the experimental model in the laboratory. The difference between the frequencies and mode shapes obtained from the analytical finite element model and the experimental measurements is minimized. The measured natural frequencies and mode shapes of the first seven modes are shown in Figure 2.14. Only the mode shape values at the 14 beam-column joints are shown. The First order modal sensitivity-based updating method Friswell and Mottershead (2013) is used. It should be noted that the first 7 measured frequencies and their associated 7×14 mode shape values on the beam-column joints are used in the updating procedure and 70 elastic modulus values and 6 support stiffness values are required to be updated. Table 2.1 shows the first seven measured and analytical frequencies before and after updating, and Table 2.2 shows the Modal Assurance Value (MAC) associated with those seven modes. It can be observed from Table 2.1 and Table 2.2 that a very good updating has been achieved to match the analytical and measured modal information. The detailed model updating process can be found in Li *et al.* (2012a). This updated finite element model is taken as

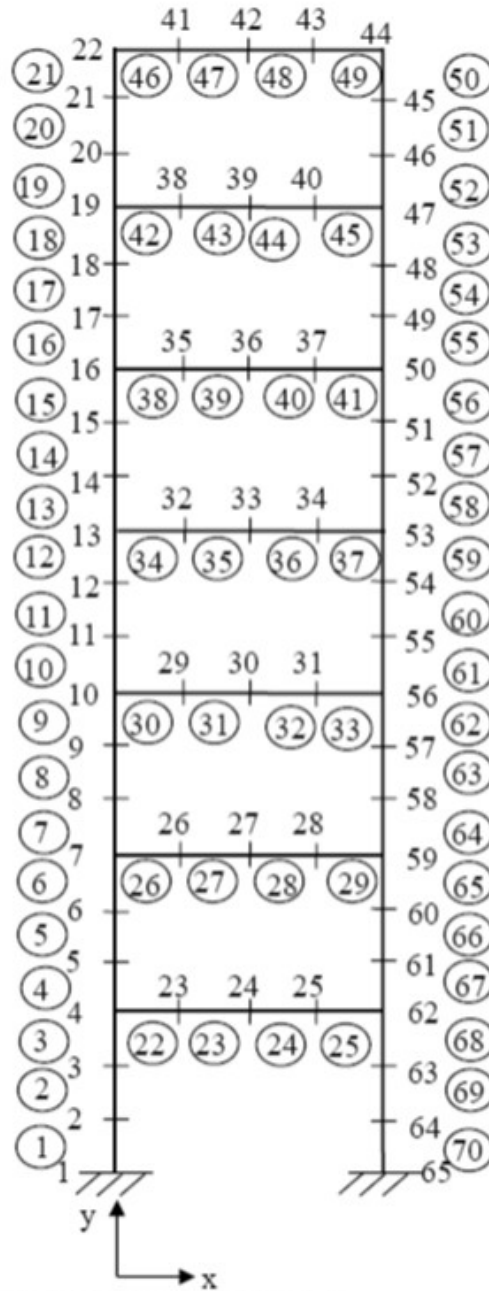


(a) Steel frame model.



(b) Dimensions

Figure 2.12: Laboratory model and dimensions of the steel frame structure.



Note: (1): 1 denotes the node number
 (2): ① denotes the element number in the structure

Figure 2.13: Finite element model of the steel frame structure.

the reference model for generating the training and validation data.

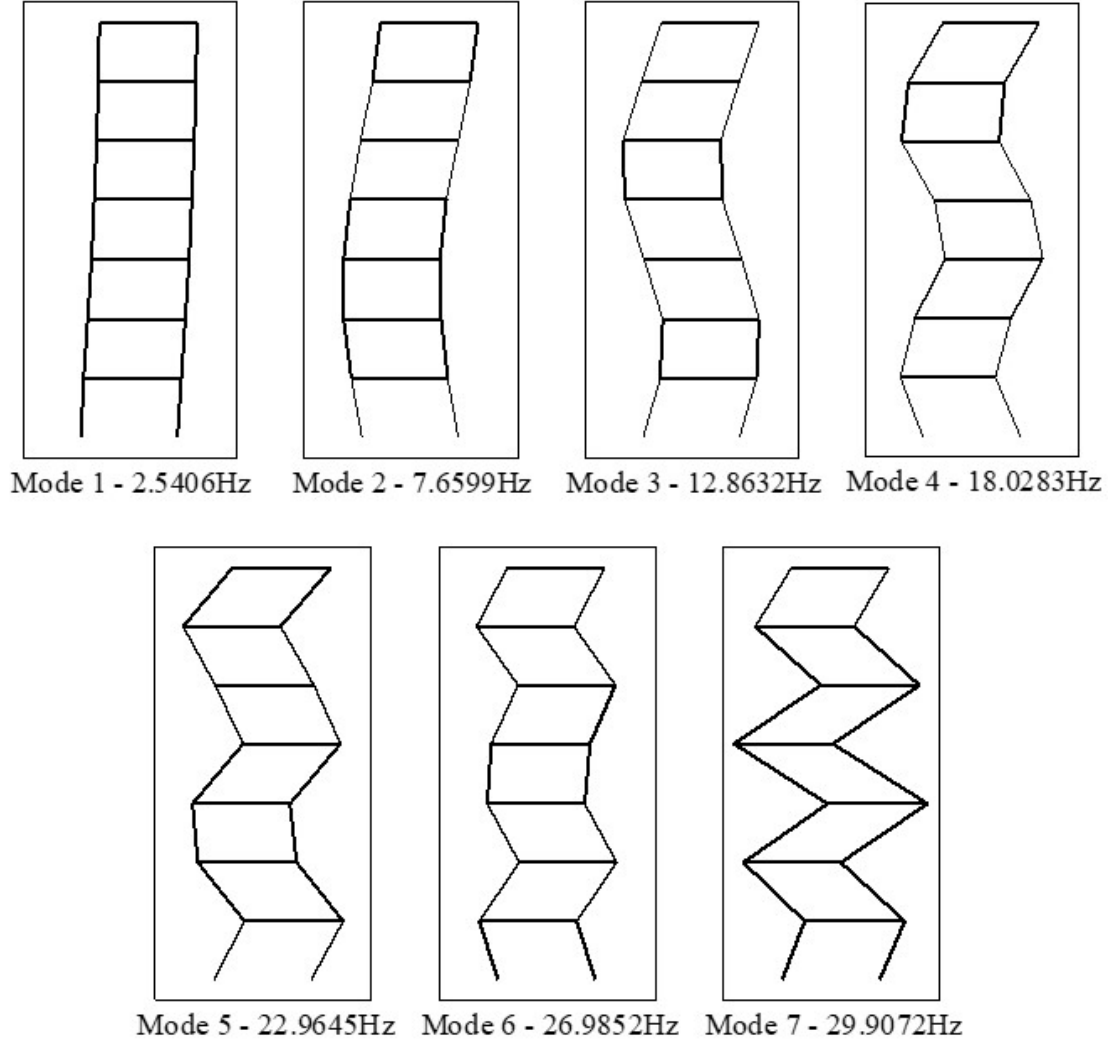


Figure 2.14: The first seven measured frequencies and mode shapes of the frame structure.

2.8.1.2 Data Generation For Numerical Studies

Modal analysis is performed using the baseline model described above with different simulated damage scenarios to generate the input and output data for training the proposed models in this thesis. The first seven frequencies and the corresponding mode shapes of 14 beam-column joints are obtained as the input data to the network. The elemental stiffness parameters are normalized to the range between 0 and 1, where 1 denotes the intact state and 0 denotes the completely damaged state. For example, if the stiffness parameter of a specific element is equal to 0.9, it means that

Table 2.1: Measured and updated frequencies before and after updating.

Mode	Measured	Before Updating		After Updating	
		Analytical(Hz)	Error (%)	Analytical(Hz)	Error (%)
1	2.541	2.520	0.82	2.544	0.13
2	7.660	7.583	1.01	7.655	0.07
3	12.863	12.661	1.57	12.871	0.06
4	18.028	17.626	2.23	18.035	0.03
5	22.965	22.266	3.04	22.984	0.08
6	26.985	26.147	3.11	27.045	0.22
7	29.907	28.796	3.72	30.000	0.31

Table 2.2: MAC Values before and after updating.

Mode	Before Updating	After Updating
1	0.9998	0.9999
2	0.9998	0.9997
3	0.9997	0.9998
4	0.9991	0.9991
5	0.9998	0.9998
6	0.9995	0.9996
7	0.9995	0.9996

10% stiffness reduction is introduced in this specific element. 12,400 datasets are generated based on the baseline model to include both single element and multiple element damage cases. In single element damage cases, the stiffness parameter for each element varies from 1, 0.99, 0.98, , to 0.7 while keeping other elements undamaged. Therefore, 30 datasets are generated for the scenario when a local damage is introduced in a specific element. With 70 elements in the finite element model, 2,100 single element damage cases are obtained. In multiple element damage cases, the stiffness parameters for random two or more elements vary from 1, 0.99, 0.98, , to 0.7 while keeping the other elements undamaged. 10,300 multiple element damage cases are defined. The first seven frequencies and the corresponding mode shapes at those 14 beam-column joints are taken as the input, and the pre-defined elemental stiffness reduction parameters are considered as the labeled output.

2.8.2 Experimental Studies

Experimental verifications of utilizing the proposed deep learning techniques in this thesis for damage identification are next performed using the following laboratory models. The data was acquired with laboratory models via sensors with no simulated data used. The detailed experiment settings will be described separately on experiment section in each chapter.

2.8.2.1 8-Storey Model

An eight-story shear-type steel frame model is fabricated in the laboratory for experimental validations of the proposed approach. Figure 2.15 shows the testing steel frame model in the laboratory. The height and width of the frame structure are 2000mm and 600mm, respectively. Thick steel bars of with dimension of 100mmx25mm are used as the floors of the frame model, and two flat bars of the same cross section with a width of 50 mm and a thickness of 5 mm are used as columns. The beams and columns are welded to form rigid beam-column joints. The bottom of the two columns is welded onto a thick and solid steel plate, which is fixed to a strong floor. The initial elastic modulus of the steel is estimated as 200GPa, and the mass density 7850kg/m³. Dynamic tests are conducted to identify the vibration characteristics of the testing frame model. A modal hammer with a rubber tip is used to apply the excitation on the model. Accelerometers are installed at all the floors to measure horizontal acceleration responses under the hammer impact. The sampling rate is set as 1024Hz, and the cut-off frequency range for the band-pass filter is defined from 1Hz to 100Hz for all tests. An initial shear-type finite element model with 8 lump masses is built based on the dimensions and material properties of the frame. Vibration testing data from the experimental model under the healthy state are used to perform an initial model updating



Figure 2.15: A steel frame model in the laboratory.

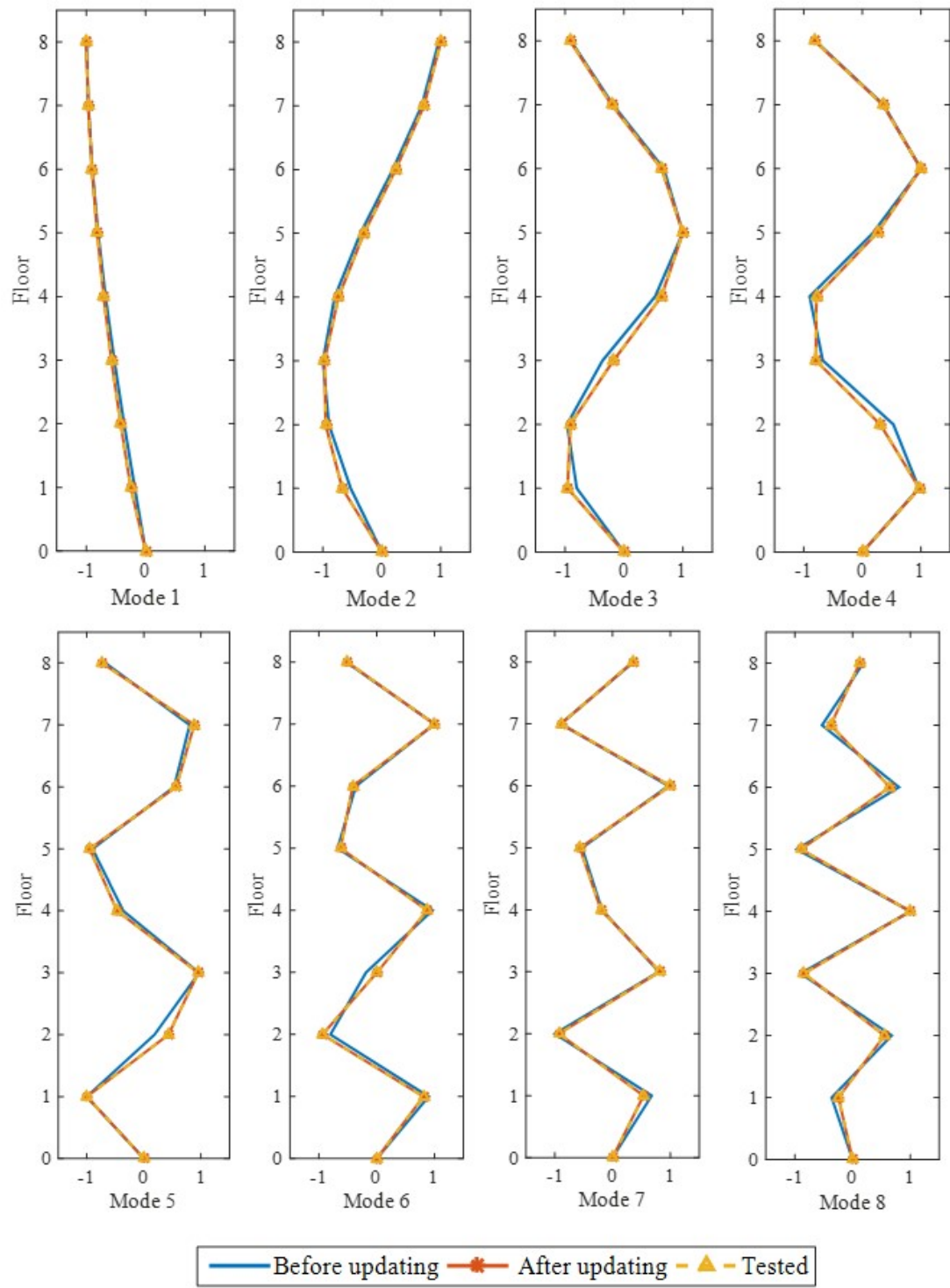


Figure 2.16: Mode shapes before and after updating.

to minimize the differences between the measured and analytical vibration characteristics, i.e. frequencies and mode shapes. The First-order sensitivity based method is employed for the updating Friswell and Mottershead (2013); Lu and Wang (2017). Environmental noise and uncertainties are inevitable in such kind of settings. The detailed experimental test setup and model updating procedure are referred to Ref. Ni *et al.* (2018). The measured and analytical natural frequencies of the experimental model before and after model updating are listed in Table 2.3. The maximum error in the frequencies after updating is only 0.28% at the eighth mode, indicating a very good agreement. The measured and analytical mode shapes of the model are shown in Figure 2.16. The mode shapes after model updating match very well with the measured mode shapes from the vibration tests. This well updated finite element model is achieved to serve as the baseline model in the following studies for generating the training data and validating the performance of the proposed framework in structural damage identification. The following sections will present the data generation process based on the baseline finite element model for network training and validation, the architecture design of the Autoencoder based framework and ANN, and the investigation of using the vibration characteristics from the damaged laboratory model for damage identification with the proposed approach and ANN. Results from ANN and the proposed approach will be compared to demonstrate the performance for a reliable structural damage identification with experimental testing measurements.

Table 2.3: Measured and updated frequencies before and after updating.

Mode	Measured	Before Updating		After Updating	
		Analytical(Hz)	Error (%)	Analytical(Hz)	Error (%)
1	4.645	4.810	3.55	4.636	0.19
2	13.705	14.267	4.10	13.714	0.06
3	22.554	23.238	3.03	22.558	0.02
4	30.695	31.418	2.36	30.776	0.26
5	38.241	38.528	0.75	38.225	0.04
6	44.434	44.325	0.25	44.422	0.03
7	48.826	48.614	0.43	48.712	0.23
8	52.306	51.246	2.03	52.161	0.28

2.8.2.2 T-Beam Model

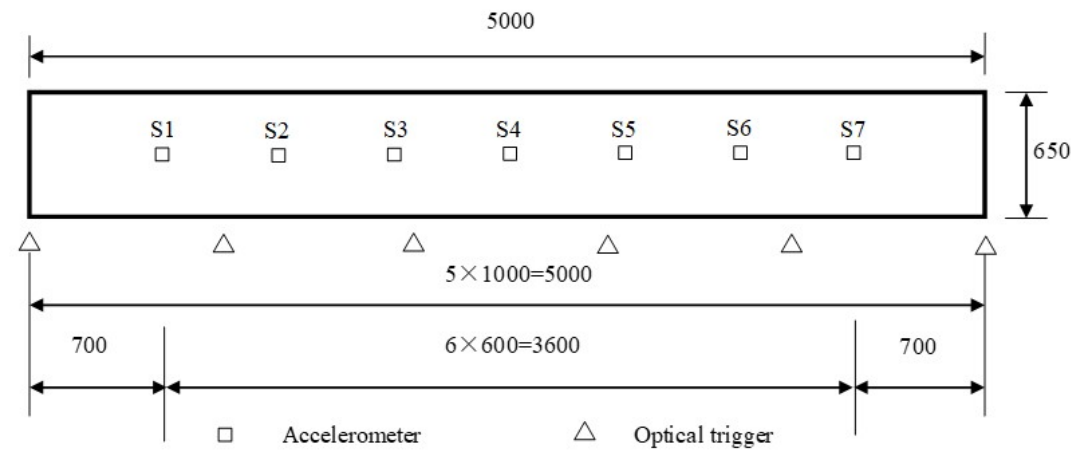
A simply supported T-section prestressed concrete bridge model, as shown in Figure 2.17, is fabricated in the laboratory and tested to verify the effectiveness of applying the proposed framework for structural damage identification. Figure 2.18 shows the dimensions of the bridge model and the locations of placed accelerometers for the modal tests. The bridge is 5m long. The widths of the slab and web are 0.65m and 0.15m, respectively. The height of the beam is 0.415m. The

initial Young's modulus and density are $2.6 \times 104MPa$ and $2707.7kg/m^3$, respectively. Three prestressing tendons with each having $99.8mm^2$ area are included in the bridge web with a total prestress force of $140kN$. The tensile strength of the tendons is $1949N/mm^2$. The cable profile is parabolic and locations of the prestress tendons at the ends and mid-span of the bridge model are shown in Figure 16. The cable duct is grouted after prestressing. Seven accelerometers are placed on the top of the bridge model for recording the dynamic vibration responses in the vertical direction.

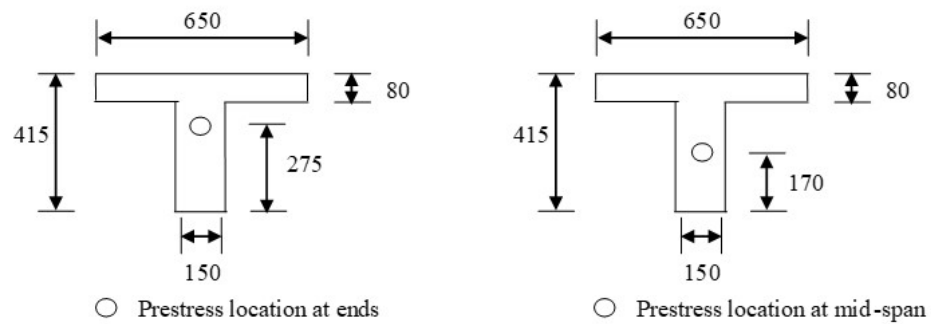


Figure 2.17: The experimental testing model.

The initial model updating is conducted to create a baseline model for generating the training data. Dynamic vibration tests are conducted to identify the vibration characteristics of the model by using a modal hammer to excite the model. The sampling rate is set as $2000Hz$ to well cover the frequency range of excited modes of the bridge model. An initial finite element model of the bridge is built with flat shell elements, as shown in Figure 2.19. The finite element model consists of 90 elements and 114 nodes with 6 DOFs at a node. The model has 684 DOFs in total. The initial model updating is conducted to adjust the built finite element model to serve as the baseline model. The model updating is conducted by minimizing the difference between the first three natural frequencies and mode shapes calculated from the finite element model and measured from the tests. In the initial model updating, the Youngs modulus of slab and web of beam and the support stiffness are selected as parameters to be updated. The dimensions and mass density are measured and not included as the updating parameters. The identified damping ratios



(a) Plan view



(b) Cross-sections

Figure 2.18: Dimensions of the testing model and the sensor placement.

of the beam are included in the initial finite element model. Identified modal information, e.g. natural frequencies and mode shapes of the first three modes, are used to perform the initial model updating. The updated natural frequencies are close to the measured ones, as shown in Figure 2.20. The detailed experimental model, updating procedure and results can be referred to Li *et al.* (2013c). This baseline model will be used in the following studies to generate the training data.

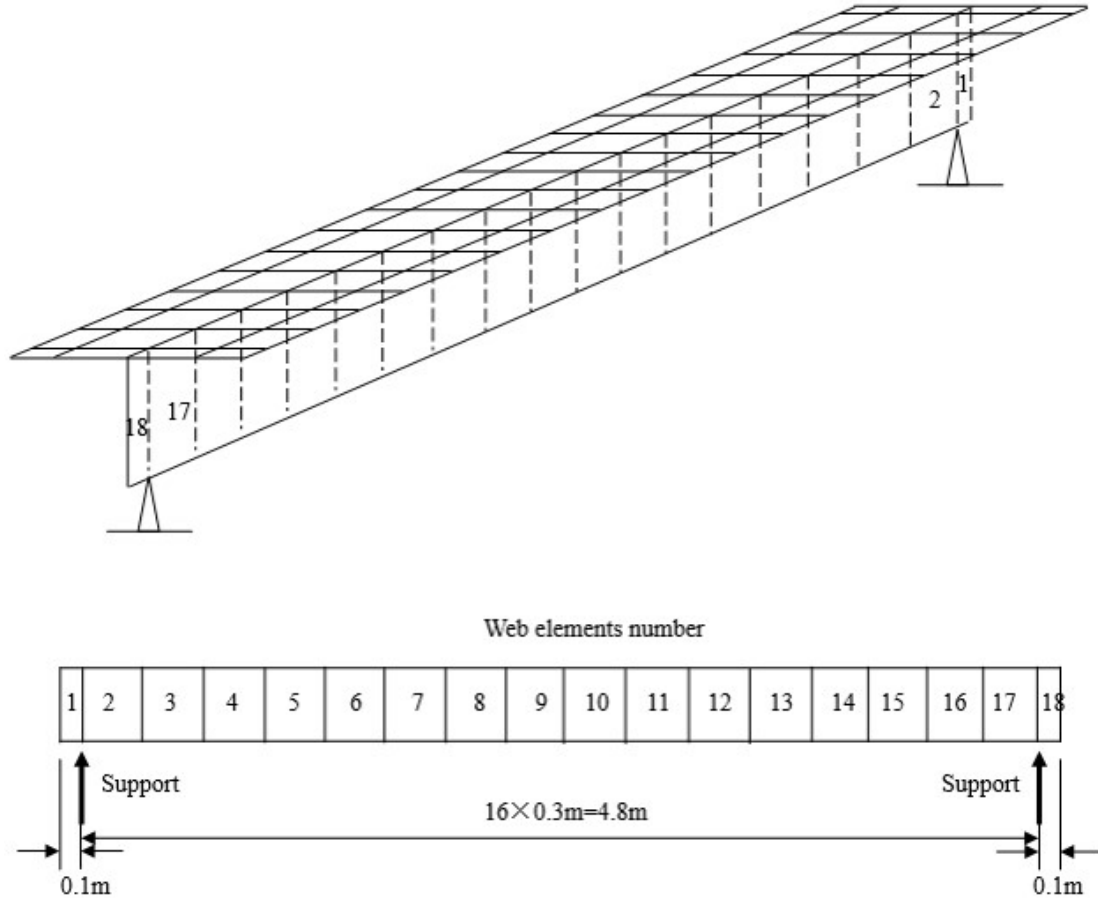


Figure 2.19: Finite element model of the testing bridge.

2.8.2.3 T-Beam: Introduced damage scenario and training data generation

Two-point static loads are applied in the mid-span of the bridge model to introduce the cracks in the model. The static load is continuously increased to 180kN and a number of cracks are observed in the web elements at the middle span, as shown in Figure 2.21. This will be considered as the damaged state. The vibration tests on the damaged bridge are conducted again to identify the first three natural frequencies and the corresponding mode shapes, which will be used later to investigate the performance of the trained sparse autoencoder model defined in this experimental

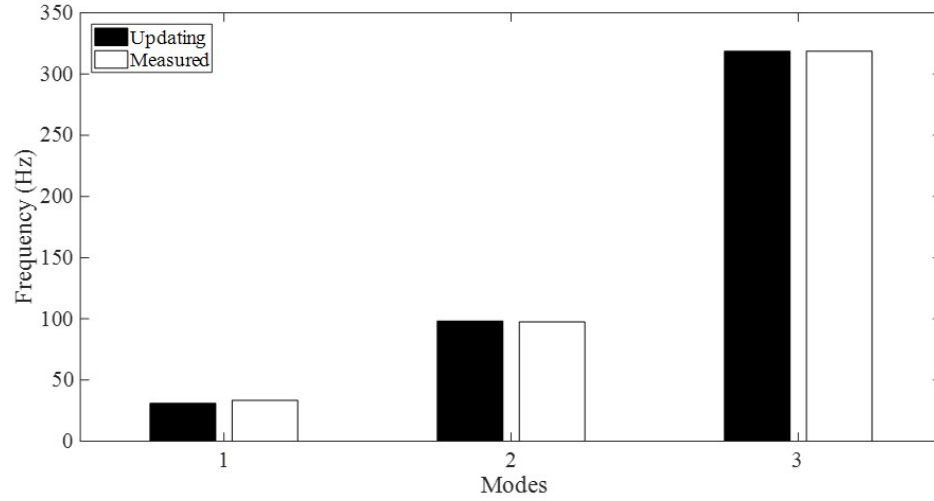


Figure 2.20: Updated frequencies from the finite element model.

study for damage identification with the real measurement data.

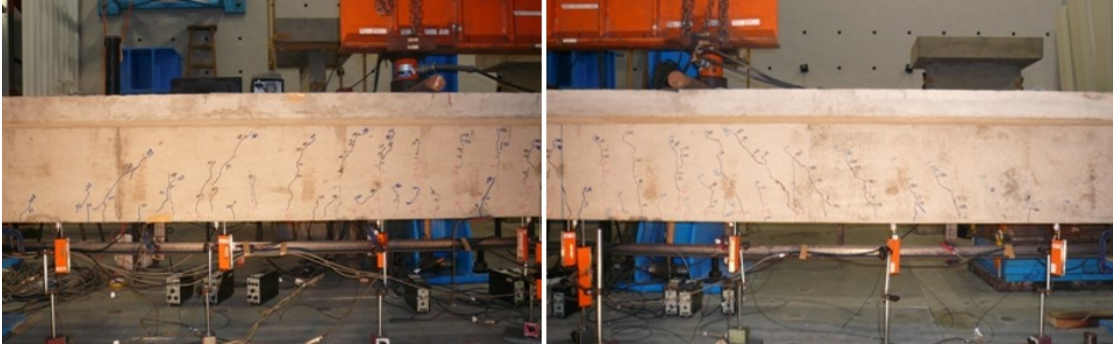


Figure 2.21: Introduced cracks in the web elements of the tested bridge.

When using the baseline finite element model to generate the training data with simulated different damage scenarios, it should be noted that only the web elements No. 2 – 17 as shown in Figure 2.19 are included in the damage identification since the damages are mainly observed in the web elements of the bridge model. Uniformly distributed random damages are simulated in all the 16 web elements from No. 2 – 17 with the damage severity distributed between 0 and 15%. 20,000 data samples are generated. Eigenvalue analysis is then conducted based on the finite element model with the introduced damage to obtain the vibration characteristics, i.e. the first three frequencies and mode shapes at the placed sensor locations. The obtained modal information will be taken as the input to the neural networks, and the simulated damage will be targeted as the labelled output. These datasets will be taken as the training data in this experimental investigation.

2.9 Summary

This chapter provides some basic introductions on preliminary knowledge related to the topic of this thesis. First, an introduction to the evolution of machine learning from pattern recognition is presented in detail. Next the machine learning domains are explained followed by typical linear and non-linear techniques such as LDA, PCA, SRC and Neural Networks etc. In particular, the failures of neural networks are brought to attention while paving the way for the introduction of deep learning. In adherence to the main focus of the thesis, autoencoder model along with its evolution is discussed next. Optimization methods involves a key role in training deep learning models thus posses utmost importance to choose a good optimization method to achieve better results with the such deep models. These aspects are discussed next while presenting a general introduction on regularization strategies to overcome the one of the main problems in deep learning domain which is, overfitting due to the complexities involved. In order to evaluate the proposed model deep learning techniques in this thesis, two application domains (computer vision and civil engineering) are mainly considered. A general introduction on Face Recognition (FR) and Structure Health Monitoring (SHM) which are popular problems in computer vision and civil engineering domains respectively, is presented lastly. Furthermore the popular databases that are utilized to perform experiments in this thesis are presented under the respective categories.

With all the background information given, the following chapters will describe a basic deep learning framework for pattern recognition, the extended basic framework for better performance, a discriminant deep learning framework for labeled data and deep learning system designs for complex problems.

Chapter 3

Basic Framework For Pattern Recognition

Machine learning has been a rapidly expanding discipline in the past decade. It has applications in many areas, including character recognition, remote sensing, target tracking, biomedical image analysis, fingerprint analysis, industrial automation, and robotics. To maximize the scope of application for machine learning, researchers actively searched for a method to automate machine learning, creating the field of deep learning (DL). Today, deep learning, which can be treated as the most significant breakthrough in the field of machine learning, has greatly affected the methodology of related fields like pattern classification and regression in both academia and industry. Deep learning is a relatively promising field, and there remains a lot of research to explore the full potential of deep learning. Recent advancements in unsupervised and transfer learning (Bengio, 2011) methods of Deep Learning Network (DLN) have seen a complete paradigm shift in machine learning. DLNs have a proven pathway of dramatic revolution of current state-of-the-art technologies in almost all walks of AI, ranging from computer vision Le *et al.* (2013), natural language processing, and audio processing. The recent evolution of deep learning technologies Sun *et al.* (2014); Zhu *et al.* (2013) which can even surpass human-level performance in face verification context, not only takes us to a brand new era but also unveils the power of computing.

Deep learning is a set of algorithms in machine learning that attempt to learn in multiple levels, corresponding to different levels of abstraction. It is typically used to extract useful information from data. The levels in these learned statistical models correspond to distinct levels of concepts, where higher-level concepts are defined from lower-level ones, and the same lower level concepts can help to define many higher-level concepts. Alternatively, the main advantage of deep learning is about learning multiple levels of representation and abstraction that can be effectively utilized to make sense of data such as images, sound, text and other types of numerical data.

Unsupervised learning is a crucial component in building successful learning algorithms for deep architectures aimed at approaching better performance on pattern analysis tasks including face recognition (Section 2.7) and structural health monitoring (Section 2.8). It is due to the reasons such as:

- Dealing with future unknown tasks: When there is no knowledge on future learning tasks that model will have to deal with, yet these tasks are assumed to be defined with respect to the existing state of the world (i.e., random variables) that the model can observe now, it would appear very sensible to collect and integrate as much information as possible about this world so as to learn the structure of the data. Typically, most of the information available about this world is not labeled thus could only be used in unsupervised learning context.
- Learning a good high-level representation could make other learning tasks (e.g., supervised or reinforcement learning) much easier. It is due to the effectiveness of the unsupervised feature learning process with deep learning nets that could reassure the salient factors of variation in the input data.
- To avoid gradient vanishing: With layer-wise pre-training much of the learning could be done using information available locally in one layer or sub-layer of the architecture, thus avoiding the hypothesized problems with supervised gradients propagating through long chains with large fan-in artificial neurons.
- Better initialization: Unsupervised learning could put the parameters of a supervised learning machine in a region from which gradient descent (local optimization) would yield good solutions. It has been verified empirically in several settings in Bengio *et al.* (2007a); Larochelle *et al.* (2009); Erhan *et al.* (2009).

3.1 Deep Learning on Non-Linear Problem Domain

Many non-linear characteristics can be observed in various real-world problems. For example, Face recognition (FR) under varying expressions has been a big challenge in automatic FR systems due to the non-linear characteristics that are observed in various facial expressions. Linear methods such as PCA, LDA, SRC (Section 2.7) will fail to capture these non-linearities that exist in the problem domain effectively. Therefore accounting for non-linearity in the learning domain is essential when building a successful learning model. Engineering a deep non-linear model that can address these non-linearities is favorable to overcome the bottlenecks of the linear and shallow methods. Methods such as Zhu *et al.* (2014, 2013) are based on recently-emerged deep learning framework and show a tremendous power of learning highly non-linear transformations at the cost of boosted complexity and added vulnerability to the effects of over-fitting due to their immense power and flexibility in fitting a task. In Zhu *et al.* (2014, 2013) a novel DL model based on Deep Convolutional Networks (DCN) was introduced to convert a random face of an identity to its frontal representation and proven to yield promising results in the pertinent field. One of

the drawbacks of employing convolutional networks is having a very high number of parameters (weights) to be trained thus requiring significant amounts of data to pre-train and then to fine-tune the model accurately. Also, this model consists of eight layers imposing practical challenges in employing such kind of a system without a tremendous computing power.

3.2 Autoencoders as Generic Building Blocks

Hinton and Salakhutdinov (2006) have reported that the neural networks structured as Autoencoders can model nonlinear interactions, and scale well to large datasets. Also, autoencoders can make use of the excessive amounts unlabeled data to learn effective patterns of the data. It was shown that Autoencoders could be composed to create an efficient and flexible dimensionality reduction algorithm Vincent *et al.* (2008). The idea of composing simpler models in layers to form more complex ones has been successful with a variety of basic models, e.g., stacked denoising autoencoders (Hinton and Salakhutdinov, 2006), Boureau *et al.* (2008)'s model, Vincent *et al.* (2010)'s model, etc. These models were frequently employed for unsupervised pre-training. A layer-wise scheme is used for initializing the parameters of a multi-layer perceptron, which is subsequently trained by minimizing an appropriate loss function over real versus model-predicted labels of the data. The original applications mainly focused on face detection, objective recognition, speech recognition and detection, and natural language processing (Schmidhuber, 2015; Arel *et al.*, 2010). Recently it has been developed for fault detection and diagnosis in mechanical engineering (Jia *et al.*, 2016; Gan *et al.*, 2016). Therefore, autoencoder based models can be seen as a decent but feasible DL alternative to the additional complexity posed in DCN.

Inspired by Kan *et al.* (2014), we propose a novel unified deep learning framework (based on autoencoders) which defines the protocols for dealing with complex non-linearities in real-world problems. For example one can perform face recognition across various conditions that describe the intrusive nature of a face and establish the mapping between input modal information and the output structural stiffness parameters, etc. Hence the main objective of such a generic framework is to be able to learn an effective mapping between a set of inputs to a set of outputs and provide optimal solutions for problems of highly non-linear nature. Autoencoders are unsupervised training models (Section 2.4.1) that have a proven track record to learn better feature representations while performing the dimension reduction (Kan *et al.*, 2014). A typical deep autoencoder (Section 2.4.4) can be utilized for effective feature learning through hierarchical non-linear mappings via the multiple hidden layers of the model (Vincent *et al.*, 2010). We utilize such properties of autoencoders to build the proposed deep autoencoder framework that is different from a typical deep autoencoder model (Section 2.4.4) to perform both dimension reduction and relationship learning. A direct application of a typical deep autoencoder (Section 2.4.4) on highly non-linear problem

domain will be intractable due to the high complexities involved in the problem itself (Kan *et al.*, 2014). In contrast, the proposed framework introduces the progressive deep structure with each shallow AE designed to achieve limited but tractable goal, i.e., part of the global non-linearity of the problem. Specifically, as demonstrated in Section 3.4, each shallow AE of our proposed framework is designed to learn efficient feature representations of the input \mathbf{x} and the mapping to the desired output \mathbf{y} which can then be utilized for classification or regression. Such a strategy can enforce the deep network to approximate its eventual goals layer by layer efficiently. Our contributions in this aspect can be listed as follows:

- Problem formulation flexibility with the unified DL framework to yield a better feature space to prominently improve the final objectives of a regression or classification task.
- A novel Multiple-Encoder Single-Decoder feature fusion model and other related techniques to break down the ultimate objective into smaller but tractable goals while generalizing the model to a decent level.
- An elegant and flexible design that can fit into problems with similar non-linear nature.

3.3 Proposed Framework (AutoNet)

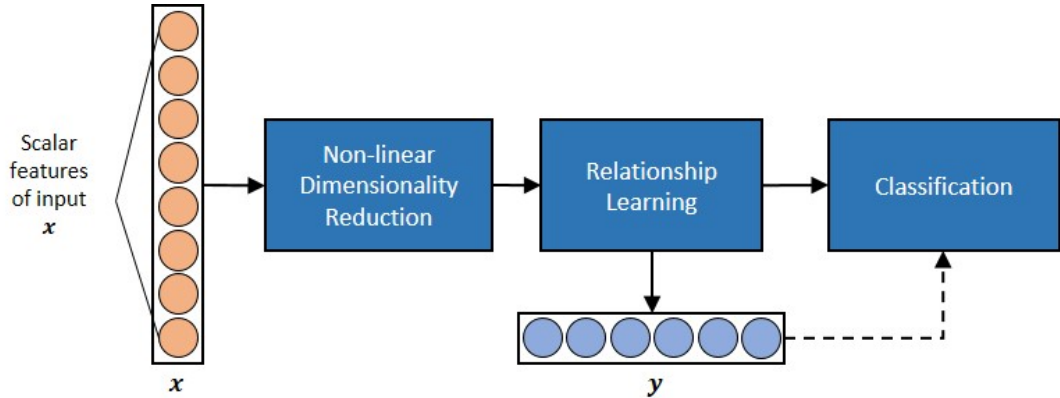


Figure 3.1: Highlevel view of the proposed framework.

In the proposed approach, the relationship of the input \mathbf{x} and the corresponding output \mathbf{y} , is exploited. The input is required to be vectorized before being fed into the proposed framework as shown in Figure 3.1. Since a high dimensional input feature may contain unnecessary information due to redundant data and noise effect, learning a relationship from such a high dimensional feature to the corresponding output will likely to be less accurate than using reduced dimensional features. It is therefore understandable that the problem can be better addressed in two steps. The

first step is to attempt to reduce the dimensionality of the feature preserving the required information, while the second is to learn the relationship between the feature with the reduced dimension and the output. The proposed framework consists of two main components connected sequentially where each component is optimized on a specific objective with relevance to the final goal of a given problem.

The first component of the proposed framework with non-linear activation units supports non-linear dimension reduction. It utilizes the ability of autoencoders to learn the complex low dimensional manifold that can represent the given set of high dimensional features. Since choosing a hidden layer dimension r of the autoencoders with the same or larger than the input dimension d could lead to failure of the autoencoder in learning efficient features than the input (because the AE will simply learn the identity function as the mapping copying the values at the input to the output), an under-complete representation where $r < d$ is necessary to learn useful patterns in the input. Hence \mathbf{h} (hidden feature) can be seen as a compression of \mathbf{x} (input). The quality of the reduced dimensional feature is assessed by observing the amount of information preserved in the reduced dimensional feature by performing the reconstruction of the original feature. This is the objective of the first component of the proposed framework.

Next, the second component of the proposed framework is utilized to learn the relationship between the reduced dimensional feature and the output. A novel single decoder, multi-encoder architecture is developed to perform the learning. The former hidden layers that perform the encoding are pre-trained to perform the non-linear dimensionality reduction while the last hidden layer is trained to learn the relationship between the reduced dimension feature and the output. In this manner, the proposed framework is forced to retain only the required information to establish the relationship between the learned representation and the expected output while encoding the original feature vector. Details of the two components will be discussed in the following subsection.

3.3.1 Dimensionality Reduction

An autoencoder model with a deep neural network architecture is trained for the dimensionality reduction, where the 1st hidden layer is defined to perform the feature fusion of all scalar features that represents a sample while the subsequent 2nd to k^{th} hidden layers further compress the hidden features, as shown in Figure 3.2. One can visualize this model as the encoding architecture of a typical deep autoencoder (Section 2.4.4), but not strictly the generic deep autoencoder model with the decoding structure. The model input is defined as follows:

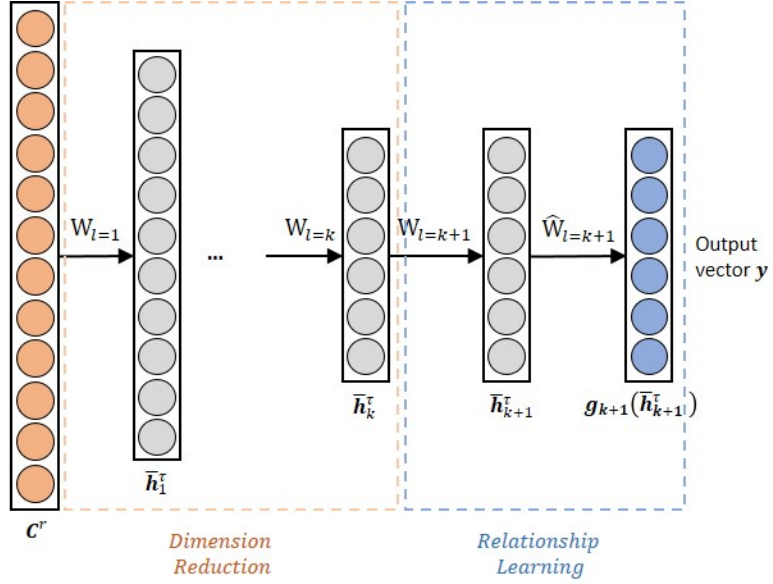


Figure 3.2: Architecture of the proposed framework.

$$\mathbf{c}^r = [q_1^r, \dots, q_n^r] \quad (3.1)$$

\mathbf{c}^r represents the combined high dimensional input vector where q_i^r is the i^{th} ($i = 1 \dots n$) scalar feature of r^{th} sample. Bengio (2009) proposes a layerwise pre-training scheme to train each layer individually with an autoencoder before fine-tuning all the layers together. As discussed in Erhan *et al.* (2009), unsupervised pre-training can be considered as a special form of regularizer (and prior). It amounts to a constraint on the region in parameter space where a solution is allowed. Erhan *et al.* (2009) shows that random parameter initialization for the lower layers (closer to the input layer) will lead to perform poorly on both training and test sets. These experiments show that the effect of unsupervised pre-training is most marked for the lower layers of a deep architecture. We follow a similar layer-wise pre-training scheme for all the layers of this dimensionality reduction component with the following cost function:

$$J_{cost}^{l=p}(W_{l=p}^*, b_{l=p}^*, \widehat{W}_{l=p}^*, \widehat{b}_{l=p}^*) = \arg \min_{W, b, \widehat{W}, \widehat{b}} \sum_{r=1}^N \left\| h_{p-1}^r - g_p(f_p(h_{p-1}^r)) \right\|_2^2 \quad (3.2)$$

where $p = 1 \dots k$ with k being the number of layers in the dimensionality reduction component, N is the number of data samples involved in the training, and g_p and f_p are the decoder and encoder functions of the p^{th} layer, respectively. h_{p-1}^r is the lower dimensional representation that is established in the $(p-1)^{th}$ layer for the r^{th} sample where $h_0^r = c^r$. Encoder function f_p is set to be $\tanh(\cdot)$ (hyperbolic tangent) since the value 0 is contained in its activation region thus supports a sparse representation of the input when the activation of a hidden unit becomes 0. Decoder function g_p is set to be $\text{purelin}(\cdot)$ since it needs to reconstruct the real values of the input. The factor $1/2$ in Eq.3.2 is used to eliminate 2 when taking the gradient of the mean square error, so to have a clear derivative of the cost function. Having multiplication factor on the cost will not change the optimal solution reached via the optimizing method. The compressed representation features learned in the k^{th} layer h_k^r is then fed to a non-linear relationship learning component next.

3.3.2 Relationship Learning

The relationship learning component, as shown in Figure 3.2, is defined to perform the regression task utilizing the low dimensional feature learned at the k^{th} layer, which is a better feature representation than the original input to predict the output feature vector. Note that the input and the output feature vectors may exist in a different feature space. A simple autoencoder model with non-linear activation function is utilized to perform this task. It is essential that a non-linear activation function is utilized to capture the non-linearities that exist in the mapping. Hence a simple linear activation function would underperform in learning highly non-linear mapping. The cost function for this model is defined as:

$$J_{cost}^{k+1}(W_{k+1}^*, b_{k+1}^*, \widehat{W}_{k+1}^*, \widehat{b}_{k+1}^*) = \arg \min_{W, b, \widehat{W}, \widehat{b}} \sum_{r=1}^N \|o^r - g_{k+1}(f_{k+1}(h_k^r))\|_2^2 \quad (3.3)$$

where g_{k+1} and f_{k+1} are respectively the decoder and the encoder functions of the $(k+1)^{th}$ layer (Section 2.4.5), h_k^r is the low dimensional representation obtained at the k^{th} layer (also the last layer) of the dimensionality reduction component for the r^{th} sample, and o^r is the labeled output vector, the corresponding output feature vector for a given input feature vector that define the r^{th} sample. Once the relationship learning is completed, both the hidden feature (h_{k+1}^r) and/or the predicted output could be utilized for classification in classification context. h_{k+1}^r consists of the mapping information from the low dimensional representation h_k^r to the output. Furthermore, both the encoding ($W_{l=k+1}$) and decoding ($\widehat{W}_{l=k+1}$) weights of the relationship learning component

are utilized in the final network to perform the training as described in the next section.

3.3.3 Fine-Tuning

Once the optimal mapping weight coefficients and bias parameters of all the hidden layers are obtained with the pre-training scheme, both the components are fine-tuned to optimize all the layers as a whole with the following cost function:

$$J_{cost}(W_l^*, b_l^*) = \arg \min_{W, b} \sum_{r=1}^N \|o^r - p(c^r)\|_2^2 \quad (3.4)$$

where $p(\mathbf{c}^r) = g_{k+1}(f_k(f_{k-1}(\dots(f_1(\mathbf{c}^r))))$ is the predicted output vector through the activations of all the layers in both the dimensionality reduction and relationship learning components. It is essential to perform layer-wise pre-training followed by fine-tuning to improve the training efficiency and achieve better accuracy from the proposed framework.

3.4 Applications

In order to assess the applicability of the proposed generic framework, we conduct experiments in two different application domains, One is in computer vision and another is in civil engineering. Computer vision involves visual signals (images) that are sparse in representation while civil engineering data typically represent set of numerical values of measurements. By choosing the two vastly different application domains, the wide applicability and easy adaptivity of the proposed framework are demonstrated.

3.4.1 Computer Vision Application - Face Recognition

Face recognition (FR) under varying expressions has been a big challenge in automatic FR systems due to the non-linear characteristics that are observed in various facial expressions. Despite its importance, very little attention is given to 2D image based approaches compared to 3D model-based approaches Drira *et al.* (2013); Wang *et al.* (2014); Li *et al.* (2013b) in the recent history. The facial deformations that appear in some expressions, such as closed eyes, teeth, etc., will

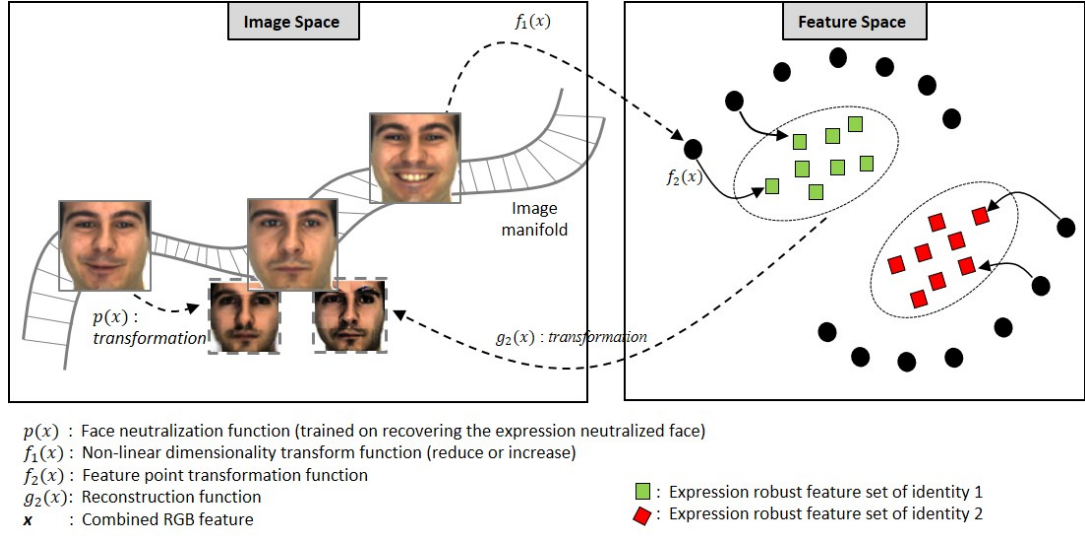


Figure 3.3: Transformation functions that are learned by the proposed AutoNet framework via its hidden layers and the formation of the low dimensional feature space.

introduce additional noise, thus degrade the performance of face recognition systems. The existing approaches can be divided into linear and non-linear methods.

Commonly used linear approaches include subspace model analysis (PCA, FLD) (Tsai and Jan, 2005), and their extensions such as Enhanced Fisher Linear Discriminant (EFLD) (An and Ruan, 2006) and Exponential Discriminant Analysis (EDA) (Zhang *et al.*, 2010). In these methods, a linear subspace is learned to extract facial features followed by a classifier. The techniques such as Global/Local Linear Regression (Xiujuan Chai *et al.*, 2007), Linear Regression Classification (LRC) (Naseem *et al.*, 2010) etc. model the FR as a linear regression problem. A comprehensive review of all of the above methods was rigorously carried out by evaluating their strengths and weaknesses in performing robust face recognition under varying expressions in Kumar *et al.* (2014).

Recently, a decent attempt that outperforms all linear methods were proposed in Wright *et al.* (2009) based on sparse representation coding (SRC) that exploits the tremendous potential of Compressive Sensing (CS) theory for problems in pattern recognition domain. In the SRC framework, it is assumed that the whole set of training samples from a dictionary (each image is a base atom) can approximate a test image of a given class by discriminatively finding sparse coefficients in which they form a linear combination of the atoms in the dictionary. While the SRC-based methods depict the power of constraining the sparsity in FR problems via L1 minimization, they have some disadvantages due to the fundamental design of the SRC framework. Firstly, for accurate recognition, a sufficiently large training image set for each subject is needed to construct a good over-complete dictionary. In practice, it may not be possible to acquire a large set of images

per identity. Secondly, if the model is to be applied to thousands of real-world identities, the size of the dictionary would be really large and thereby pose practical challenges in processing speed and performance. Furthermore, the SRC framework also suffers from the linear nature due to its fundamental assumption of having linear combinations of dictionary atoms to approximate a given test image.

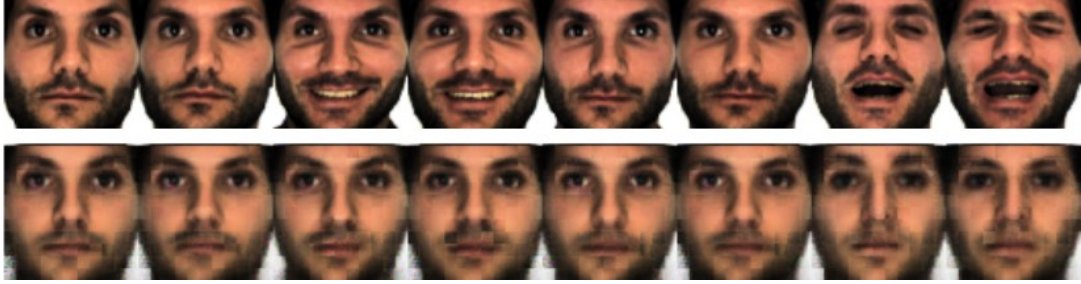


Figure 3.4: Recovered neutral expressions faces against the noisy version of it. The first row of faces with expressions are neutralized and shown in the second row.

All of the above methods are well understood for their fundamental linear nature and in a sense have reached their maturity. Researchers are now looking for non-linear models to address problems that consist of non-linear characteristics. To overcome the bottlenecks of models of linear nature, some approaches such as Zhu *et al.* (2014, 2013) were suggested to incorporate non-linearity in the learning domain of the problem. These methods are based on recently-emerged Deep Learning (DL) framework and show a tremendous power of learning highly non-linear transformations at the cost of boosted complexity and added vulnerability to the effects of over-fitting due to their immense power and flexibility in fitting a task. To overcome the additional complexities posed in complex deep learning nets such as DCN, a decent but feasible DL approach (based on auto-encoders) focused only on pose variation problem was proposed in Kan *et al.* (2014). They argue that the pose variations change non-linearly and smoothly along the manifold thus a stacked progressive auto-encoder model is designed where shallow progressive auto encoders are used to map a face image at a larger pose¹ to a virtual view at smaller poses² while keeping those images at smaller poses unchanged. This method works well for pose variation problems. However, since an expression cannot be learned progressively or subdivided into different stages of posing (non-sequenced face images), it is not possible to follow this approach directly for expression-robust FR. To the best of our knowledge, no comparative research work has been done on the autoencoder (AE) framework to determine suitable methods for expression invariant face recognition.

Our proposed framework allows effective feature learning through hierarchical non-linear mappings via the framework's components. The simple design and ease of training (back-propagation of error gradient) of the proposed framework reduce the essential complexity compared to DCN.

¹Poses that are larger than 30 degrees rotated from the frontal pose.

²Poses that are less than or equal to 15 degrees rotated from the frontal pose.

3.4.1.1 AutoNet For Face Recognition

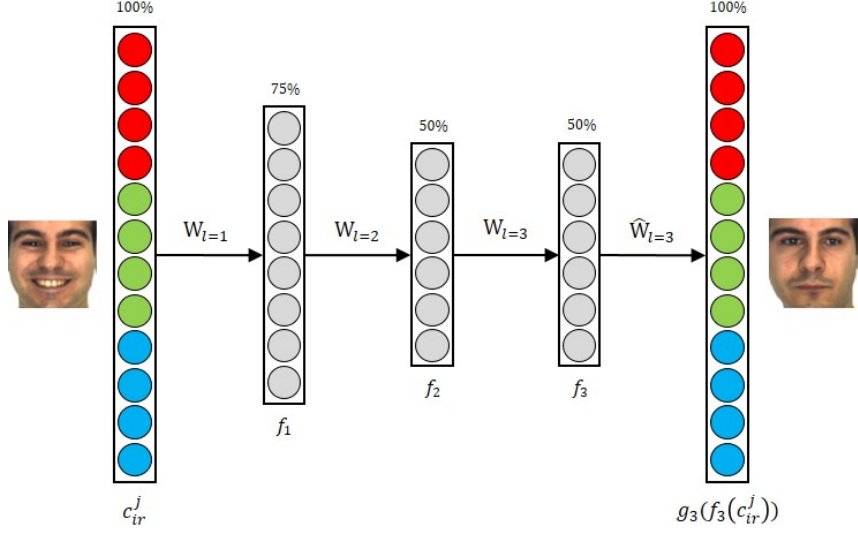


Figure 3.5: The proposed framework where $f_1 \in \mathbb{R}^{50}$ denotes low dimensional noisy feature learnt at layer 1, while $f_2 \in \mathbb{R}^{50}$ denotes the noiseless feature learnt at layer 2 in the observed low dimensional space. We halves the image space by 50% to constraint the framework to learn an effective low dimensional feature.

In our approach, we try to exploit the fact that facial expressions generally change smoothly along the face manifold. Hence recovering the neutral expression from a simple expression is more accurate than recovering it from an extreme one. We consider a face with an expression as a neutral face exposed to noise. AutoNet is trained to de-noise this face noise while learning a better low-dimensional feature space to perform the recognition as shown in Figure 3.5. In here, AutoNet is utilized on visual data to de-noise the face noise as described above. An illustration of such de-noised faces are shown in Figure 3.4. Moreover, the dimension reduction component of the proposed framework can be utilized to perform non-linear dimension reduction while relationship learning component can be utilized on de-noise the noisy representation learned from the former component. We ensure that de-noising is supported by a strong supervisory signal which is the neutral face. Hence the under-complete representation that is learned in the de-noising layer will maximize the mutual information between the neutral face and an expression face (Figure 3.3). In this way, a typical auto-encoder learned representation \mathbf{h} , which is affected by noise, can be de-noised. During this process, it will result in useful features at the right-most hidden layer in Figure 3.5 for a better representation that is invariant under varying noise. A good representation is one that can be obtained robustly from a corrupted input and that will be useful for recovering the corresponding clean input (Vincent *et al.*, 2010). In a nutshell, as mentioned in Pathirage *et al.* (2016):

- The proposed framework supported by the deep architecture subdivides the global non-

linear transformation into series of sub-objectives where each hidden layer of AutoNet is trained on each sub-objective to support the face neutralization process.

- The proposed framework learns better weights for each colour component at the pixel level in observing low dimensional expression-robust feature space.
- The proposed framework adds invariability to expression-induced spatial deformities undergone by a face at the patch level due to patch-based training and it can promote further division of the global learning objective into limited but tractable learning goals.

We choose two hidden layers to perform dimensionality reduction of the input (face with the expression) while utilizing one layer to perform the mapping between the reduced dimensional feature and the output which is, in this case, the neutral expression face. The layer-wise pre-training procedure is performed as shown in the Figure 3.6. Once the optimal parameters are obtained in the pre-training stage, both the dimensionality reduction and relationship learning components (whole network) are fine-tuned (Figure 3.5) together to optimize all layers jointly as described in Section 3.3.3. Both pre-training and fine-tuning are carried out with the full batch gradient descent optimization algorithm for simplicity.

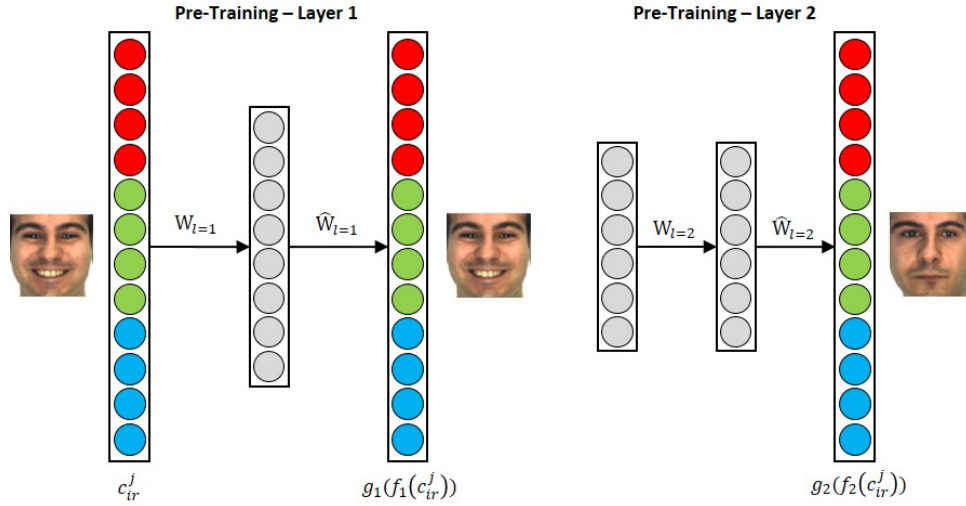


Figure 3.6: Progressive pre-training of respected layers to achieve better initial weights prior to the training phase. Left figure denotes the non-linear dimension reduction layer pre-training while de-noising layer pre-training is shown in the right figure.

3.4.1.2 Experiments

In this section, the proposed AutoNet framework is assessed against the state-of-the-art methods on two publicly available colour image databases, AR (Martínez and Benavente, 1998) and Curtin (Li *et al.*, 2013a), where each of the setups are based on frontal faces with different facial expressions in uniform illumination. All images were cropped, aligned and resized into the resolution of 66x66. The main objective here is to find the identity of a person irrespective of the facial expressions that a face can show. The evaluation of the AutoNet was done through three different experiments, each of them consisting of six (6) test cases to evaluate the AutoNet's invariability on six (6) different expressions as shown in Figure 3.7. These expressions include: opened and closed mouth (smiling), closed eyes and etc. A neutral face is included in these expressions to evaluate the AutoNet's performance over typical FR scenarios as well. Furthermore, the performance graphs compare the results obtained with:

- SRC (Wright *et al.*, 2009), LDA, PCA
- Deep learned feature (at the expression-robust hidden layer) on SRC(AutoNet-SRC), LDA(AutoNet-LDA), PCA(AutoNet-PCA)
- Reconstructed neutral face (at the output layer) on SRC(Reconst), LDA(Reconst), PCA(Reconst), Nearest Neighbor with V channel of the HSV color space (HSV-V)

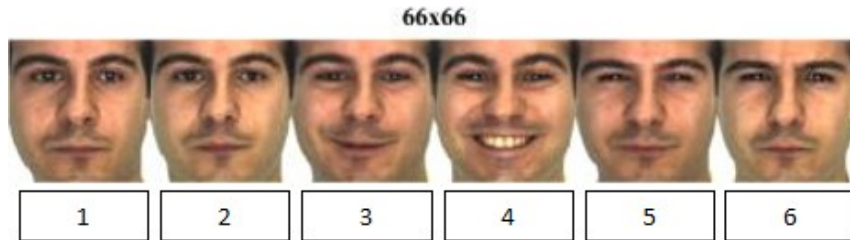


Figure 3.7: Images with different expressions and their corresponding indices.

3.4.1.3 Isolated Database Experiment

In the first experiment, training and testing were performed on the same database. AR database was used primarily in this experiment.

Same Identity Testing. In the first setup, the training set included 5 images from each subject whereas the testing set included only 1 image that was left out of the same subject. This is to ensure

that the AutoNet is trained with the maximum number of expression variations in its training phase. Such settings generally demonstrate better performance than using one image as the gallery Kan *et al.* (2014). The image selections were done in a round robin fashion for each test case and details of the six (6) test cases are given below:

1st Test Case: Images 2 ~ 6 of each identity were taken for training and image 1 of each identity was used for testing.

For each Test Case i : Test on i^{th} image of each identity and train on remaining 5 images of each identity.

Figure 3.8 demonstrates the distribution of training dataset, validation dataset and testing dataset of 100 identities from the AR database for Test Case 3. The recognition accuracies for the first experiment are shown in Figure 3.9.



Figure 3.8: Data splitting in Test Case 3. Images 1, 2, 4-6 from 75 identities were taken for training. Images of the same indices (1, 2, 4-6) from the remaining 25 identities were used for validation. Images 3 of the respective 75 identities were used for testing.

Table 3.1: Results of the tests performed with 75 identities in AR database.

Test case Index	Reconstruction				PCA	LDA	SRC	AutoNet		
	HSV-V	PCA	LDA	SRC				PCA	LDA	SRC
1	100.0	60.0	93.3	98.7	94.7	100.0	100.0	98.7	100.0	100.0
2	100.0	80.0	97.3	98.7	93.3	100.0	98.7	98.7	100.0	100.0
3	89.3	86.7	98.7	100.0	88.0	100.0	98.7	94.7	100.0	100.0
4	94.7	96.0	92.0	100.0	92.0	98.7	97.3	97.3	100.0	100.0
5	84.0	89.3	94.7	97.3	84.0	97.3	96.0	94.7	100.0	97.3
6	92.0	90.7	97.3	98.7	94.7	100.0	97.3	97.3	100.0	98.7

Discussion: As denoted by the red solid line in Figure 3.9, AutoNet framework performs consistently high and stable compared to other methods in different types of expressions. The per-

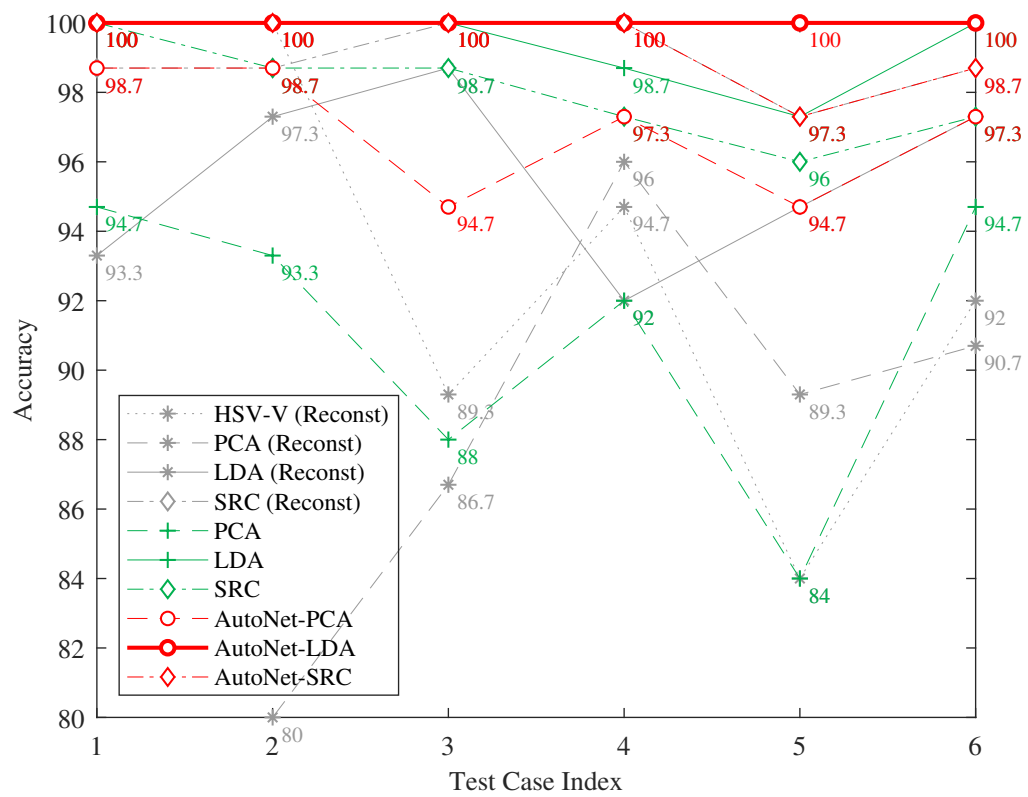


Figure 3.9: Results of the tests performed with 75 identities in AR database.

formance differences shown in Table 3.1, AutoNet-LDA vs. LDA and AutoNet-PCA vs. PCA clearly show the improvement of AutoNet over the two popularly used methods. The proposed AutoNet which can be seen as a non-linear dimensionality reduction technique clearly outperforms the other linear dimensionality reduction techniques. In addition, it performs comparatively better than the popular SRC classification algorithm.

Cross Identity Testing. Next, training was performed with images that belong to a set of identities whereas testing was performed on images that belong to another set of identities in the same database, with no overlap between the identities used for training and testing. This setting intends to evaluate the AutoNet’s generalization ability on mutually exclusive datasets that were built under the same environmental conditions such as lighting, reflection, camera alignment etc. Once the AutoNet is trained, we compare and contrast the recognition accuracies under each expression as shown below. Figure 3.10 shows the data split from AR database for Test Case 3.

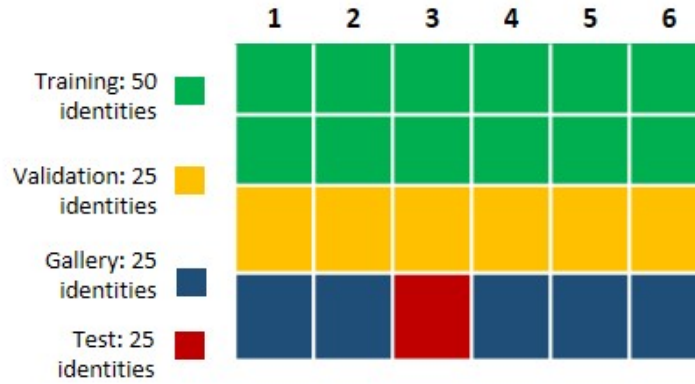


Figure 3.10: The data splitting of a cross identity test case. All images (1-6) from 50 identities were taken for training whereas the other 50 identities were split (25 identities each) for validation and testing. Each test case concerns one expression for recognition accuracies. Shown in the figure is when Images 3 were used for testing.

Table 3.2: Results of the experiments performed on cross subject arrangements.

Test case Index	Reconstruction				PCA	LDA	AutoNet		
	HSV-V	PCA	LDA	SRC			PCA	LDA	SRC
name	100.0	67.0	100.0	91.3	78.3	100.0	100.0	100.0	100.0
name	100.0	73.0	100.0	100.0	91.3	100.0	95.7	100.0	100.0
name	91.3	91.3	95.7	95.7	75.0	100.0	87.0	100.0	100.0
name	91.3	100.0	100.0	100.0	87.0	100.0	95.7	100.0	100.0
name	82.6	87.0	100.0	95.7	78.3	95.7	95.7	100.0	100.0
name	95.7	91.3	100.0	100.0	91.3	95.7	91.3	100.0	100.0

Discussion: As shown in Figure 3.11, the results show that the proposed framework performs

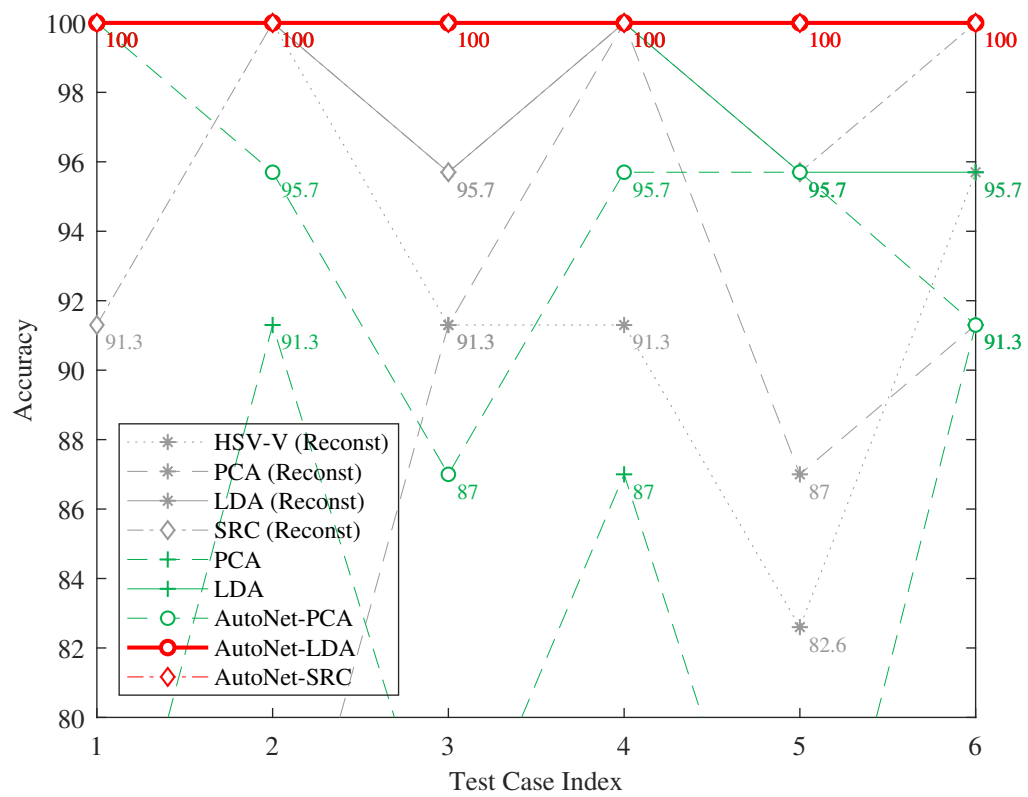


Figure 3.11: Results of the experiments performed on cross subject arrangements.

very well for cross identity datasets that were built under the same environmental conditions. All expression variations of one dataset were used in training thus exploiting the proposed AutoNet framework for observing an expression-robust feature space for classification. Table 3.2 shows the efficiency of feature representations in the learned deep latent space with the classification algorithms such as SRC, LDA, PCA. Experiments on SRC algorithm on raw images were excluded due to its non-applicability in cross identity training setup.

3.4.1.4 Combined Database Experiment

In the second experiment, training and testing were performed on a combined database built by merging 100 identities from AR database and 50 identities from Curtin database. This setting intends to evaluate the AutoNet’s flexibility of handling a large number of subjects despite the fact that the images of those subjects were obtained under different environmental conditions such as illumination, lighting etc. The six (6) Test Cases were formed in the same fashion as described in Section 3.4.1.3. Once the proposed framework is trained, we compare and contrast the recognition accuracies under each expression as shown in Figure 3.13. The distribution of training dataset, validation dataset and the testing dataset of 150 identities from the combined database for Test Case 3 is shown in Figure 3.12. All the splits consist of a random mix from the two databases.

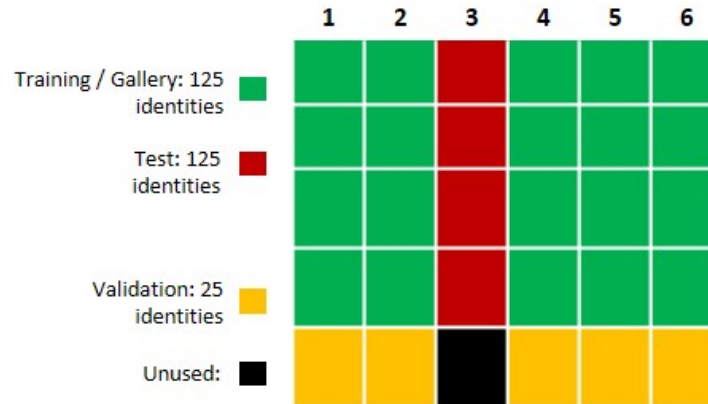


Figure 3.12: Data splitting in Test Case 3. Images 1, 2, 4-6 from 125 identities were taken for training. Images of the same indices (1, 2, 4-6) from the remaining 25 identities were used for validation. Images 3 of the respective 125 identities were used for testing.

Discussion: As shown in Figure 3.13, AutoNet still performs consistently higher than the other methods including the SRC classification algorithm. Table 3.3 shows the performance values for further reference. Clearly the proposed framework possesses the ability to perform learning on subjects that were taken under different environmental conditions which demonstrates the generalization ability of the framework.

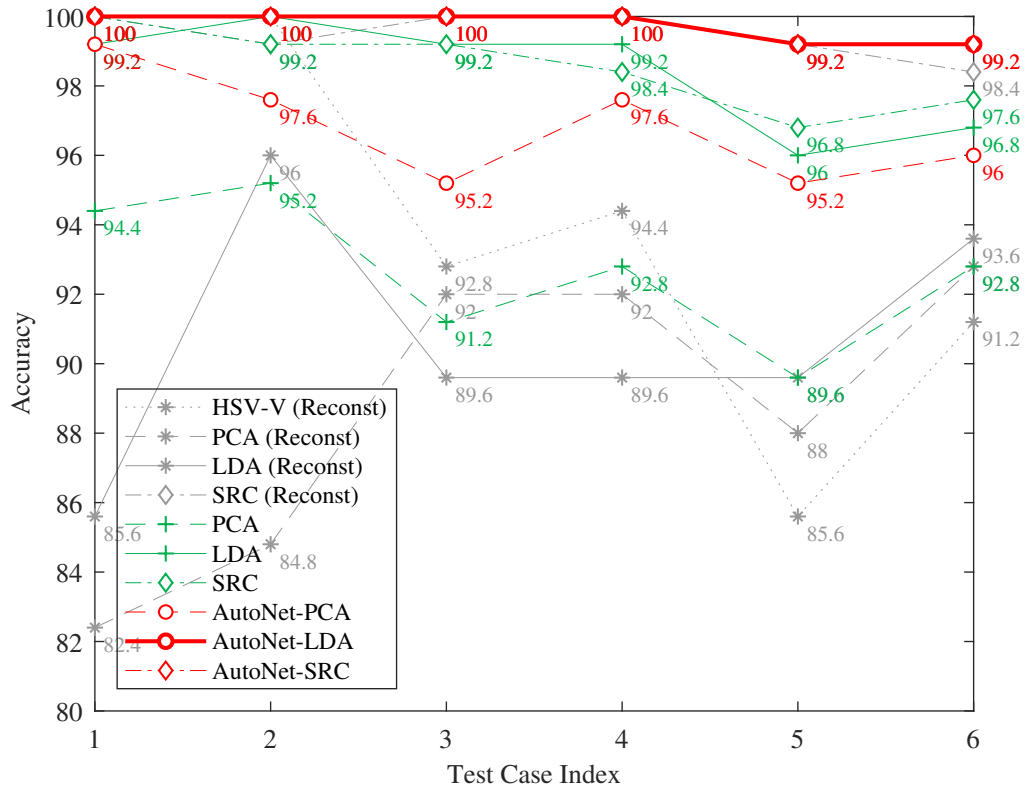


Figure 3.13: Results of the experiments performed on the combined database.

Table 3.3: Results of the experiments performed on the combined database.

Test case Index	Reconstruction				PCA	LDA	SRC	AutoNet		
	HSV-V	PCA	LDA	SRC				PCA	LDA	SRC
1	100.0	82.4	85.6	100.0	94.4	99.2	100.0	99.2	100.0	100.0
2	100.0	84.8	96.0	99.2	95.2	100.0	99.2	97.6	100.0	100.0
3	92.8	92.0	89.6	100.0	91.2	99.2	99.2	95.2	100.0	100.0
4	94.4	92.0	89.6	100.0	92.8	99.2	98.4	97.6	100.0	100.0
5	85.6	88.0	89.6	99.2	89.6	96.0	96.8	95.2	99.2	99.2
6	91.2	92.8	93.6	98.4	92.8	96.8	97.6	96.0	99.2	99.2

3.4.1.5 Cross Database Experiment

In the 3rd experiment, training was performed on the 100 identities in the AR database and testing was performed entirely on another database (Curtin). Due to the difference in the environments where the two databases are captured, this setup is more challenging and is used to demonstrate the immense generalization ability of the proposed AutoNet framework. The six (6) test cases were formed in the same manner as described in Section 3.4.1.3. In the AR database, the training set consists of 75 subjects and the remaining 25 subjects were taken as the validation set. The 50 subjects from Curtin database were used for testing. The recognition accuracies for this experiment are shown in Figure 3.14.

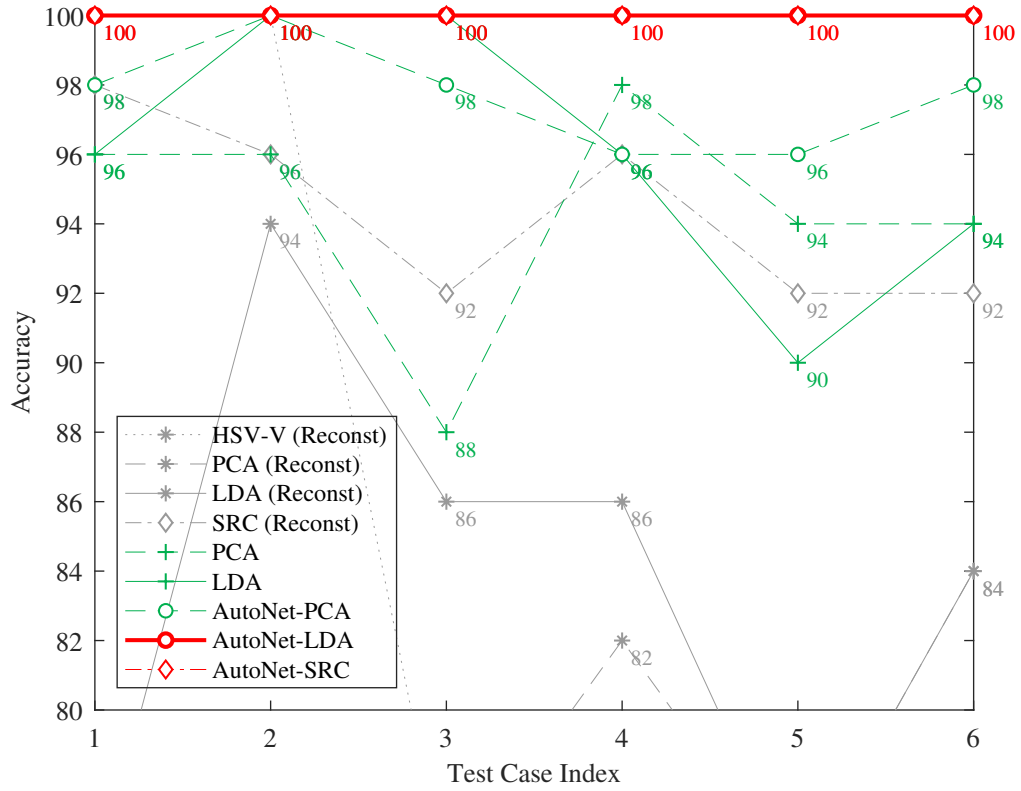


Figure 3.14: Results of the experiments performed on the cross-database arrangement.

Discussion: As shown in Figure 3.14, the proposed framework depicts a very strong generalization ability towards the unseen subjects and expressions in training, even if the training and testing images are taken in different environmental settings for different identities. As shown in Table 3.4, it consistently outperforms other methods, showing the true invariability of the proposed framework towards facial expressions. Experiments on SRC algorithm for raw images were excluded due to its non-applicability in cross database training setup.

Table 3.4: Results of the experiments performed on the cross-database arrangement.

Test case Index	Reconstruction				PCA	LDA	AutoNet		
	HSV-V	PCA	LDA	SRC			PCA	LDA	SRC
1	100.0	67.0	78.0	98.0	96.0	96.0	98.0	100.0	100.0
2	100.0	70.0	94.0	96.0	96.0	100.0	100.0	100.0	100.0
3	78.0	75.0	86.0	92.0	88.0	100.0	98.0	100.0	100.0
4	73.0	82.0	86.0	96.0	98.0	96.0	96.0	100.0	100.0
5	70.0	78.0	75.0	92.0	94.0	90.0	96.0	100.0	100.0
6	68.0	84.0	84.0	92.0	94.0	94.0	98.0	100.0	100.0

3.4.2 Civil Engineering Application - Structure Health Monitoring

Structural health monitoring (SHM) aims to assess the structural performance and evaluate the safety conditions of civil infrastructure under operational conditions. Civil structures continuously accumulate damage during their service life due to material deterioration, cyclic loading, and environmental conditions. By analyzing the measurements from various sensors installed on structures, SHM techniques detect and track the possible anomalies that could potentially produce more damage and finally lead to catastrophic structural failures with a huge loss. Measured data from the structures are widely used to detect not only the existence and location of possible damage, but also the severity of the damage (Section 2.8). Vibration-based structural identification has been popularly used to detect the possible damage in structures. Modal information, such as frequencies and mode shapes, are popularly used for structural damage detection to indicate the health conditions of civil structures.

Artificial neural networks are computational approaches based on machine learning to learn and make predictions based on data and have been applied successfully in diverse applications including vibration based damage identification in civil engineering. Yun *et al.* (2001) presented using the neural networks technique to estimate structural joint damage from modal data. It has been found that noise injection learning with a realistic noise level for each input component is effective to better understand the noise effect. Ni *et al.* (2002) proposed a statistical approach to take into account the effect of uncertainties in developing an artificial neural network model. In general, artificial neural networks are particularly applicable to problems where a significant database of information is available, but difficult to specify an explicit algorithm.

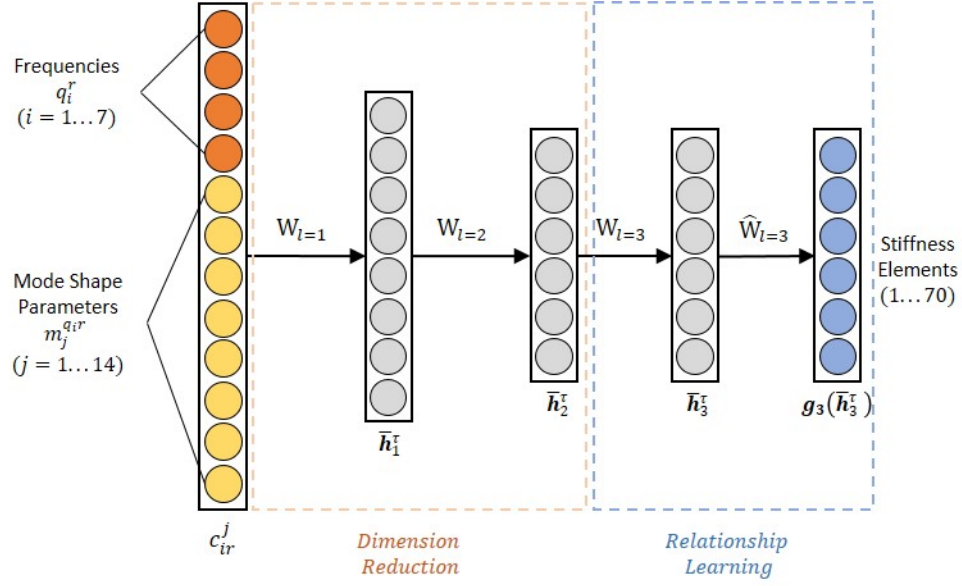


Figure 3.15: Proposed autoencoder based framework.

3.4.2.1 AutoNet For Structure Health Monitoring

AutoNet is utilized to exploit the fact that the natural frequencies and their corresponding mode shapes of a structure are related to the physical properties, such as stiffness. The modal information, such as frequencies and mode shapes, are used as the input to the proposed framework and the output will be the structural elemental stiffness parameters. A new feature is formed with all the frequencies along with their mode shapes to be fed into the proposed framework as shown below:

$$\mathbf{c}^r = [q_1^r, \dots, q_i^r, m_1^{q_1,r}, \dots, m_j^{q_i,r}] \quad (3.5)$$

where q_i^r is the i^{th} structural natural frequency included in the r^{th} sample and $m_j^{q_i,r}$ is the j^{th} mode shape parameter corresponding to the i^{th} frequency.

According to the modal analysis described in Section 2.8.1.1, the first 7 ($i = 1 \dots 7$) measured frequencies and their associated 7 x 14 mode shape values on the beam-column joints ($j = 1 \dots 14$) are considered in our study. \mathbf{c}^r is the concatenated high dimensional feature that combines these 7 frequencies and 7 x 14 (= 98) mode shapes with each frequency having 14 corresponding mode shape values. It is used in the proposed framework as the input to perform the pre-training and training as mentioned in Section 3.3.

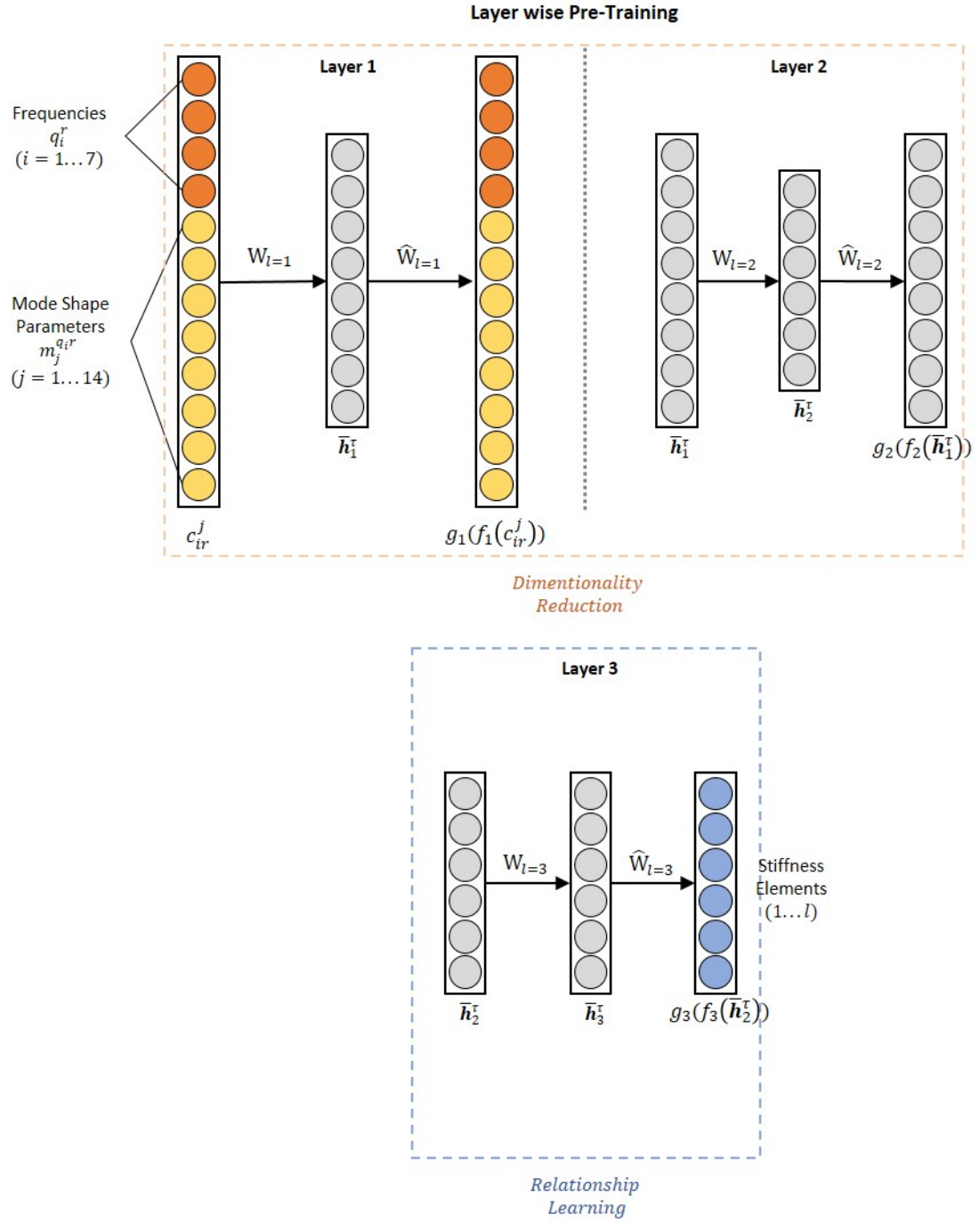


Figure 3.16: Pre-training the proposed autoencoder based framework.

Three layers are used in total in AutoNet, out of which the first two are pre-trained as the dimensionality reduction layers of the input \mathbf{c}^r . A relationship learning layer after the 2nd hidden layer is introduced to facilitate the relationship learning process by utilizing the low dimensional feature learned at the 2nd hidden layer which is in fact a better feature representation than the input itself. The output of the framework which is, in this case, the structural elemental stiffness parameters explains both the locations and the magnitudes of the damages. The layer-wise pre-training procedure is performed as shown in Figure 3.16 and once the optimal parameters are observed in the pre-training stage, both the dimensionality reduction and relationship learning components are fine-tuned (Figure 3.15) together to optimize all layers jointly as described in Section 3.3.3. Both the pre-training and fine-tuning are carried out with the full batch gradient descent optimization algorithm for simplicity. The proposed framework is expected to perform non-linear dimension reduction preserving the necessary information to facilitate the relationship learning task from input to output. Hence for the numerical studies performed on 7-storey reference model, the data generation process is explained in Section 2.8.1.2.

3.4.2.2 Data Pre-Processing

Since the concatenated feature \mathbf{c}^r contains both the frequencies and mode shapes that are measured in different scales, the frequencies and mode shapes are normalized separately. After the normalization process, both the frequencies and the mode shapes will be in the range from -0.5 to $+0.5$. This is due to the choice activation function (\tanh) that is occupied in AutoNet during the experiments. Since \tanh may lead to saturate at either tail of -1 or $+1$ and kill gradients at these regions, an effective operating range of $\tanh(x)$ could be chosen to be fallen to -0.5 to $+0.5$ and the corresponding inputs should be in the same range. Since structural damage may only occur in a few elements, sparse output vector is defined in the proposed framework to the ease the training. The accuracy and efficiency of the proposed framework is evaluated based on the generated dataset, as described in the following sections.

3.4.2.3 Evaluation of the AutoNet

The main objective of the proposed framework is to learn the relationship between the concatenated feature vector that is fed to the input and the output stiffness parameters. This is achieved via non-linear dimension reduction followed by a relationship learning phase as described above. Hence the evaluation of the proposed framework is performed separately on two components. First the effectiveness of the features learned via dimensionality reduction component is assessed. Then the quality of learning the mapping as a whole is evaluated. The next two sections describe the

evaluation process in details.

3.4.2.4 Effective Dimension Reduction

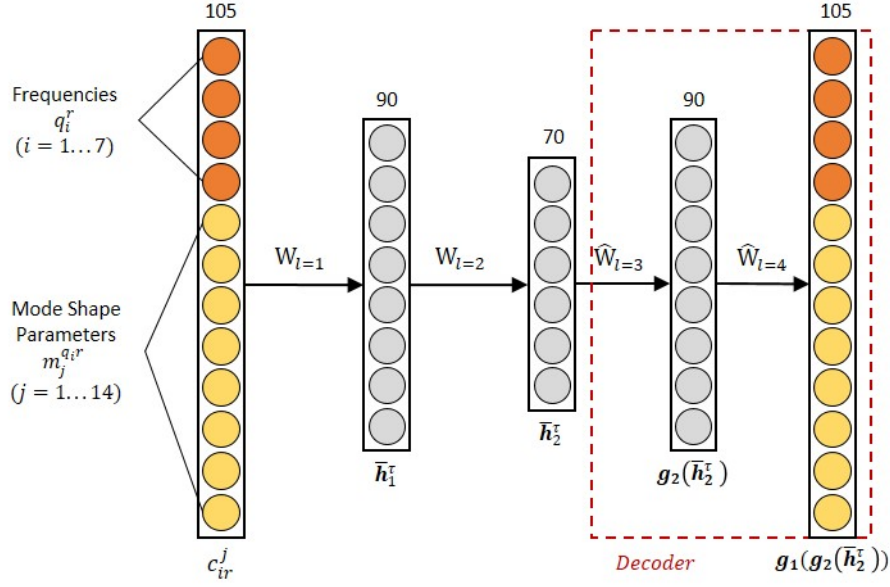


Figure 3.17: Dimensionality reduction component with decoder.

The hidden feature obtained in the 2^{nd} layer is decoded by introducing a decoder part to the dimensionality reduction component of the proposed framework as shown in Figure 3.17. The decoder facilitates the progressive dimensionality gain as opposed to the gradual dimension reduction via the encoding layers. We choose the nodes for each hidden layer as shown in Figure 3.17. The pre-training for layers 1 and 2 is performed as described in Section 3.3.1. Once pre-trained weights are obtained (layer 1, layer 2 encoding weights), these encoding weights ($W_{l=1}^*$, $W_{l=2}^*$) are utilized to tie the weights (Section 2.4.4) for the corresponding decoders as shown below:

$$\hat{W}_{l=3} = (W_{l=2}^*)^T, \hat{W}_{l=4} = (W_{l=1}^*)^T \quad (3.6)$$

The whole network is fine-tuned at the end to jointly optimize all layers as mentioned in Section 2.4.4. This optimization strategy will push the network towards reconstructing the original feature \mathbf{c}^r at the output with the decoder weights that are optimized to perform non-linear dimensionality gain. Once the model is fine-tuned, the test samples are fed forward to generate the corresponding reconstructed features at the output thus used to assess the quality of the reconstructed feature against the original feature. It is important to note that the frequencies and mode shapes of the

original feature are compared with the frequencies and mode shapes of the reconstructed feature via utilizing the MSE criterion. The reconstruction errors that are observed in different phases are summarized in Table 3.5.

Table 3.5: Evaluation results of the proposed framework with the decoder for reconstruction of original feature.

Phases	Dataset Size	Validation Error (MSE) %	Error (MSE) %
Pre-Training Layer 1	1200	23.31	5.46
Pre-Training Layer 2	1200	39.50	9.26
Fine-Tuning	1200	0.39	0.09
Testing	600	-	0.77

As shown in Table 3.5, it is conceivable that the fine-tuning indeed shows a significant improvement to the global non-linearity involved in performing the effective dimension reduction. The mean squares error is reduced to 0.09% in the final training stage after fine tuning. The testing error is as low as 0.77% indicating the accuracy of the proposed framework in performing the dimensionality reduction to represent the original input features.

3.4.2.5 Effective Relationship Learning

Since the quality of the reduced dimensional feature is satisfactory for reconstructing the original feature as demonstrated above, the quality of the proposed AutoNet framework for learning the relationship mapping is assessed in this section. The cost functions described in Section 3.3.2 in pre-training the layers and fine-tuning the whole network at the end are optimized. The effectiveness of the relationship learning process is evaluated with the mean squared error between the simulated stiffness parameters and the predicted outputs. The test samples are fed into the fine-tuned framework to generate the elemental stiffness parameters in the output. 105 modal parameters including 7 frequencies and their corresponding mode shapes on 14 beam-column joints are included in the input vector, and 70 stiffness parameters in the output vector. 90, 70 and 70 neural nodes are chosen for the first, second and third hidden layers of the proposed framework as shown in Figure 3.15. The predicted output stiffness values are compared against the expected stiffness parameters to observe how close the predicted outputs are to the ground truth. The errors that are observed in the different training phases of the proposed framework are summarized in Table 3.6.

Different layers of the proposed framework training have an effect on the global non-linearity involved in performing the effective dimensionality reduction while the error is further reduced in the latter training stages especially after fine-tuning as shown in Table 3.6. The validation

Table 3.6: Evaluation results of the proposed framework.

Phases	Dataset Size	Validation Error (MSE) %	Error (MSE) %
Pre-Training Layer 1	1200	3.84	0.90
Pre-Training Layer 2	1200	2.87	0.67
Pre-Training Layer 3	1200	2.86	0.69
Fine-Tuning	1200	2.20	0.54
Testing	600	-	2.9

error gradually decreases along with the depth of the network. It is also indicated that the error between the identified stiffness parameter values and the ground truth is 2.9% on the test dataset, demonstrating that the accuracy and efficiency of the proposed framework for structural damage identification are highly satisfactory.

3.4.2.6 Single Damage Identification

AutoNet is firstly evaluated with single structural damage cases and an example from the test data set is shown in Figure 3.18. It can be clearly seen that the identified stiffness reduction is very close to the actual value where the values of positive and negative false identifications are very small (in fact close to zero), and the identified locations of the damage are clearly distinguishable, while close to the actual locations.

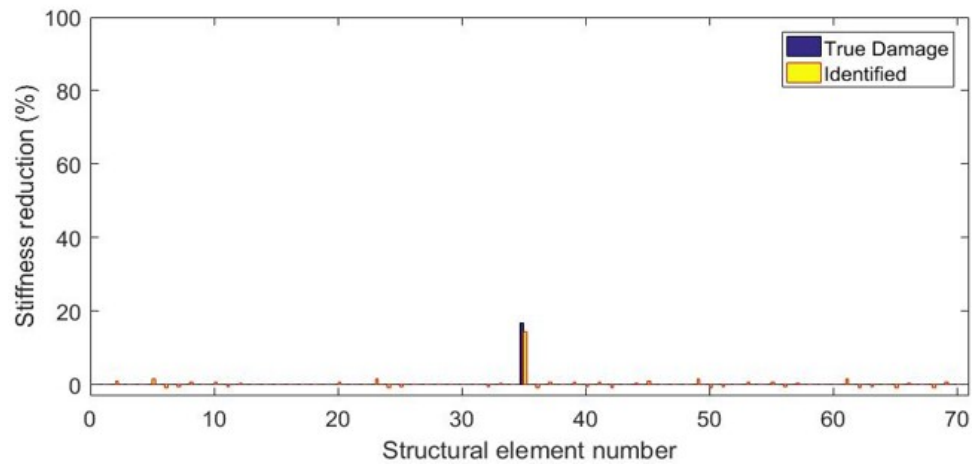


Figure 3.18: An example of single damage identification.

3.4.2.7 Multi Damage Identification

Multiple structural damages identifying is challenging, and needs more precision on the identification of accurate stiffness reductions at the exact stiffness elements compared to the single damage cases. The proposed framework is applied for such cases and an example is shown in Figure 3.19. The identified stiffness reductions are very close to the actual values with very small false identifications. The identified locations of the damage also show a very good level of accuracy.

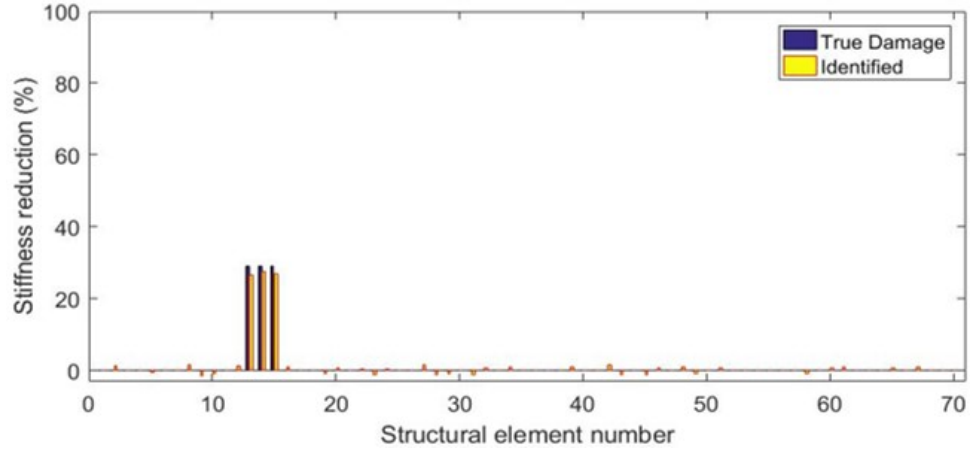


Figure 3.19: An example of multi-damage identification.

3.5 Summary

In this chapter, we proposed a basic framework for machine learning problems existing in two vastly different domains such as computer vision and civil engineering. The proposed framework consists of two components namely the dimensionality reduction component and relationship learning component that utilize the autoencoders as a generic building block. Hence the framework is easily adaptable to different problem domains due to its generic design.

In computer vision related applications, the experiments were set up to evaluate the proposed AutoNet framework in different setting varying from typical case to the most challenging case such as isolated database experiment, combined database experiments and then cross-database experiments. Isolated database experiment included a cross-identity test case to evaluate the performance of the proposed framework with the images of non-overlapping identities that were acquired in the same environment settings like lighting, camera, etc. The experiment results show that our proposed framework outperforms other state-of-the-art methods significantly thus confirming its effectiveness as a generic framework in the computer vision domain.

In civil engineering related application, the proposed AutoNet framework was utilized to train the relationship that exists in the modal information, such as frequencies and mode shapes, to structural stiffness parameters. The experiments were set up to evaluate the efficiency of the dimensionality reduction component and relationship learning component while evaluating the proposed framework on single element and multiple element damage cases. The proposed framework was able to obtain a good stiffness prediction with testing data while fusing both the frequencies and mode shapes in the input layer. Due to its generic design, it can be extended to combine more frequencies and mode shapes values and offer more information towards learning an effective mapping thus improving the performance for structural health monitoring and damage identification.

The complexity of machine learning problems in computer vision and civil engineering may vary in different magnitudes, especially when noise is taken into consideration. When data is acquired from the real world such as faces with glasses, measurement noise in structural health monitoring, etc., the noise can significantly increase the total complexity of the problem. Hence the proposed AutoNet framework needs to be extended to handle such complexities involved in data to leverage its full potential. This phenomenon is addressed in detail with the introduction of a robust and an efficient framework in Chapter 4.

Chapter 4

Robust Framework For Pattern Recognition

4.1 Introduction

In real-life applications, it is highly likely that the data used for training may contain noise thus affects the learning process of a learning algorithm drastically. For example, consider the SHM problem in civil engineering domain, the measurements acquired via sensors are prone to be noisy, and it is necessary to build robust learning algorithms to tackle this noise efficiently. It is important to investigate techniques to control the model complexity to match the problem complexity to enhance the generalization of the learning algorithm. Since the proposed AutoNet framework in the last chapter does not consider the concerns above in learning effective feature hierarchies, it starts to fail when various types of noise are included in data, especially when measurement noise and uncertainty errors in constructing the finite element model for civil structures are both present in data. Moreover, generalization of the technique is vital for building a solid ground to the proposed AutoNet framework in order for it to be applicable in real-life machine learning problems.

4.1.1 Generalized Machine Learning

A simple machine learning algorithm would work very well on a wide variety of important problems, but it may not succeed in achieving the highest level of generalizability on previously unobserved inputs, especially when the complexities involved in a problem domain is considerably high. In fact, this was one of the main reasons for the failure of traditional machine learning algorithms compared to deep learning. The central problem in machine learning is to make an algorithm generalize well on new inputs, not just on the training data.

In general, when training a machine learning model, we have access to a training set where some error measured on the training set (training error) can be calculated. We aim to reduce this training

error which is also known as the optimization problem. The separation of machine learning from optimization occurs when the generalization error, also called the test error, is required to be low. The generalization error is defined as the expected value of the error on new input. Here the expectation is taken across different possible inputs, drawn from the distribution of inputs we expect the system to encounter in practice. Generalization error of a machine learning model is typically estimated by measuring its performance on a test set of examples that were collected separately from the training set. No overlap between the two sets is expected. Statistical learning theory Vapnik (1995) shows the performance on the test set can be affected via observing only the training set under certain assumptions known collectively as the i.i.d (independent and identically distributed) assumptions on how the training and test set are collected.

A probability distribution over datasets called the data generating process is expected to generate the training and test data. The assumptions are that the samples in each dataset are independent of each other and that the test set and training set are identically distributed, drawn from the same probability distribution. These assumptions allow us to describe the data generating process with a probability distribution over a single sample. The same distribution is then utilized to generate every training sample and every test sample. Hence this shared underlying distribution is the data generating distribution which could be denoted by p_{data} . This probabilistic framework and the i.i.d. assumptions allow us to study the relationship between training error and test error mathematically. One immediate connection that can be observed between the training and test error is that the expected training error of a randomly selected model is equal to the expected test error of that model. Suppose we have a probability distribution $p(\mathbf{x}, y)$ and we sample from it repeatedly to generate the training set and the test set. For some fixed value \mathbf{w} , the expected training set error is precisely the same as the expected test set error due to both expectations being formed using the same dataset sampling process. The name assigned to the dataset is the only difference between the two conditions. In utilizing a machine learning algorithm, the parameters are not fixed before both datasets are sampled. Instead, we sample the training set first, then use it to choose the parameters to reduce training set error, then sample the test set. Under this process, the expected test error is greater than or equal to the expected value of the training error. The factors determining how well a machine learning algorithm perform will depend on its ability to make both the training error and the gap between the training and test errors small. These two factors correspond to the two central challenges in machine learning: underfitting and overfitting (Section 2.6). A model unable to obtain a sufficiently low error value on the training set leads to underfitting while the growth of the gap between the training and test errors lead to overfitting.

A model's capacity can be altered to control the overfitting or underfitting phenomenon. Informally, a model's capacity is its ability to fit a wide variety of functions. Models with low capacity may struggle to fit the training set. Models with high capacity can overfit by memorizing properties of the training set that do not serve them well on the test set. Machine learning algorithms

generalize well toward unseen data when the capacity is appropriate with respect to the true complexity of the task. The amount of training data utilized in the training phase of such algorithms will be beneficial in generalization as well. Models with insufficient capacity are unable to solve complex tasks. Models with high capacity can solve complex tasks, but when their capacity is higher than necessary for the present task, they may overfit.

Modern ideas about improving the generalization of machine learning models are refinements of thought dating back to philosophers at least as early as Ptolemy. Many early researchers invoke a principle of parsimony which is now widely known as Occam’s razor Rasmussen and Ghahramani (2001). It states that among competing hypotheses that explain known observations equally well, one should choose the ”simplest” one. This idea was formalized and made more precise in the 20th century by the founders of statistical learning theory Vapnik and Chervonenkis (2015); Vapnik (2006); Blumer *et al.* (1989); Vapnik (1995).

Statistical learning theory provides various insights of quantifying model capacity. Among these, the most popular is the Vapnik-Chervonenkis dimension or VC dimension where it measures the capacity of a binary classifier. The VC dimension is defined as the largest possible value of m for which there exists a training set of m different \mathbf{x} points that the classifier can label arbitrarily. According to the statistical learning theory, Quantifying the capacity of the model allows making quantitative predictions. Furthermore, the discrepancy between the training error and generalization error is bounded from above by a quantity that grows as the model capacity grows but shrinks as the number of training samples increases Vapnik and Chervonenkis (2015); Vapnik (2006); Blumer *et al.* (1989); Vapnik (1995). It is defined as:

$$E_{test} \leq E_{train} + O\left(\sqrt{\frac{d_{VC}}{N}} \log N\right) \quad (4.1)$$

where E_{train} , E_{test} , N , d_{VC} is the training error, generalization error, number of training samples and model complexity respectively. These bounds provide intellectual justification that machine learning algorithms can work, but they are difficult to be used in practice when working with deep learning algorithms. It is in part because the bounds are often quite loose and also because it can be quite challenging to determine the capacity of deep learning algorithms. The problem of determining the capacity of a deep learning model is especially difficult because the productive capacity is limited by the capabilities of the optimization algorithm, and there is still little theoretical understanding of the very general non-convex optimization problems involved in deep learning.

Since simpler functions are more likely to generalize, i.e., to have a small gap between the training and test error, it is important to choose a sufficiently complex hypothesis to achieve low training

error. Typically, training error decreases until it asymptotes to the minimum possible error value as model capacity increases (assuming the error measure has a minimum value). Typically, generalization error has a U-shaped curve as a function of model capacity, as illustrated in Figure 4.1

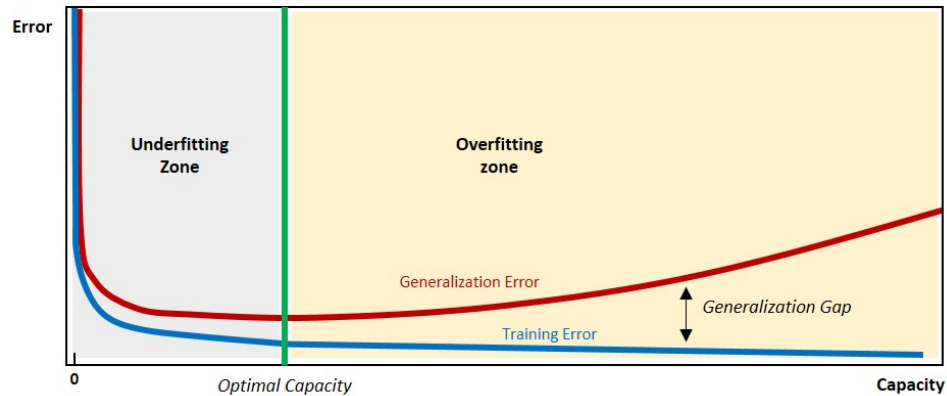


Figure 4.1: Images with different expressions and their corresponding indices.

The training and generalization error vary as the size of the training set varies. Expected generalization error can never increase as the number of training samples increases. Note that it is possible for the model to have the optimal capacity and yet still have a significant gap between the training and generalization error. In such situations, it is required to expand the training set to reduce this gap.

One way to control the capacity of a learning algorithm is by choosing its hypothesis space, the set of functions that the learning algorithm is allowed to select as being the solution. For example, the linear regression algorithm has the set of all linear functions of its input as its hypothesis space. Linear regression can be generalized to include polynomials, rather than just linear functions, in its hypothesis space. Doing so increases the model's capacity. Increasing the number of hidden nodes in a layer has a similar effect depending on the activation function utilized.

Another way is to give a preference to a learning algorithm for one solution in its hypothesis space to another. It means that both functions are eligible, but one of them is preferred. The unpreferred solution is chosen if it fits the training data significantly better than the preferred solution. For example, the training criterion for linear regression can be modified to include weight decay to minimize the sum of mean squared error on the training data while having a preference for the weights to have smaller squared L2 norm. Thus the training criterion can incorporate an extra term that can be minimized combinedly to achieve smaller weights. There are many ways of expressing preferences for different solutions, both explicitly and implicitly. Together, these different approaches are known as regularization. Regularization is any modification to a learning algorithm that is intended to reduce its generalization error but not its training error. Regularization

is one of the central concerns in the field of machine learning, rivaled in its importance only by optimization.

4.1.2 Regularization

Regularization has been used for decades antecedent to the advent of deep learning. Linear models such as linear and logistic regression allow simple, straightforward, and effective regularization strategies. Limiting the capacity of the model is a popular regularization strategy for models, such as neural networks, linear regression, or logistic regression. It is done via adding a parameter to the objective function $J(\theta)$ as shown in Section 2.6.2.

Many regularization strategies exist in the machine learning context. Extra constraints on a machine learning model where it adds restrictions on the parameter values and extra terms in the objective function which can be seen as soft constraints on the parameter values, etc. If chosen carefully, these additional constraints and penalties can lead to improved performance on the unseen data (test set). Sometimes these penalties and constraints are designed to encode specific kinds of prior knowledge. Other times, these constraints and penalties are designed to express a general preference for a simpler model class to promote generalization. Sometimes constraints and penalties are essential to make an underdetermined problem determined. Other forms of regularization include ensemble methods that combine multiple hypotheses that explain the training data. In the context of deep learning, regularizing estimators is utilized as the basis for most regularization strategies.

Regularization of an estimator works by trading increased bias for reduced variance as mentioned in Section 2.6. The efficiency of a regularizer depends on how well it makes a profitable trade, reducing variance significantly while not overly increasing the bias. The goal of regularization is to take a model from the underfitting or overfitting regime into the perfect fit regime that can match the true data generating process (Figure 4.1). In practice, overly complex models do not necessarily include the true data generating process or the target function, or even a close approximation of either. It is nearly impossible to have access or have complete knowledge on true data generating process thus impossible to know if the models being estimated include the generating process or not. However, most applications of deep learning algorithms are to domains where the actual data generating process is almost certainly outside the model family. Deep learning algorithms are typically applied to extremely complicated domains such as images, audio sequences, and text, for which the true generation process essentially involves simulating the entire universe. In simple terms, it is an attempt to fit a square peg (the data generating process) into a round hole (model family). Choosing the model of the right size, with the right number of parameters may not always be the straightforward strategy for controlling the complexity of the model, but regulariza-

tion. In practical deep learning scenarios, it is possible to find that best fitting model (in the sense of minimizing generalization error) which is a large model that has been regularized appropriately.

The no free lunch theorem (Wolpert, 1996) states that there is no best machine learning algorithm, and, in particular, no best form of regularization. Instead, the form of regularization must be chosen to suit the particular task of interest. In general, the philosophy of deep learning is that a wide range of tasks (such as all of the intellectual tasks that people can do) may all be solved effectively using very general-purpose forms of regularization. In fact, developing more effective regularization strategies has been one of the major research efforts in the field. A revision of several regularization strategies is presented in Section 2.6.

In the previous chapter, constrained deep learning was overlooked in favor of pre-training scheme which affects the generalizability of the proposed approach on noisy inputs. The model is extended to couple with effective regularization techniques to enhance its generalizability as discussed in the following sections.

4.2 Proposed Extended Framework

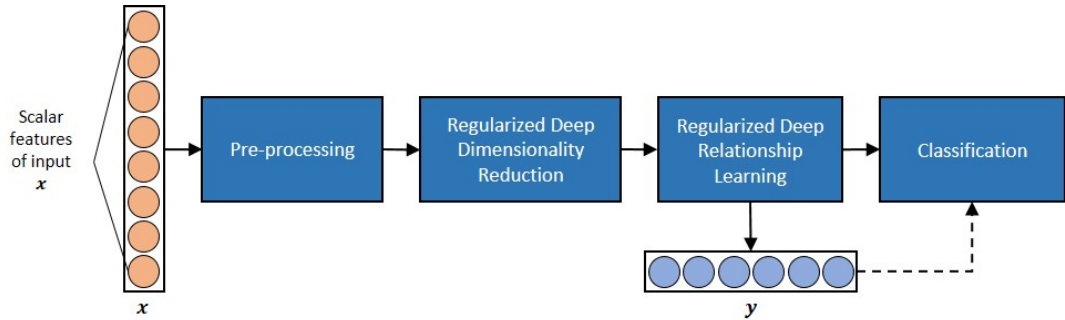


Figure 4.2: Highlevel view of the proposed extended framework.

The curse of dimensionality phenomenon makes many machine learning problems become exceedingly difficult due to the dimensionality of the data being high. The number of possible distinct configurations for a set of variables increases exponentially as the number of variables increases. A high dimensional input feature may contain unnecessary information (noise) due to redundant data thus shrinking the dimensionality with effective regularization strategies are essential in achieving stable performance rates on the test datasets. The noise in the input will negatively impact the performance of the non-regularized model thereby affecting its generalizability on the test data. We extend the proposed AutoNet framework described in Chapter 3, on both the dimensionality reduction and relationship learning components with effective regularization strategies to deal with such cases. Furthermore, we introduce a preprocessing stage in the framework to aid the

final goal of learning the mapping from a given input to an output. The new framework is shown in Figure 4.2. The objective function of the extended framework can be generically expressed as below:

$$J(w) = J_{cost} + \lambda R(w) \quad (4.2)$$

where $R(w)$ is called the regularization term and λ is called the regularization strength a hyperparameter that is chosen via cross-validation as discussed in Section 2.6.3.

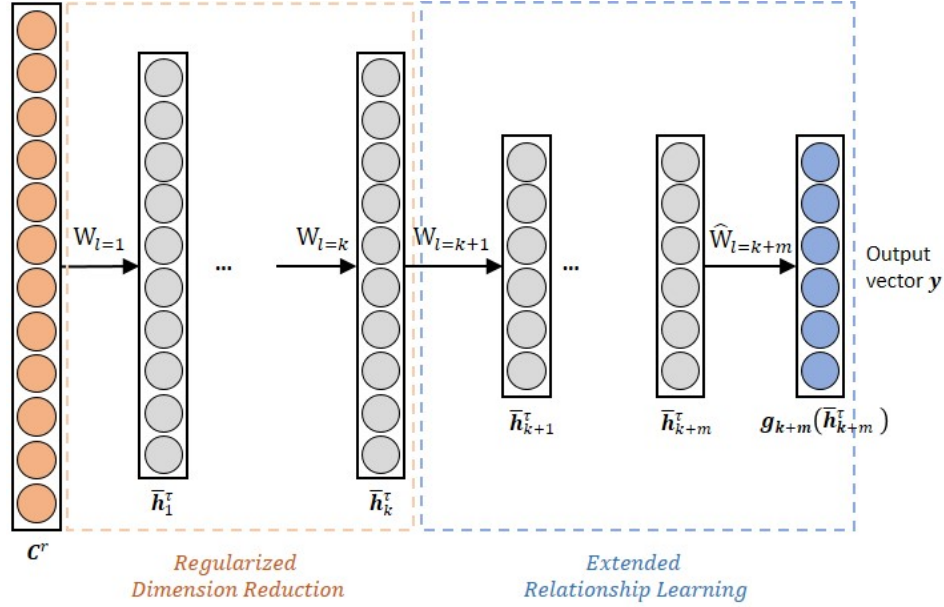


Figure 4.3: Architecture of the proposed extended framework.

4.2.1 Extended Dimensionality Reduction Component

The burden of choosing a suitable number of hidden nodes for each layer in the dimensionality reduction component has been a problem in the proposed AutoNet framework introduced in Section 3.3. This is due to the fact that choosing a hidden layer dimension r for the autoencoders as the same or larger than the input dimension d could lead to failure of the autoencoder in learning efficient features than the input, because the AE will simply learn the identity function as the mapping copying the values at the input to the output. Hence an under-complete representation where $r < d$ is necessary to learn useful patterns in the input of the dimensionality reduction component as introduced in Section 3.3. This limitation could be avoided by using effective regularization term for $R(W)$ as shown below:

$$J_{weight}(W, \hat{b}) = \frac{1}{2} \sum_{l=1}^2 \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} (w_{ji}^l)^2 \quad (4.3)$$

where w_{ji}^l represents an element in W^l and s_l denotes the number of units in the l^{th} layer. $R(w)$ is also known as the L2-weight decay term and we utilize this term on all weights of the deep learning network in both pre-training and training. The L2-weight decay term denoted in Eq. 4.3 constrains the weights of a neuron to be smaller thus having the hidden node dimensionality in control with respect to the problem complexity. This avoids the overfitting in the overall training process. During the pre-training, this mechanism of regularization helps to constrain the number of hidden nodes in a layer by pushing some weights to zero (thereby making the inputs to some nodes close to zero, making the neuron's response less significant). In addition, L2-weight decay constrains the hidden nodes of a layer Bengio (2009) thus allowing the dimensionality reduction component to utilize hidden layers with the number of hidden nodes same as its input. A typical autoencoder without L2-weight decay would learn the identity mapping from its input to output. The burden of choosing a suitable number of hidden nodes for each layer is thus handled to a certain extent with the introduction of this constraint.

This regularization strategy could be successfully applied in both the dimensionality reduction component and, the relationship mapping components as shown in Figure 4.2 to make the framework robust to noisy inputs. In this manner, the proposed extended framework is forced to utilize the required amount of hidden nodes in each layer to retain only the required information in establishing the relationship between the learned representation and the expected output while encoding the original feature vector.

4.2.2 Extended Relationship Learning Component

The main objective of this component is to learn the relationship between the reduced dimensional feature h_k^r and the output as shown in Figure 4.3. In Section 3.3.2 only a single layer with 'tanh' activation function was utilized to achieve this task since having more layers could suffer from overfitting. It limits the true capacity of the framework to perform better in relationship learning phase. In the proposed extended framework, the relationship learning component was given the freedom to have deeper layers with following regularized cost formulation for each layer:

$$J_{cost}^q(W, b) = J_{MSE}^q(W, b) + \lambda J_{weight}^q(W, b) \quad (4.4)$$

The same weight decay function as described in Eq. 4.3 is utilized. The reconstruction loss function ($J_{MSE}^q(W, b)$) of each mapping layer can be further derived as:

$$J_{MSE}^q(W, b) = \sum_{r=1}^N \|o^r - g_q(f_q(h_q^r))\|_2^2 \quad (4.5)$$

where $q = k + 1, \dots, k + m$ for the m^{th} layer in the relationship learning component, $g(\cdot)$ and $f(\cdot)$ are the decoder and the encoder functions respectively, h_{q-1}^r is the lower dimensional representations that are obtained at the layer for the r_{th} sample and o^r is the labeled output vector for the r_{th} sample.

Every layer is defined to have an effect on the global nonlinearity involved in performing the effective relationship learning process. In this way, the error is further reduced in the latter layers. A significant improvement to the global nonlinearity of the problem is achieved via stacking the layers for the final joint optimization. Pre-training for all layers are conducted with the full batch scaled conjugate gradient backpropagation algorithm as discussed in Section 2.5.3. Once the optimal parameters are observed, the whole network is fine-tuned again to optimize all the layers jointly as mentioned in Section 3.3.3.

4.2.3 Effective Enforcement Of Sparsity

A popular approach to constraint the information content in the latent representation is to make it sparse or low dimensional Glorot *et al.* (2011). A Rectified Linear Unit (ReLU) has special properties over the alternative non-sparse activation functions in the context of deep neural networks such as information disentangling, efficient variable-size representation, linear separability, distributed sparsity, and sparsity-induced regularization, etc. It is also beneficial over other activation functions from optimization point of view as mentioned Section 2.4.5. Nevertheless, applying a too strong sparsity constraint may hurt the network prediction performance for an equal number of neurons, since it reduces the effective capacity of the framework. In the proposed extended framework, in the dimension reduction component, ReLU is utilized carefully to regularize the reconstruction loss function by introducing both a sparsity-inducing term and a weight decay function. The objective cost function is hence defined as:

$$J_{cost}(W, \hat{b}) = J_{MSE}(W, \hat{b}) + \lambda J_{weight}(W, \hat{b}) + \beta J_{sparse}(W, \hat{b}) \quad (4.6)$$

where $J_{MSE}(W, \hat{b})$ is the reconstruction loss function; $J_{weight}(W, \hat{b})$ is the weight decay function (L2 regularization of all weights); $J_{sparse}(W, \hat{b})$ is the sparsity penalty term as defined in Section 2.4.3 which is employed for a better de-noising ability; λ and β are the regularization parameters to balance the reconstruction accuracy and the sparsity constraints on the solution. The optimal parameter values are obtained via cross validation (Section 2.6.3). The weight decay term $J_{weight}(W, \hat{b})$ defined in Eq. 4.3 is utilized to avoid overfitting. The layers pre-trained with such constrained autoencoders are stacked together to form a deep architecture and learn a robust representation for the input. This strategy enables us to introduce deeper layers in dimensionality reduction component unlike the proposed AutoNet framework in Section 3.3. The robust feature space observed from the last hidden layer of the dimensionality reduction component will preserve the useful information to constraint the mapping to output in the relationship learning component as discussed in Section 4.2.2.

4.2.4 Constraining the Complexity of Framework

The methods discussed so far increase or decrease the framework's capacity by adding or removing hidden nodes/layers while constraining the input weights associated with them. They are equivalent to adding or removing functions from the hypothesis space of solutions the framework is able to choose. In order to avoid overfitting effectively, the following techniques are utilized in addition to the methods described above.

4.2.4.1 Early stopping

Early-stopping combats overfitting by interrupting the training procedure once a framework's performance on a validation set gets worse. A validation set is a set of examples that we never use for gradient descent, which is not a part of the test set either. The validation examples are considered to be representative of future test examples. Early stopping is effectively tuning the hyper-parameter number of epochs/steps.

Intuitively as the framework sees more data and learns patterns and correlations, both the training and test error goes down. After enough passes over the training data the it may start overfitting and learning noise in the given training set. In this case, the training error would continue going down while the test error would get worse no matter how well we generalize. Early stopping is all about finding the right moment with minimum test error (Figure 4.1).

4.2.4.2 Dataset augmentation

An overfitting model (neural network or any other type of model) can perform better if the model processes more training data. While an existing dataset might be limited, for some machine learning problems there are relatively easy ways of creating synthetic data. For images, some common techniques include translating the picture by a few pixels, rotation, scaling etc. For classification problems, it is usually feasible to inject random negatives. For example, unrelated pictures.

There is no general recipe on how the synthetic data should be generated and it varies a lot from problem to problem. The general principle is to expand the dataset by applying operations which reflect real-world variations as closely as possible. Having better dataset in practice significantly helps the quality of the models, independent of the architecture.

4.2.4.3 Data Pre-Processing

The input can be pre-processed to un-correlate the feature dimensions and standardize the spread of the data in each dimension. It improves the effectiveness of dimensionality reduction and performed via the data whitening process, which is a linear transformation that transforms a vector of random variables with a known covariance matrix into a set of new variables with the identity covariance matrix. Data whitening process un-correlates and spheres (standard deviation equal to 1) the data based on PCA and provides pre-processed datasets with less redundancy to perform the training, validation, and testing of the network. The detailed process is explained in Section 2.2.2. The whiten components that correspond to eigenvalues less than $1e - 10$ are discarded and the rest is kept preserving as much as the information existed in the original data.

4.3 Applications

The proposed extended framework is evaluated in the same application domains discussed in Section 3.4 for its wide applicability and easy adaptivity. With the regularization strategies introduced in Section 4.2, AutoNet in Chapter 3 can now be extended with many layers resulting in a deeper network for effective feature learning from noisy inputs in both the domains where one involves visual signals and the other involves a set of numerical values of measurements.

To evaluate the effectiveness of enforcing sparsity on the proposed extended framework, two variants named as **AutoDNet** and **SAF** are introduced for the experiments performed below. Au-

toDNet is the extended version of the proposed AutoNet framework in the previous chapter with weight decay constrain while SAF denotes the proposed extended framework with both weight decay and sparsity constraints along with other regularization strategies.

4.3.1 Computer Vision Application - Glass Removal For Face Recognition and Verification

Glasses/spectacles are widely worn in the real world by people as a fashion statement or a treatment for vision impairments (e.g shortsighted). Thus the presence of glasses can be considered as a face semantic feature that has a potential to negatively impact recognition proficiency Righi *et al.* (2012). In order to improve the performance in face recognition in such situations, detection and removal of glasses on faces are necessary. There have been a number of researches conducted on glasses-related applications in pattern recognition and computer vision community in the last two decades. Jiang *et al.* (2000) might be one of the earliest attempts to detect the presence of glasses. Their approach is based on the level of intensity difference measurement surrounding the eyes. The assumption is that it is highly likely for glasses to have significantly different colour compared to facial skin, leading to a high level of intensity discontinuity around eyes, indicating their presence. Another approach proposed in the same period was by Jing *et al.* (2000) which incorporated the Bayes rule on edge features extracted from Sobel filter. They also attempted to remove the contour of the glasses by applying an adaptive median filter. A few years later, Wu *et al.* (2004) adopted the idea of Markov-chain Monte Carlo technique for localizing the glasses segment and passing it through a reconstruction process to remove the glasses for image synthesizing purpose. The performances of glasses detection and removal by these approaches are generally limited to visual perception. There is no experiment conducted to measure the effect of these approaches on facial recognition.

Some other implementations aimed at improving the facial recognition rate. For instance, Wang *et al.* (2010) proposed an idea to localize glasses with Active Appearance Model (AAM) technique Cootes *et al.* (2001) and remove it via a reconstruction process with PCA. Despite a significant improvement made on the accuracy of facial recognition, there is no experiment to evaluate its glasses detection accuracy by determining whether a person is wearing glasses or not. Another unique approach was proposed by Heo *et al.* (2004) where they combined the information from both visible features (pixel values) and thermal infrared (IR) images. This idea of utilizing thermal infrared was extended further in Wong and Zhao (2013) by attempting facial reconstruction on infrared space relying on information around the eyes from the normal image. This novel way of including extra data from thermal infrared images shows a significant performance improvement. However, it requires a specific device used for capturing thermal infrared images. We believe it is

more preferable to focus only on colour/grey-scale images since they are more widely available.

4.3.1.1 AutoDNet For Glass Removal

We utilize AutoDNet as a component in a completely autonomous glasses removal system which can work on any frontal face image to remove glasses effectively. Firstly, the concept of pictorial-tree-structured models introduced in Liang *et al.* (2014); Liang (2017) trained with many glassed-faces is utilized to detect the presence of the glasses and extract the landmarks with the aim to improve the performance of the next phase which is, the glass removal.

The glasses removal phase was conducted by recovering the region of interest via state-of-the-art image reconstruction technique called Non-Local Colour Total Variation (NLCTV) (Duan *et al.*, 2015). NLCTV performs glasses removal process via reconstruction by inpainting with the help from a mask as the NLCTV inpainting approach requires the boundary information. It considers the glasses segment as noise and reconstructs it based on the surrounding skin texture.

Next, the glasses removal process is further refined by AutoDNet to remove last traces of glasses and slight light reflection. This step is essential to achieve the best results. These two approaches were arranged in a cascade structure to act as a double-layered filters to remove the glass (noise) on the face images. Since the NLCTV inpainting can remove most traces of the glasses, the denoising via AutoDNet will be highly efficient for face classification. The whole framework is summarized in Figure 4.4 and the complete system pipeline can detect the presence and location of glasses automatically without assuming its existence (able to distinguish faces with and without glasses).

Due to the difficulties in acquiring publicly available databases specifically designed for glasses model training, various glasses and non-glasses images were compiled from CMU multiPIE Gross *et al.* (2010) as training dataset. In fact, this data augmentation strategy positively impacts the performance in training the proposed deep learning framework above. The robustness of the system is evaluated on various face databases based on three well-known classification techniques PCA, LDA, and SRC (as mentioned in Section 2.2).

4.3.1.2 Face Recognition

The complete system is assessed against a large dataset named CAS-PEAL-R1 Gao *et al.* (2008) with greyscale images. This is due to the fact it contains a large number of participants (438) on

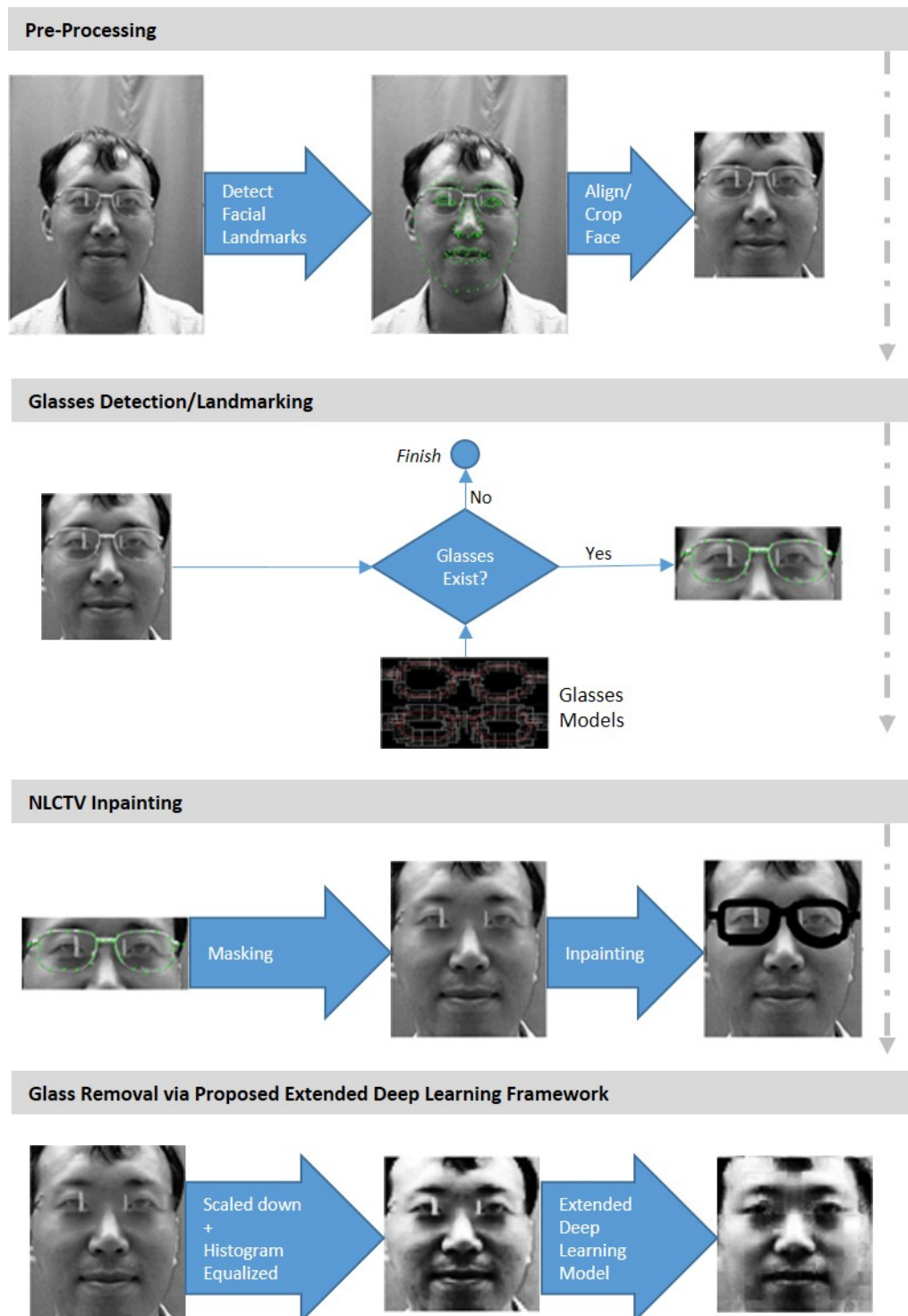


Figure 4.4: Complete system pipeline for glasses detection and removal.

various glasses.

However, some of the images are affected by strong illumination on the lenses occluding significant part of the eyes. The accessories section consists of 3 different glass types. Since occlusion is not the focus of attention here, a thorough selection resulting in 340 chosen subjects was made to subset the dataset for experiments. In Chapter 6, a complex deep learning system design is introduced to handle such cases of occlusions in general.

This experiment was conducted with a cross-identity setup where the training image set and gallery/testing image set do not share the same subject. The training set consists of 4 non-glasses images including neutral face per subject to learn the transformation function of AutoDNet. The non-neutral facial expressions are considered as 'noisy' faces, and we want to train AutoDNet to reconstruct them into neutral faces via supervised learning. The trained AutoDNet is used to attempt further reconstruction to remove the remaining traces of glasses after inpainting. In total, we choose 98 subjects in this set. On the other hand, the testing involves one neutral face image as the gallery and two glasses images as the query from each identity for the remaining 242 subjects. The results were considered mainly on three scenarios: face with glasses, inpainted glasses (NLCTV), and reconstructed glasses (NLCTV + AutoDNet). Once the glasses are completely removed, linear classification approaches (LDA, PCA, SRC in Section 2.2) were utilized to perform face recognition. The result is summarized in Figure 4.5.

Faces with the presence of glasses display the lowest performance rate as expected due to the disruption of added noise (which is, in this case, the glasses) on the face. Inpainted glasses appear to provide slight improvement towards recognition rate. The observations could lead to two possibilities. Firstly, due to the restricted availability of data, we can only use CAS-PEAL-R1 which contains only grey-scale images. However, the NLCTV inpainting is able to reconstruct the image texture based on the color information. Its full potential is utilized in this case. Secondly, the proportion of the inpainted area compared to the size of the whole face is relatively small. Even though the result is better, the changes only affect local parts of the face (around eyes). This is why we added another layer of glasses filter via AutoDNet. Its de-noising process covers the whole face including glasses regions. In addition, since the NLCTV has removed most of the glasses segments, it becomes easier for AutoDNet to de-noise the remaining traces of the glasses and the slight lens reflections. Significant improvement is achieved with the proposed scheme. The combination of NLCTV and AutoDNet reduce the error rate by approximately **50%**, **52.25%**, and **57.09%** for PCA, LDA, and SRC respectively.

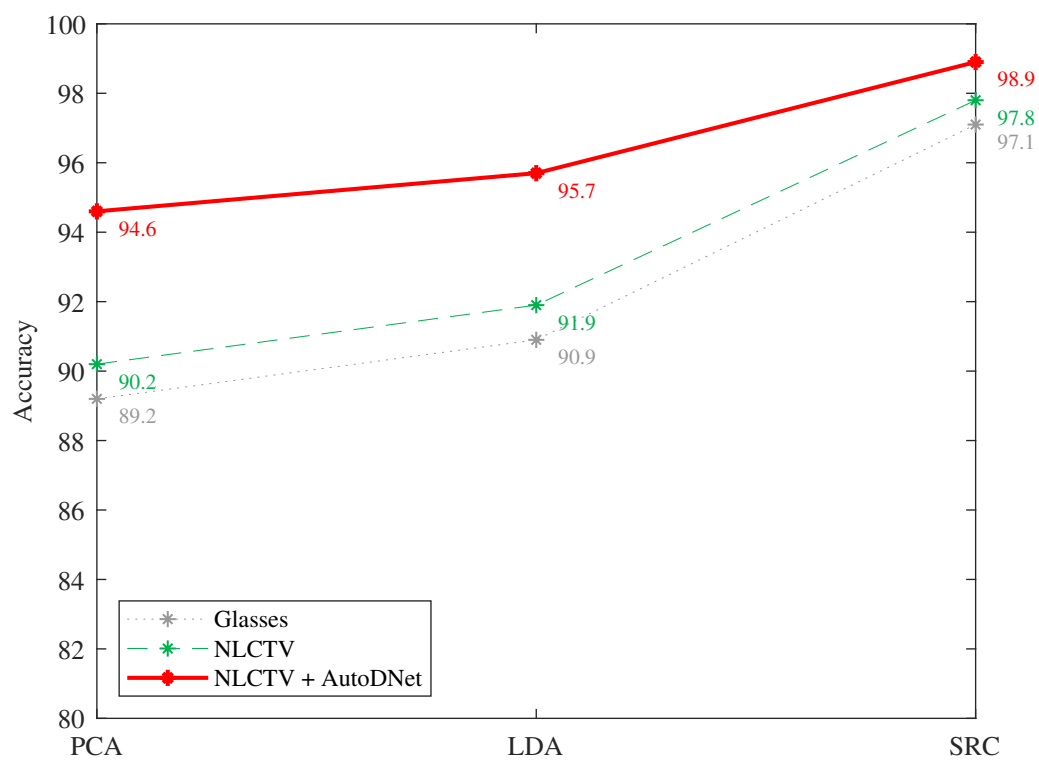


Figure 4.5: Facial recognition with classification approaches PCA, LDA, and SRC. This result proves that removing presence of glasses improves the facial recognition rate.

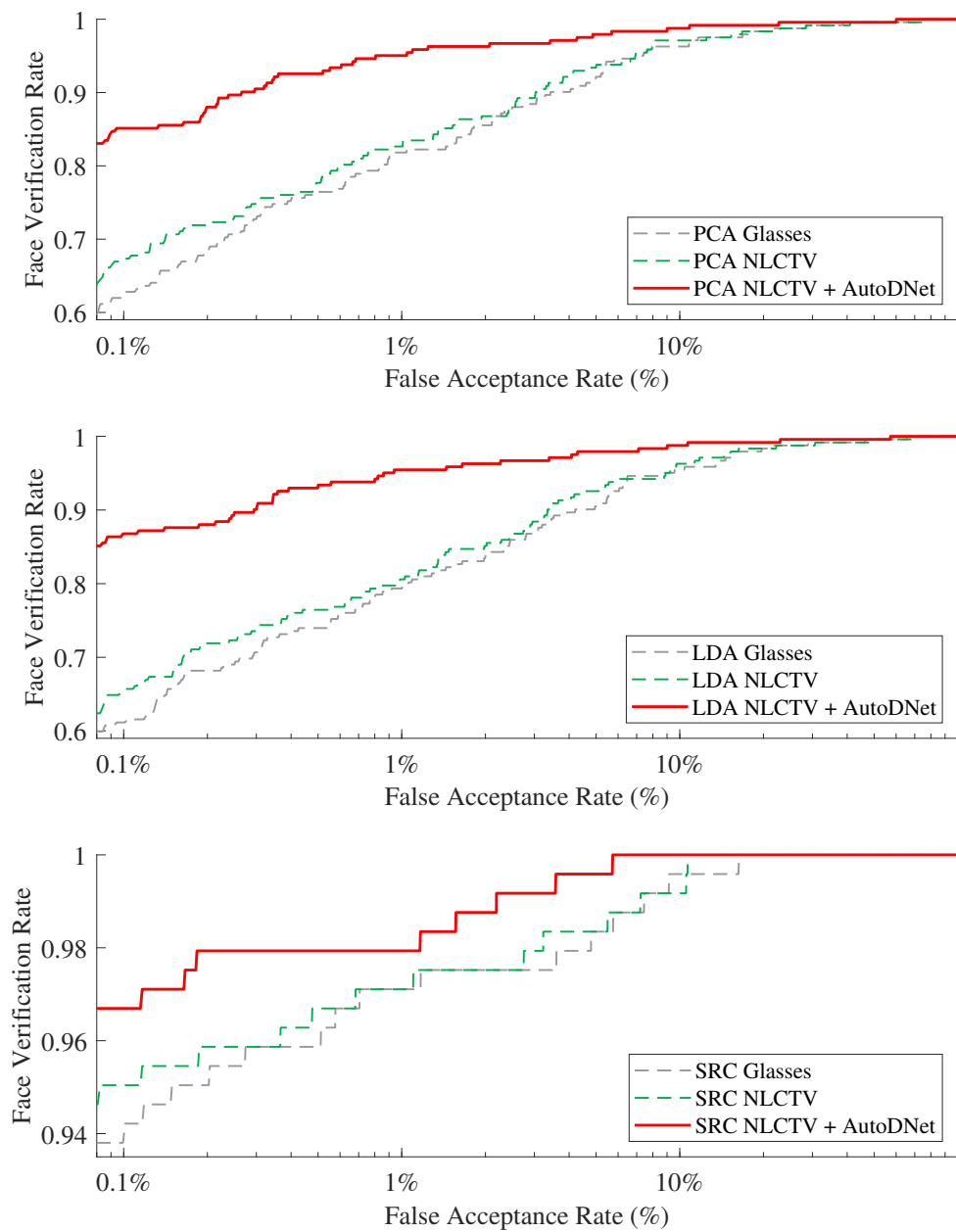


Figure 4.6: ROC curves on thin glasses with PCA, LDA, SRC respectively.

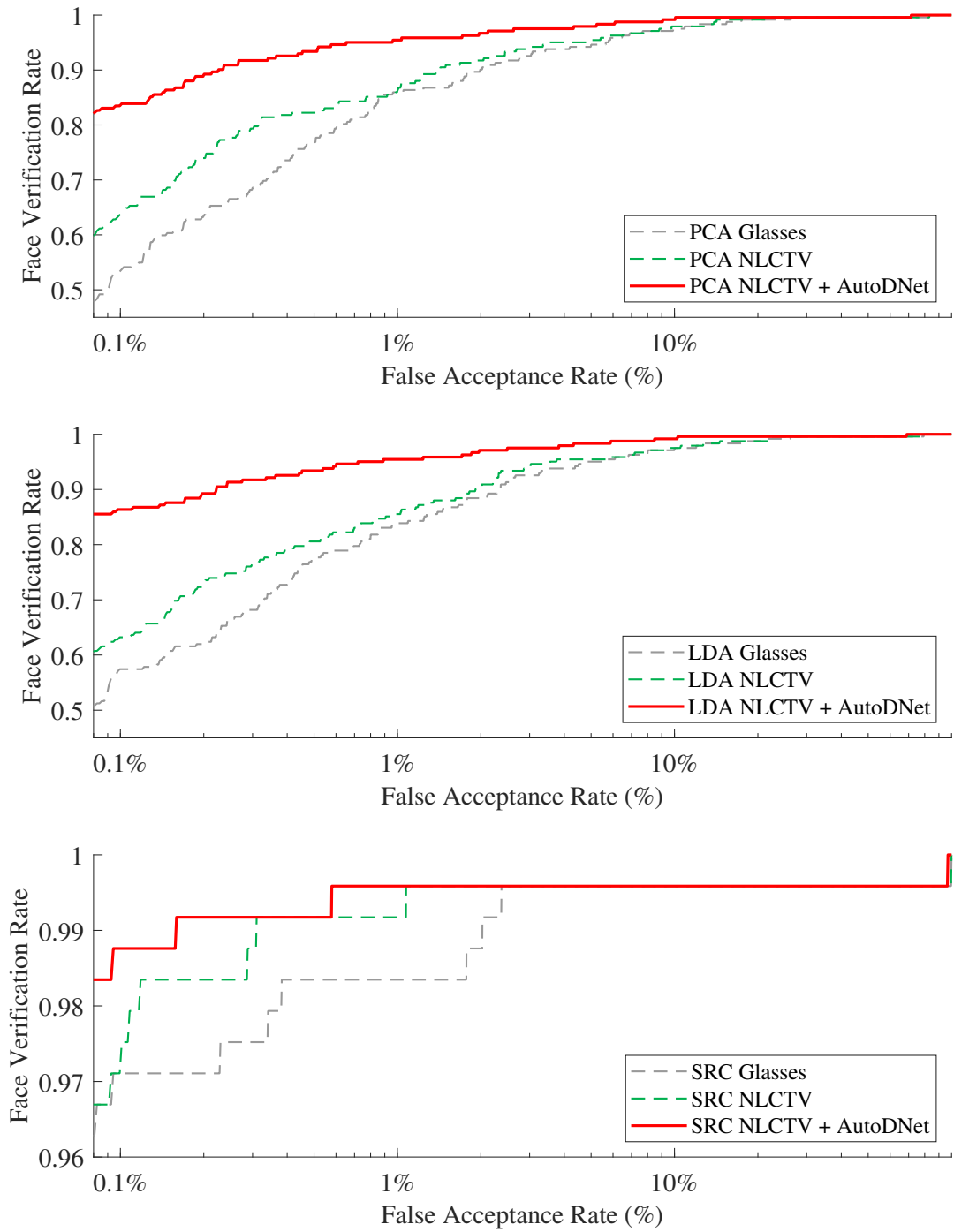


Figure 4.7: ROC curves on thick glasses with PCA, LDA, SRC respectively.

4.3.1.3 Face Verification

The next experiment is on the accuracy of face verification. It is different from facial recognition where a query face is compared to a set of gallery images and choose the one with the highest matching score. On contrary, face verification is a one-to-one face matching on which the decision is made based on a threshold of the score. The Receiver Operating Characteristic (ROC) curve is generated on various thresholds. Choosing the best threshold is not a simple task due to the tradeoff between the true and false acceptance rate. It is widely accepted to only consider threshold with 0.1% False Acceptance Rate (FAR).

Firstly, a transformation function is learned with AutoDNet by using images of the same 98 training subjects and evaluated it on 242 testing subjects (with thin and thick glasses), same as in the previous experiment. However, the testing setup is now different due to face verification's one-to-one matching nature. Neutral face from each subject can be paired with the other 242 faces wearing glasses. This creates a single correct pair and 241 false pairs for each participant. In total, we have 242 true matches and $242 * 241 = 58,322$ false matches for each scenario.

The test was conducted in two scenarios: with thin and thick glasses. For each scenario, the classification of faces is performed with PCA, LDA, and SRC. Verification performance is compared between images of the original glasses-wearing faces, the inpainted faces (NLCTV) and the reconstructed faces (NLCTV + AutoDNet) images. The ROC curves are shown in Figure 4.6 and Figure 4.7. The verification rate at 0.1% False Acceptance Rate (FAR) is summarized in Table 4.1. As expected, the verification performance is significantly improved following the glasses removal process. The improvements are especially distinct for PCA and LDA.

Table 4.1: Face verification rate at 0.1% False Acceptance Rate (FAR) before and after glasses removal.

Classification	Thin Glasses			Thick Glasses		
	Glasses	NLCTV	NLCTV + AutoDNet	Glasses	NLCTV	NLCTV + AutoDNet
PCA	62.81	67.36	85.12	53.72	64.05	83.88
LDA	61.16	65.70	86.78	57.44	63.22	86.36
SRC	94.21	95.04	96.69	97.11	97.52	98.76

4.3.2 Civil Engineering Application - Structure Health Monitoring

Vibration-based structural identification has been popularly used to detect the possible damage in structures and there have been some attempts based on Artificial Neural Networks (ANN) as discussed in Section 3.4.2. It is difficult to optimize the weights in the ANN networks that have

multiple hidden layers due to the vanishing gradient problem discussed in Section 2.3. In deep learning, a carefully designed pre-training strategy as used in AutoNet (Section 3.4.2.1) can avoid such problems efficiently. However, the basic framework proposed in the previous chapter will fail when varying degree of noises are present in the input. This is due to the increased complexities involved in the problem domain. Specially in SHM, when real data (not simulated) from experimental models are in concern, AutoNet performs poor due to the effects of overfitting.

The proposed extended framework have many layers in the dimensionality reduction component and relationship learning components with due regularization strategies as discussed in Section 4.2. This helps to address the additional complexities introduced by the noise in the data acquired in both the numerical and experimental studies performed. Numerical studies on a steel frame structure are conducted to investigate the accuracy and robustness of using the proposed extended framework for structural damage identification, particularly considering the effects of noise in the measurement data and uncertainties in the finite element modeling. Experimental studies on a prestressed concrete bridge in the laboratory are conducted to further validate the performance of using the proposed extended framework for structural damage identification. To evaluate the effectiveness of enforcing sparsity on the proposed extended framework, two variants named as **AutoDNet** and **SAF** are introduced for the experiments performed below. AutoDNet is the proposed extended framework with weight decay constrain while SAF denotes the proposed extended framework with both weight decay and sparsity constrains.

4.3.2.1 Extended Framework For Structure Health Monitoring

The same frequencies and mode shapes as described in Section 3.4.2.1, are used as the input \mathbf{c}^r while the structural elemental stiffness parameters are used as the output to the proposed extended framework. In addition, the input \mathbf{c}^r is pre-processed with data whitening to un-correlate and sphere (standard deviation in each dimension = 1) the dataset based on PCA to reduce redundancy and standardize the data to perform the training, validation and testing efficiently. The whiten data is used in dimensionality reduction component in the proposed extended framework as the input as mentioned in Section 4.3.2.3.

The data generation process for the numerical study based on a 7-storey reference model is explained in Section 2.8.1.2. To investigate the effectiveness and robustness of using the proposed extended framework for structural damage identification, the measurement noise in the datasets and the uncertainty effect in the finite element modeling are considered. The following scenarios are considered in numerical studies, namely,

1. Scenario 1: No measurement noise and modeling uncertainty, i.e. no noise effect in the

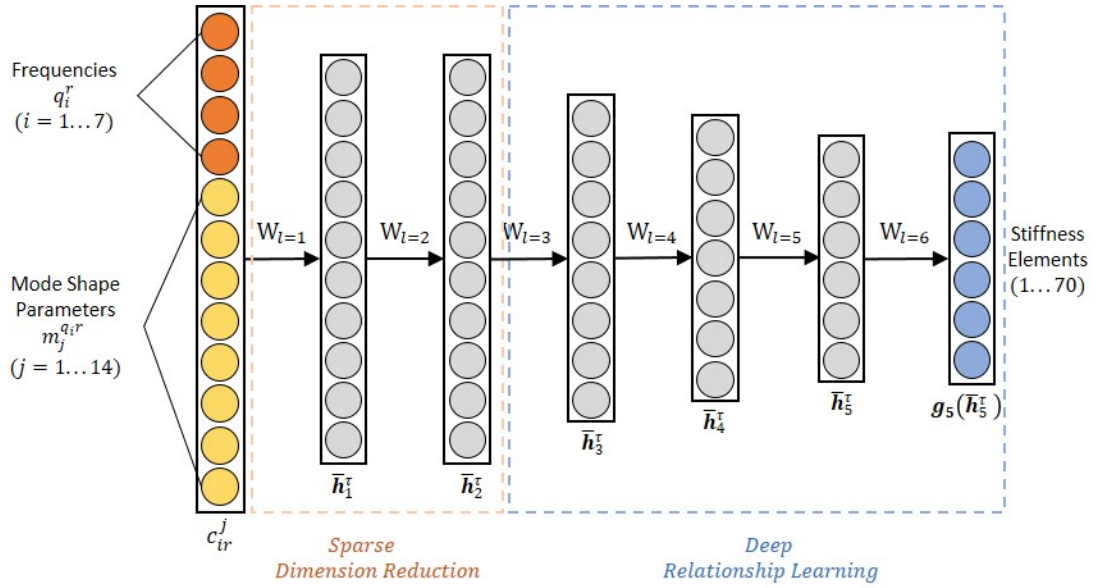


Figure 4.8: Proposed extended autoencoder based framework.

vibration characteristics and uncertainties in the finite element modeling are considered;

2. Scenario 2: Measurement noise effect considered. White noises are added on the input vectors, i.e. 1% noise in the frequencies and 5% in the mode shapes, considering structural frequencies are usually identified more accurately than mode shapes Xia *et al.* (2002);
3. Scenario 3: Uncertainty effect. 1% uncertainty is considered in the elemental stiffness parameters to simulate the finite element modeling errors;
4. Scenario 4: Both the above measurement noise and uncertainty effect simulated in Scenarios 2 and 3 are considered.

The output of the framework is the structural elemental stiffness parameters that explain both the location and the magnitude of the damages as seen in the previous chapter. The layer-wise pre-training procedure is performed as shown in the Figure 4.9 and once the optimal parameters are observed in the pre-training stage, both the dimensionality reduction and relationship learning components are fine-tuned (Figure 4.8) together to optimize all layers jointly as described in Section 3.3.3. The proposed extended framework is expected to perform non-linear dimension reduction preserving the necessary information from a noisy input to facilitate the relationship learning task from input to output.

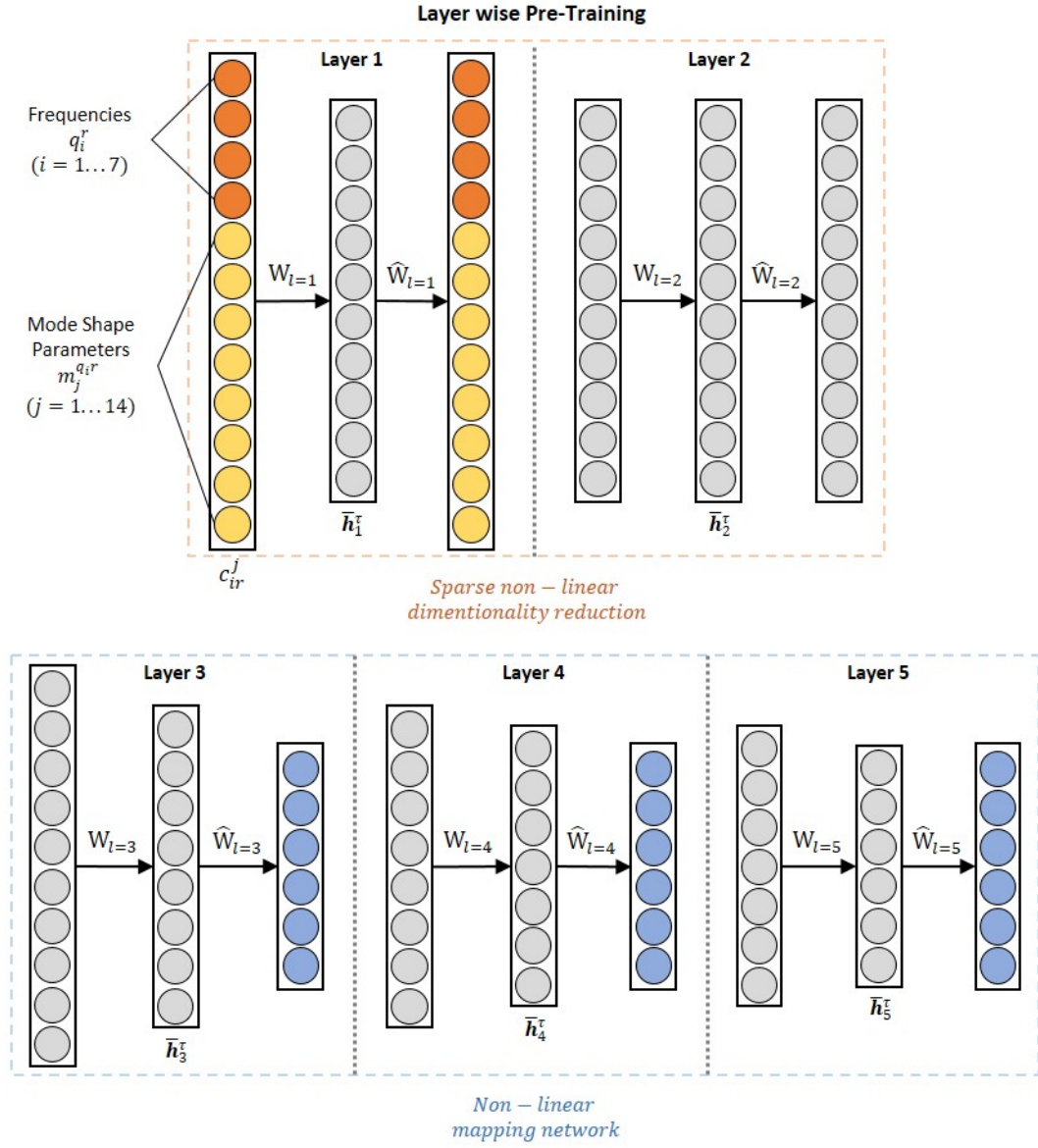


Figure 4.9: Pretraining on proposed extended autoencoder based framework.

4.3.2.2 Data Pre-Processing

Concatenated feature \mathbf{c}'' includes both the frequencies and mode shapes that are measured in different scales, therefore it is necessary to perform a normalization process before feeding them into the network. Data whitening described in Section 2.2.2 followed by a feature scaling phase (scaling to a value between 0 and 1) is performed. Such pre-processing establishes a common ground for all features to be treated equally in the learning process. The data range 0 to 1 is chosen since the operating range of the activation function "ReLU" falls into 0 and 1. Furthermore, the initial weights of the neural networks before pre-training are chosen randomly in a way that the input to the "ReLU" hidden unit will lie in the positive range. Considering structural damages are usually observed at a few numbers of elements, the sparse output vector is defined by defining 0 as the intact state and 1 as the full damage state. The output is also scaled to the range from -1 to +1 to serve the operating range of the used linear activation function in the final output layer. The performance evaluation of using the proposed extended framework for structural damage identification based on the pre-processed datasets will be described in the following section.

4.3.2.3 Evaluation of The Proposed Extended Framework

Performance evaluation of both the AutoDNet framework and SAF (Section 4.3) is conducted with four different scenarios as described in Section 4.3.2.1. We expand the proposed AutoNet framework in Chapter 3 to **AutoDNet** by choosing ($k = 2$) number of hidden layers with each comprised of 100 hidden nodes in the dimensionality reduction component, and ($m = 1$) hidden layer with 80 neurons used in the relationship learning component. In addition, we utilize weight decay and other regularization strategies to enhance the performance of the framework. In order to perform a fair comparison with AutoDNet method, a sparse configuration called **SAF-0** with the same structure as used in AutoDNet method is also formed. SAF utilizes sparsity as mentioned in Section 4.2.3 thus comparable to AutoDNet to evaluate the effects of sparsity. Lastly, we denote the best configuration for the proposed extended framework named as **SAF**, which is, $k = 2$ hidden layers with each comprised of 100 hidden nodes in the dimensionality reduction component, and $m = 3$ hidden layers with each comprised of 90, 80 and 70 neurons defined respectively in the relationship learning component.

The number of entries in the original input vector includes 7 frequencies and 14 x 7 mode shape values, that is, 105 in total. 70 elemental stiffness parameters are included in the final output vector. The complexity of the model is relatively high considering that a large number of learnable weights due to the depth of the network. The nature of the problem will be considered to determine the number of hidden layers and hidden units of the proposed extended framework. ReLU and linear

function are employed in the sparse dimensionality reduction component, while tanh and linear functions are employed in the relationship learning component in the pre-training scheme. After the pre-training, the same configuration is preserved for the hidden layers in order to fine-tune the whole network. Training, validation, and testing datasets are set to have 70%, 15% and 15% of the pre-processed datasets, respectively. In order to evaluate the performance of using the proposed extended framework for structural damage identification, MSE and Regression Value (R-Value: The coefficient of multiple determination for multiple regression) are employed to show the quality in the network training. All the numerical computations are conducted by using a desktop with an Intel i7 processor, 16GB RAM and the graphics card NVidia 1080 Ti GTX by using GPU for parallel computing.

4.3.2.4 Scenario 1: No measurement noise and modeling uncertainties

In this scenario, the datasets without measurement noise and uncertainty effect are used. The performance of using AutoDNet, SAF-0 and SAF are compared by examining the MSE values and R-Values on the testing datasets. The performance evaluation results are shown in Table 4.2.

Table 4.2: Performance evaluation results for Scenario 1 in the numerical study.

Methods	MSE	R-Value
AutoDNet	2.5e-04	0.921
SAF-0	8.27e-05	0.975
SAF	2.9e-05	0.993

The complexity of this structural damage identification problem is relatively high with 70 structural elemental stiffness parameters included in the output vector. AutoDNet may not be able to achieve a very good accuracy with several hidden layers without inducing the sparsity constraint. R-Value obtained from AutoDNet is 0.921 while SAF-0 is 0.975, as shown in Table 4.2. This shows the effectiveness and improvement of the sparse dimensionality reduction process. The improvement in R-value obtained from SAF-0 and SAF shows the effectiveness of utilizing more relationship learning layers and deeper neural networks as described in Section 4.2.2. The results from SAF show the improvement in the network prediction compared with AutoDNet and SAF-0 with a smaller MSE value and a better R-Value close to 1. To further demonstrate the performance of using SAF for structural damage identification, Figures 4.10-4.12 show the identification results of several single damages and multiple damage cases randomly selected from the testing datasets.

The damage identification results of a single damage case obtained from AutoDNet and SAF are shown in Figure 4.10. It can be observed that SAF provides more accurate damage identification

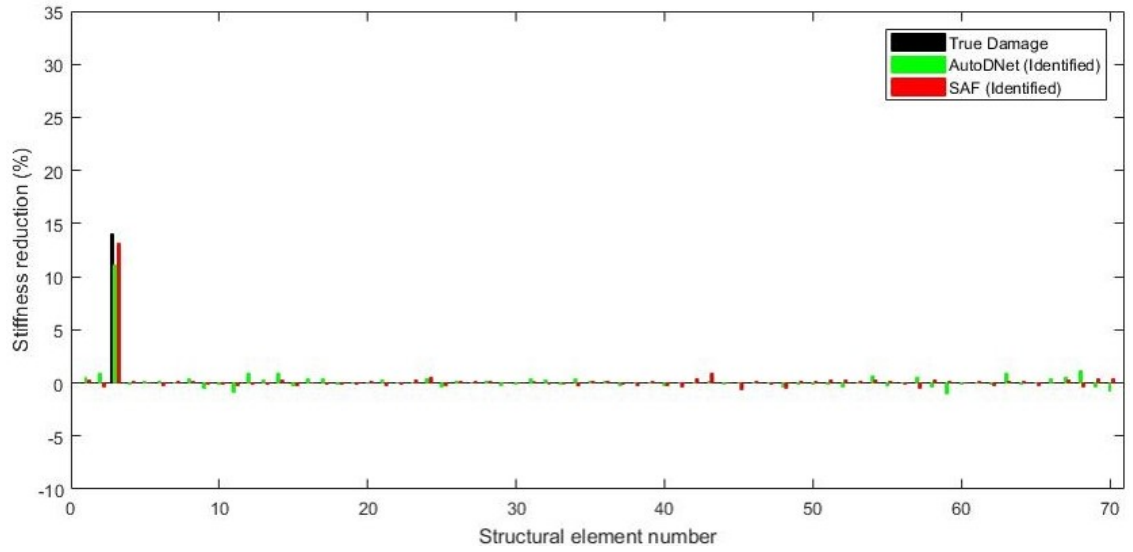


Figure 4.10: Damage identification results of a single damage case from AutoDNet and SAF for Scenario 1.

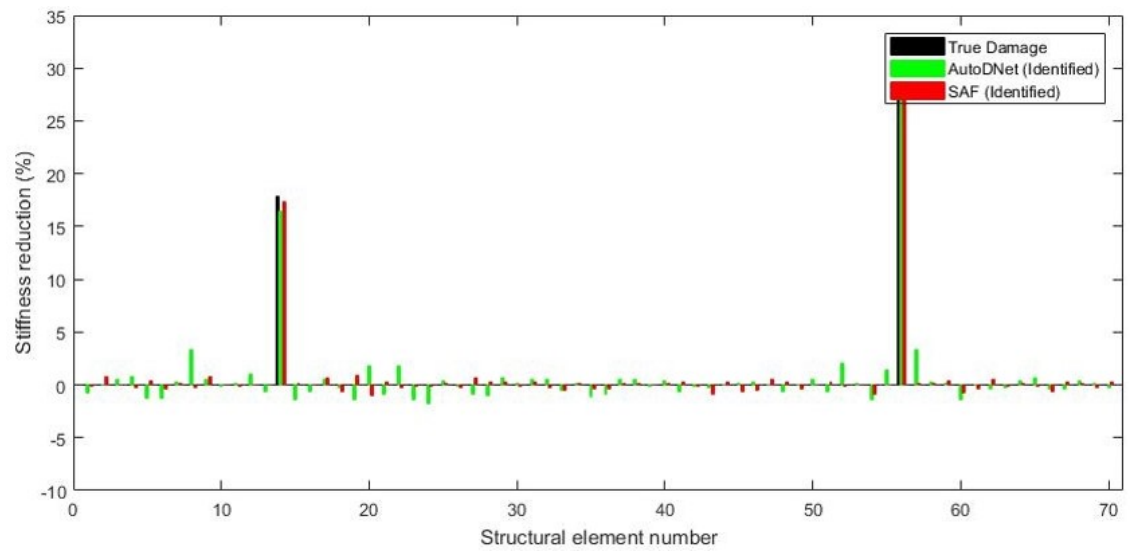


Figure 4.11: Damage identification results of a multiple damage case from AutoDNet and SAF for Scenario 1.

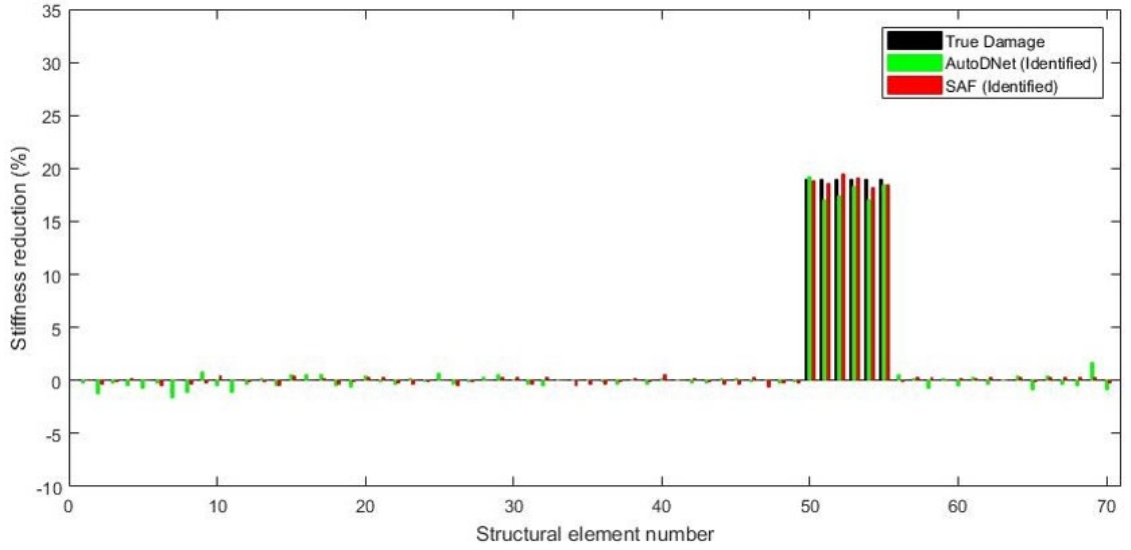


Figure 4.12: Damage identification results of a continuously distributed multiple damage case from AutoDNet and SAF for Scenario 1.

results than AutoDNet with less minor false positives and negatives. The damage location is well identified, and the identified stiffness reduction in the damaged element is very close to the actual value of SAF. The identified stiffness reduction values at the non-damage elements are very close to zero. SAF is also evaluated against AutoDNet with multiple structural damage cases, and the identification results of two multiple damage cases randomly selected from the testing datasets are shown in Figures 4.11 and 4.12. It can be seen clearly that SAF works very well for the identification of different types of multiple damage cases, e.g. two separate damages as shown in Figure 4.11 and continuously distributed multiple damages as shown in Figure 4.12. Damage locations are accurately detected, and the identified stiffness reductions are very close to the actual values with very small false identifications by using SAF.

4.3.2.5 Scenario 2: Measurement noise effect

In this scenario, the performance evaluation of SAF is conducted when the noise effect in the measurement data is considered. It is noted that 1% noise is included in the frequencies and 5% in the mode shapes. As shown in Table 4.3, a significant performance improvement in R-value is observed by using SAF compared with AutoDNet. It shows the robustness and effectiveness of using the sparsely learned features from the dimensionality reduction component for relationship learning when the noise effect is included in the measurements.

To further demonstrate the quality of the prediction in terms of both magnitudes and the loca-

Table 4.3: Performance evaluation results for Scenario 2 in the numerical study.

Methods	MSE	R-Value
AutoDNet	3.7e-04	0.794
SAF-0	2.7e-04	0.858
SAF	2.3e-04	0.886

tions in the damage identification, two multiple damage identification results are shown in Figures 4.13 and 4.14, respectively. Multiple structural damage identification is challenging and needs more precision on the identification of accurate stiffness reductions at the exact stiffness elements compared to the single damage cases, especially under the significant noise effect. It is observed that damage locations are accurately identified and the identified stiffness reductions are also very close to the actual values with very small false identifications. A better accuracy is achieved with SAF than AutoDNet regarding identifying the damage magnitudes, even for the minor damage cases as shown in Figure 4.14.

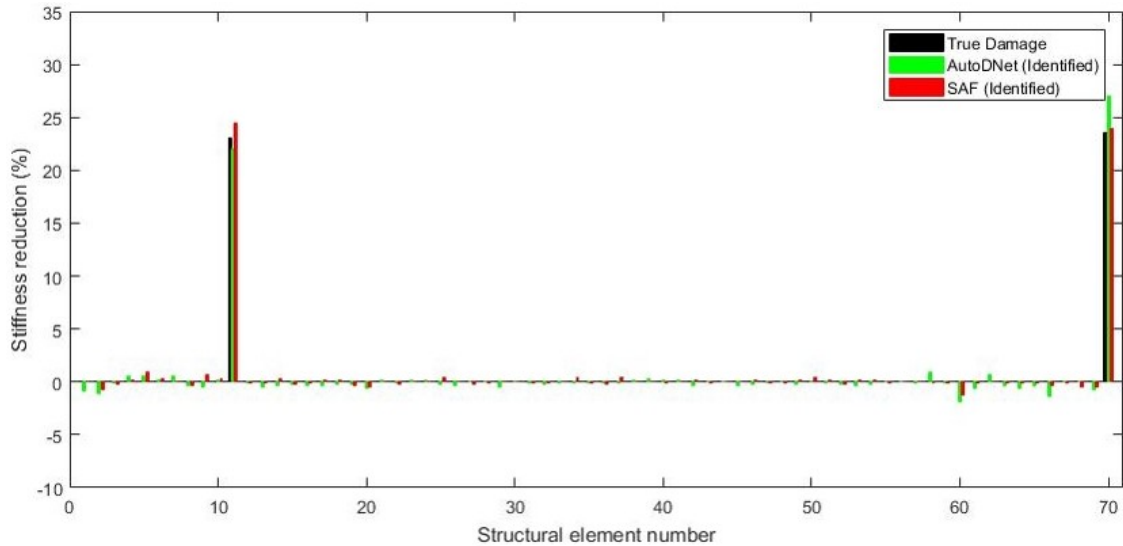


Figure 4.13: Damage identification results of a multiple damage case from AutoDNet and SAF for Scenario 2.

4.3.2.6 Scenario 3: Uncertainty effect

In this study, an accurate finite element model is required to generate the training data. When a model-based method is performed for structural damage identification, the inevitable uncertainties existed in the finite element modeling will significantly affect the accuracy and performance of structural damage identification. In this scenario, the uncertainty in the finite element model-

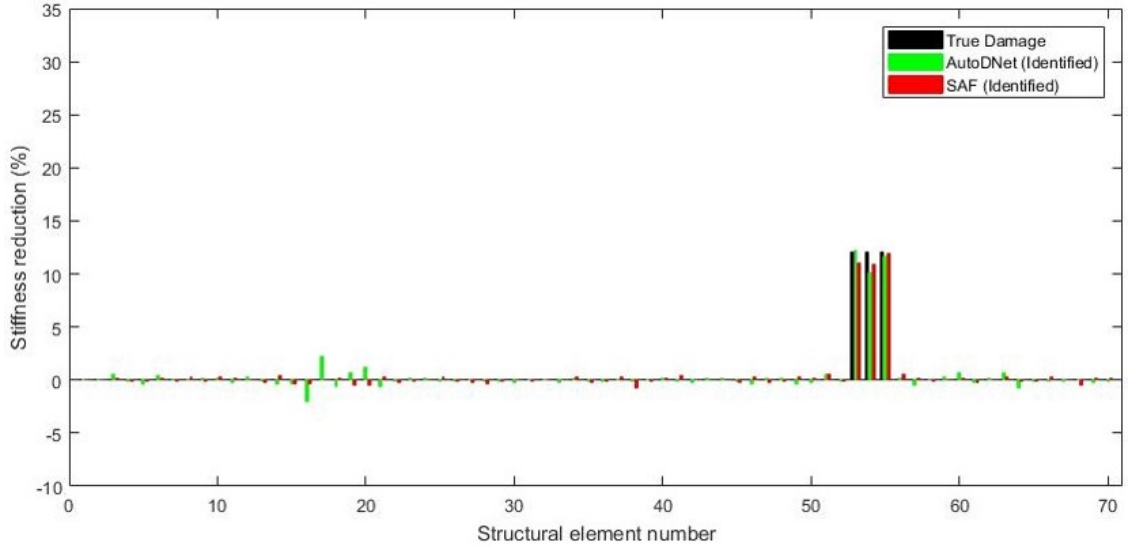


Figure 4.14: Damage identification results of another multiple damage case from AutoDNet and SAF for Scenario 2.

ing, i.e. 1% uncertainty in the elemental stiffness parameters are considered to simulate the finite element modeling errors. The performance evaluation results are shown in Table 4.4. SAF outperforms AutoDNet with an improvement in both MSE and R-value. The results from AutoDNet is affected by the uncertainty effect, as reflected by the corresponding lower R-value.

Table 4.4: Performance evaluation results for Scenario 3 in the numerical study.

Methods	MSE	R-Value
AutoDNet	2.9e-04	0.83
SAF-0	5.2e-05	0.975
SAF	2.9e-05	0.986

To demonstrate the robustness of SAF to the uncertainty effect, damage identification results of a single and a multiple damage cases randomly selected from the testing datasets are shown in Figures 4.15 and 4.16, respectively. As shown in both cases, the identified stiffness reductions are very close to the actual values with very small false identifications, even when the finite element modeling errors are considered. By comparing these identification results with those from AutoDNet, the accuracy and robustness of using SAF for structural damage identification with uncertainty effect are demonstrated. SAF gives more accurate damage identification results.

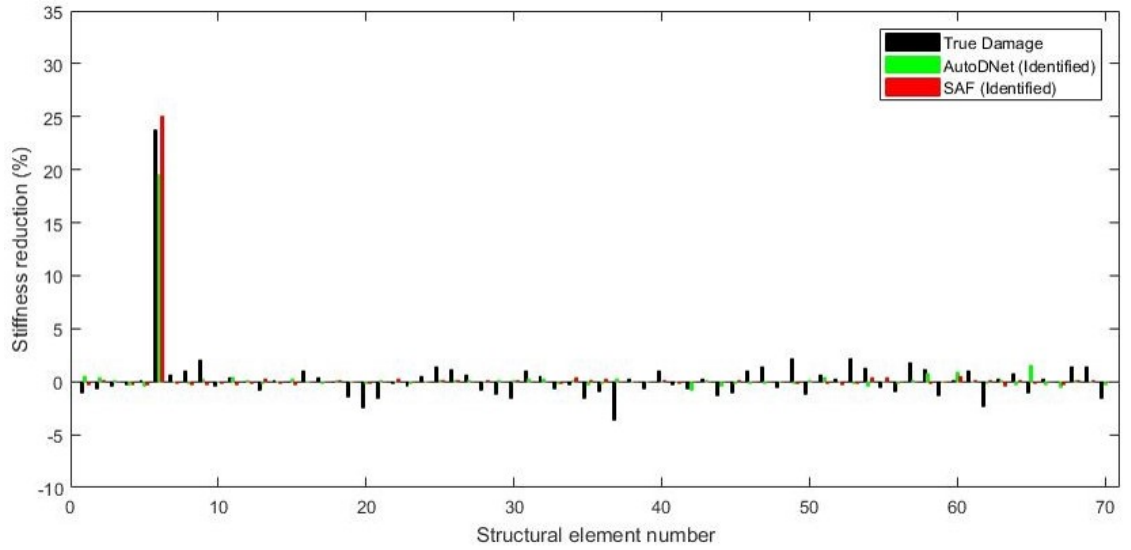


Figure 4.15: Damage identification results of a single damage case from AutoDNet and SAF for Scenario 3.

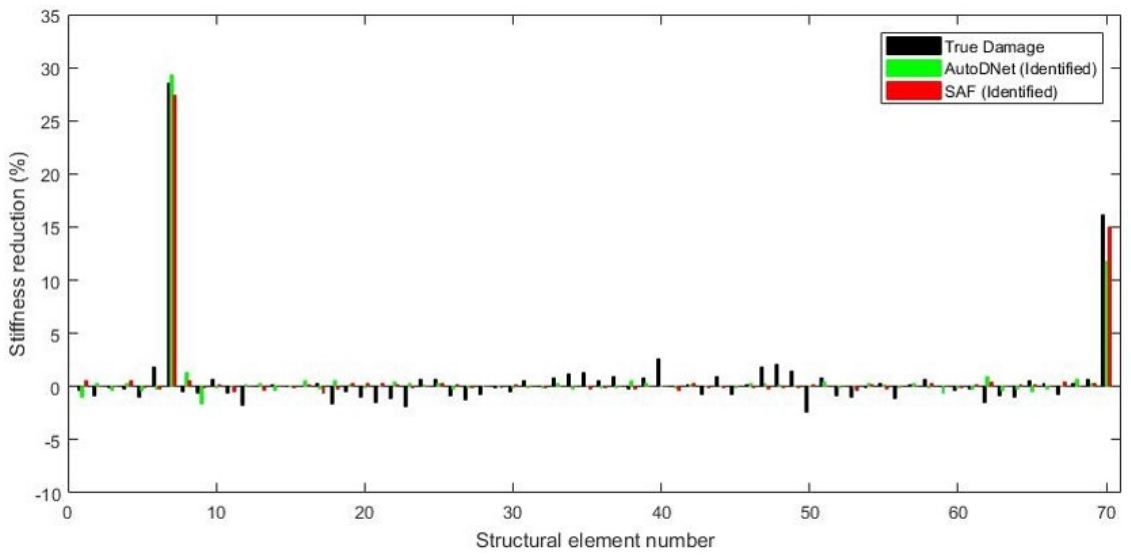


Figure 4.16: Damage identification results of a multiple damage case from AutoDNet and SAF for Scenario 3.

4.3.2.7 Scenario 4: Both the measurement noise and uncertainty

Both the measurement noise and uncertainty effect in Scenarios 2 and 3 are considered in this study to further investigate the performance and effectiveness of SAF. It is very challenging to achieve an effective and reliable structural damage identification when both measurement noise and uncertainty effect are involved. Such uncertainties may adversely affect the damage detection results. The performance evaluation results for this scenario are shown in Table 4.5.

Table 4.5: Performance evaluation results for Scenario 4 in the numerical study.

Methods	MSE	R-Value
AutoDNet	3.6e-04	0.732
SAF-0	3.3e-04	0.763
SAF	3.2e-04	0.792

As observed in Table 4.5, SAF once again outperforms the AutoDNet when both the measurement noise and uncertainty effect are considered. This is evidenced by a higher R-Value and a lower MSE value. L2-weight decay along with the sparsity constraint applied on the cost function formulation in this study ensures that SAF has a less space to over-fit the training data, and therefore improve the accuracy and robustness to identify the structural damage. Figures 4.17 and 4.18 show the identification of a single damage and a multiple damage case, respectively. It can be observed that the AutoDNet method outputs several false identification around 3% while SAF gives very minor values at those locations. The identified damage severities by using SAF are closer to the true damage values. Regarding the multiple damage case, SAF also provides much more accurate stiffness reduction predictions than AutoDNet in terms of both the damage locations and severities. Damage identification results from the above four scenarios demonstrate clearly the accuracy and robustness of using SAF in structural damage identification, compared with the latest previous study based on AutoDNet, even when the measurement noise and uncertainty effect are considered. The improvement is also demonstrated, supported by the identification results from various scenarios defined in this numerical study.

4.3.2.8 Experimental Verification

Experimental verification on a laboratory reinforced concrete bridge model (Section 2.8.2.2 - T-Beam) by using SAF for damage identification will be conducted in this section. The experimental setup, modal testing, initial model updating, training data generation, structural design of the neural networks and damage identification results will be described in the following sections.

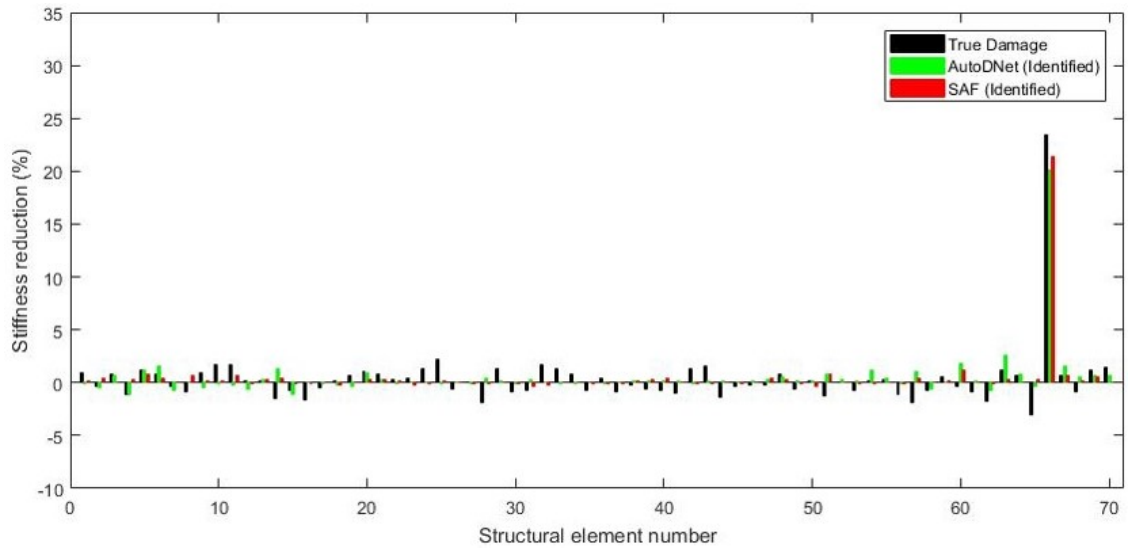


Figure 4.17: Damage identification results of a single damage case from AutoDNet and SAF for Scenario 4.

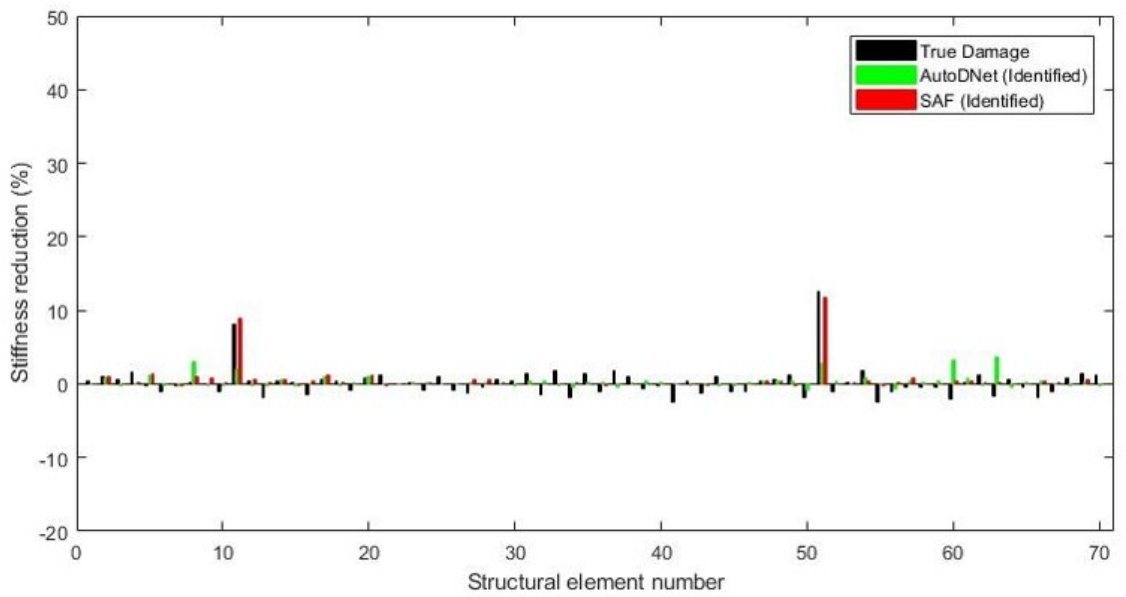


Figure 4.18: Damage identification results of a multiple damage case from AutoDNet and SAF for Scenario 4.

4.3.2.9 Network structure

Considering the complexity of the target problem and the number of parameters in the input and output vectors, SAF configuration is properly defined to have one hidden layer ($k = 1$) with 16 neurons in the dimension reduction component, and one layer ($m = 1$) with 16 neurons in the relationship learning component. The input vector includes 3 frequencies and 3 x 7 mode shape values, that is, 24 values in total. 16 stiffness reduction parameters are involved in the final output vector. ReLU and linear function are employed in the sparse autoencoders in the dimensionality reduction component, while tangent and linear functions are used in the relationship learning component in the pre-training scheme. After the pre-training, the same configuration is preserved for the hidden layers in order to fine-tune the whole network, based on the procedure as described in Section 3.3.3. To have a fair comparison, the same number of hidden layers and neurons are used to form an AutoDNet and the same training datasets are used for comparing the performance. In order to evaluate the quality of the damage predictions by using AutoDNet and SAF, MSE and R-Value are employed.

4.3.2.10 Training performance and damage identification results

Considering that the measurement data recorded in the laboratory inevitably include the noise effect, a robust deep neural network is required to accommodate this effect. To this end, the noise effect is included in the modal information of the original training data, e.g. 1% noise in the frequencies and 5% in the mode shapes. These datasets will be pre-processed with data whitening and used as the training data for SAF. Training, validation, and testing datasets are formed from the pre-processed datasets with percentages of 70%, 15% and 15% of data samples, respectively. The performance evaluation results for testing datasets by using AutoDNet and SAF are shown in Table 4.6. It can be observed that the MSE value from SAF is significantly smaller than that from AutoDNet. Besides, the regression from SAF is also improved, as observed in the R-Values.

To further investigate the performance of using real testing measurements for structural damage identification, the modal information obtained from the damaged state is used as the input to the trained SAF. Figure 4.19 shows the identified structural damage compared with those obtained from the AutoDNet and SAF. Since there is no analytical model relating the crack damage in a prestressed concrete beam with its flexural stiffness, it is not possible to calculate the analytical damage extents based on the observed cracks as shown in Figure 2.21. Therefore the identified damage pattern will be compared with the observed one. The main damages identified by SAF are located at the center of the bridge model, and this matches well with the observed cracks as shown in Figure 2.21. Significant false identification results are observed from the AutoDNet

method. There are 24 major cracks in total observed in the experimental tests, and these major cracks are mainly located in the six web elements from No.8 to No.13. It is shown in Figure 4.19 that the identified damage pattern from SAF has a good agreement with the experimental crack pattern observed in Figure 2.21. The identified damages from SAF are mainly distributed in web elements from No.8 to No.13. These results indicate that SAF can well identify the structural damages in the laboratory testing model with real measurement data.

Table 4.6: Performance evaluation results in the experimental study.

Methods	MSE	R-Value
AutoDNet	1.38e-05	0.983
SAF	4.57e-06	0.998

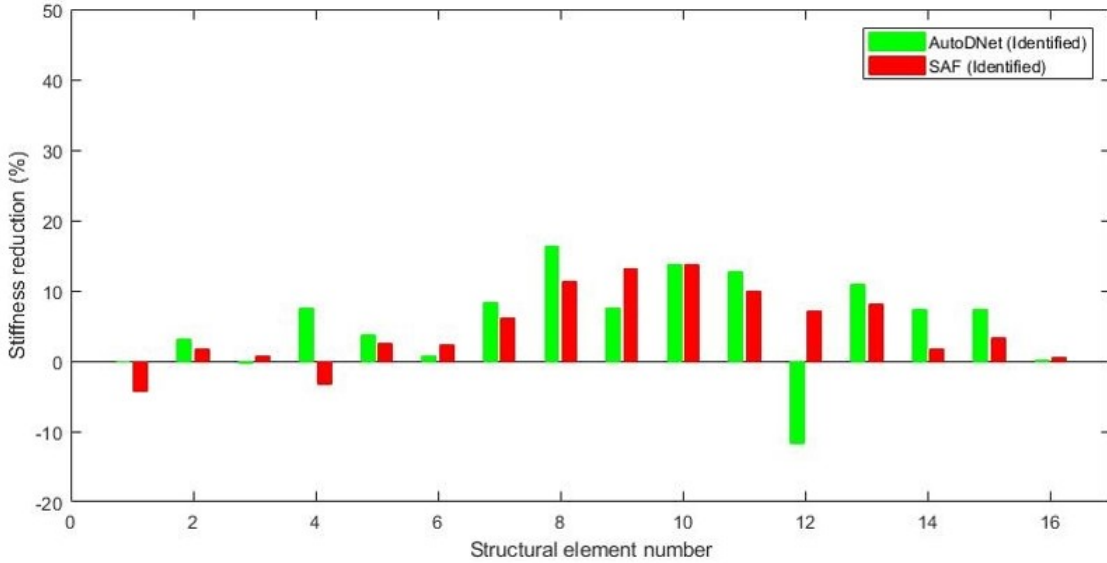


Figure 4.19: Damage identification results from AutoDNet and SAF in the experimental study.

4.4 Summary

In this chapter, we proposed an extended framework based on the AutoNet framework introduced in the previous chapter. The main focus is to build a robust framework towards handling various types of noise that exist in the data. The limitations of the previously proposed framework and the strategies to address such limitations were discussed in detail. AutoDNet and SAF are two variants introduced in this chapter to leverage the full potential of AutoNet framework mentioned previously. SAF is the sparsity addition to the non-sparse AutoDNet framework, thus could be effectively utilized in certain problem domains where sparsity is a key property.

In computer vision related application, the proposed AutoDNet framework was used as a component in a complete autonomous glasses removal system for face recognition and verification. The landmarks extracted via the system were utilized to localize glasses segments through a masking process for glasses removal where NLCTV (Duan *et al.*, 2015) inpainting followed by AutoDNet image reconstruction were performed. It could be seen as a double-layered filter to remove the presence of glasses. The experiment results revealed the high performance of the proposed AutoDNet framework on removing the presence of glasses on various face databases for both face recognition and verification thus confirming its effective noise (glasses) removal ability in computer vision domain.

In civil engineering related application, the proposed extended framework was utilized to enhance the performance in learning the relationship that exists in the modal information, such as frequencies and mode shapes, to structural stiffness parameters, specifically when noise is considered in both the input and output in the finite element modeling. The proposed extended framework is evaluated against four different cases. 1. there is no noise in the input (most trivial scenario); 2. the measurement noise effect is considered in the input; 3. the uncertainty is considered in the elemental stiffness parameters to simulate the finite element modeling errors; 4. both measurement noise and uncertainty effects are simulated. The more challenging arrangement of performing experiments on a prestressed concrete bridge in the laboratory is conducted to validate the performance of using the proposed framework for structural damage identification. The main components of the proposed extended framework namely, pre-processing, sparse dimensionality reduction, and relationship learning are exploited in depth. Compared to the AutoNet framework introduced in Chapter 3, the proposed SAF has three significantly different features which strongly improve the effectiveness and robustness of structural damage identification against the noise in the measurement data and uncertainty effect in the finite element modeling as shown below:

- Data whitening process is applied to un-correlate the data as a pre-processing procedure.
- Sparse regularization term is included in the training of the proposed framework to improve the performance in damage identification.
- A deep network structure can be defined with extended dimensionality reduction and relationship learning components to include more hidden layers and nodes without any hassle due to the regularization and sparsity constraints. The comparison results demonstrate the superiority of the proposed extended framework.

Despite the capability of the proposed SAF for robust feature learning against various types of noise, the framework cannot learn on the information that can be beneficial in a discriminant analysis, especially in a classification context. Since the primary focus of classification problems is

to recognize the identity, a strong discriminative feature projection is more desirable intuitively. A typical deep learning cost model learns a feature representation that is not necessarily discriminative itself due to the absence of discriminant information in the commonly used squared error function that is utilized to train the model in an unsupervised manner. These issues are investigated in depth in Chapter 5 along with the introduction of a deep discriminant analysis framework to perform a non-linear discriminant analysis.

Chapter 5

Non-linear Discriminant Analysis

5.1 Introduction

The predecessor AutoDNet framework introduced in Chapter 4 has been shown to be robust and efficient due to the optimization strategies utilized. It has been employed successfully in both computer vision and civil engineering related problem domains. However, the proposed AutoDNet framework is still prone to perform inconsistently in tackling with data that contains wide variations in data such as the extreme open mouth of a face image in the face recognition context. It will be even worse on extreme outliers that can exist in data, i.e., a corrupted image. The proposed AutoDNet framework cannot perform well under such conditions due to the non-beneficial properties of the squared error objective function which is typically used in unsupervised learning. If we force the framework to incorporate label information in feature learning, the deep latent space induced at the end would be highly discriminative for classification tasks irrespective of the large variations found in data. Since the primary focus in classification problems is to recognize the identity, a strong discriminative signal is favorable to guide the whole training process of the framework.

5.1.1 Why Discrimination?

Discriminant analysis is a technique that is utilized in machine learning research to analyze the data when the criterion or the dependent variable is categorical. The term categorical variable means that the variable can be divided into a number of categories. The discriminant analysis is often helpful to address questions like "are the groups different?", "on what variables, are the groups most different?", "can one predict which group an element belongs to using such variables?" etc. The objective of the discriminant analysis is to develop discriminant functions that are linear combinations of independent variables that can discriminate between the categories of the dependent variable. Also, it should be able to observe whether significant differences exist among the categories in terms of the predictor variables.

Discriminant analysis is explained by the number of categories that is possessed by the dependent variable. In many ways, discriminant analysis parallels multiple regression analysis (Multiple *et al.*, 2012). The main distinction between the two analysis methods is that discriminant analysis must have a discrete dependent variable while regression analysis deals with a continuous dependent variable. As in statistics Hastie *et al.* (1995), when the dependent variable has two categories, the type used is two-group discriminant analysis. If the dependent variable has three or more categories, the type used is multiple discriminant analysis. The major distinction between the types of discriminant analysis is that for a two-group analysis, it is possible to derive only one discriminant function. In the case of multiple discriminant analysis, many (more than one) discriminant functions can be computed.

The discriminant analysis aims at finding the most discriminant features of data, which can achieve maximum separation between classes. It has widely been employed for feature extraction and pattern classification in computer vision and pattern recognition. Pedagadi *et al.* (2013) presented a method with Fisher Discriminant Analysis (FDA) where discriminative subspace is formed from learned discriminative projecting directions, on which within between-class and inner-class distances are minimized and maximized respectively. They exploited graph Laplacian to preserve local data structure, known as LFDA. One most notable method in the discriminant analysis is the linear discriminant analysis (LDA) (Fisher, 1938; Rao, 1948; Duda *et al.*, 2000; Fukunaga, 2013). It is a statistical method that has been proven successful on classification problems, e.g., face recognition (Belhumeur *et al.*, 1997; Lu *et al.*, 2003; Yu and Yang, 2001) and text/document classification (Zifeng *et al.*, 2007; Torkkola, 2004). The objective of LDA is to find an optimal projection matrix W by maximizing the ratio of between-class scatters S_b to within-class scatter S_w as mentioned in Section 2.2.3. The equation 2.8 cannot work when S_w is singular. Unfortunately, many practical applications confront with high dimensional data, e.g., face recognition, which usually directly processes on face images and recognizes the face images as 2-D holistic patterns. A face image with a size of $m \times n$ pixels is represented by a feature vector in the mn -dimensional space. Therefore, the scatter matrix S_w in question is singular, because the dimension of data, in general, exceeds the number of training samples. This case is an intrinsic limitation of the classical LDA, i.e., the so-called small sample size or undersampled problem, which is also a common problem in the classification application (Tao *et al.*, 2006).

In the last decades, many methods have been proposed to tackle the undersampled problem that exists in LDA. Raudys and Duin (1998) used a pseudoinverse method by replacing the inversion of S_w with its pseudoinverse. In Friedman (1989), a matrix regularization technique is applied to make the scatter matrix S_w nonsingular; in fact, the regularized discriminant analysis (RDA) is a compromise between LDA and quadratic discriminant analysis. The penalized discriminant analysis is another version of RDA (Hastie *et al.*, 1995, 1994), where a small symmetric nonnegative penalty matrix is added to S_w to make it nonsingular.

The subspace discriminant analysis method is another alternative technique. In (Belhumeur *et al.*, 1997), the authors proposed a two-stage principal component analysis (PCA) + LDA (Fisherfaces), which applies PCA to reduce dimensionality such that S_w is nonsingular, followed by LDA for classification. However, one potential problem is that the PCA criterion may not be compatible with the LDA criterion; thus, some information is thrown away in the PCA step, which may contain the most discriminant information (Yu and Yang, 2001; Dai and Yuen, 2007). In (Liu *et al.*, 1992), the authors modified the traditional LDA criterion function by using the total scatter matrix $S_t (= S_w + S_b)$ as the divisor of the original criterion function instead of merely the within-class scatter matrix.

In Yu and Yang (2001), the direct LDA (DLDA) method is proposed to overcome the undersampled problem. First, the null space of S_b is removed, because there is no discriminant information, and then, it extracts the discriminant information that corresponds to the smallest eigenvalues of the within-class scatter matrix S_w . However, the smallest eigenvalues are very sensitive to noise (Jiang *et al.*, 2008); hence, one potential problem in DLDA is how the effect of noise can be prevented. In a.K. Qin *et al.* (2006), the authors proposed a generalized null-space uncorrelated Fisher discriminant analysis technique that integrates the uncorrelated discriminant analysis and weighted pairwise Fisher criterion (Loog *et al.*, 2001) for the small-sample-size problem. The author in (Ye and Li, 2005) proposed a two-stage LDA extension called LDA via QR decomposition (LDA/QR), which maximizes the separation between different classes by applying QR decomposition on scatter matrix S_b , followed by LDA to the "reduced" scatter matrices that result from the first stage. However, it has to discard the discriminant information that corresponds to eigenvalues of S_w that is equal to 0.

Nevertheless, many of these methods are based on heuristics and hand-engineering to generalized the feature extraction process for various kind of data but not capable of adapting depending on the data distribution.

5.2 Outliers in Discriminant Analysis

Discriminant analysis is quite sensitive to outliers and causes severe problems that even the robustness of discriminant analysis will not overcome. Thus learning robust and discriminative features of data has been a challenging task for real-world recognition systems. In the classical LDA based approaches (McLachlan, 2004; Zhao *et al.*, 1998), the Frobenius norm (L2-Norm) is applied to characterize the inter-class separability and intra-class compactness. Outliers may dominate the process of training since the intra-class, or inter-class distances are calculated by the sum of squared distances. Due to its sensitivity to outliers, the Frobenius norm is incompetent for

robust discriminant analysis. In Li *et al.* (2010), a rotation invariant L1-norm is used, and new linear discriminant analysis is proposed based on such norm. The L1-norm is calculated by using the sum of the absolute values without being squared. It is less sensitive to outliers than L2-norm. In fact, LDA/PCA based approaches assume that data is already centered, which is difficult to satisfy in practice especially when outliers occur (Subbarao and Meer, 2006). For example, outliers in face recognition can be caused by face noises with respect to large variations of pose, expression, lighting, occlusion, etc. All these factors affect the class means, and thereby the total data mean. These noises impose a challenging nature on the robustness of classification algorithms in varying degree. Previous research shows that correntropy is a useful tool for robust data analysis (Jeong *et al.*, 2009; Yuan and Hu, 2009) in information theoretic learning (ITL) (Principe *et al.*, 2000; Xu, 1999) and it can efficiently handle some non-Gaussian noises and large outliers (Pokharel *et al.*, 2009). This idea has been used with PCA (He *et al.*, 2011) with satisfactory performance. There has been an attempt to utilize MCC (Liu *et al.*, 2007) with LDA idea in (Zhou and Kamata, 2012). However the proposed LDA-MCC assumes that the class-specific data is already centered (fixed), and such an assumption is not realistic in many situations. Its performance improvement is also insufficient.

In this chapter, we present the way to establish efficient non-linear discriminant error criteria and the benefit of using it to guide the learning process of efficient high-level features in various face related problems. Then we show the applicability of those criteria in deep learning context to learn dynamic data-adaptive features directly from the raw pixels in contrast to the hand engineered features (LBP (Ahonen *et al.*, 2004), SIFT, SURF (Karami *et al.*, 2017)) which do not follow a learning process. These features are highly coherent and able to adequately characterize the higher order discriminant information for various problem domains such as face pose, expressions, hand-written digits, etc.

5.3 Proposed Error Criteria

In this section, the novel cost formulations for non-linear discriminant analysis and discriminant co-entropy is explained. Both of these cost formulations can be easily embedded into a deep network to directly produce discriminant feature representations in the resulting latent space while the novel discriminant co-entropy criterion has the advantage of being robust to outliers. In section 5.4, we perform non-linear discriminant analysis with a deep network (end-to-end learning) for effective feature learning for labeled data.

5.3.1 Non-Linear Discriminant Error Criterion

We propose a novel discriminant cost formulation in learning the non-linear discriminant subspace where the inter-class distance is maximized while minimizing the intra-class distance.

Assume that training data is given as $X = [x_1, x_2, \dots, x_N]$ where $x_j \in \mathbb{R}^d (j = 1, 2, \dots, N)$ represents the image of a person as a column vector and N is the total number of samples. Let N_i denote the number of samples in class $i (i = 1, 2, \dots, C)$ such that $N = \sum_i N_i$. We define the between-class cost C_b and within-class cost C_w as follows:

$$C_b(W) = \sum_{i=1}^C N_i \vartheta(\Phi(Wm_i) - \Phi(Wm)) \quad (5.1)$$

$$C_w(W, x_j) = \sum_{i=1}^C \sum_{x_j \in C_i} \vartheta(\Phi(Wx_j) - \Phi(Wm_i)) \quad (5.2)$$

where $m = (1/N) \sum_{i=1}^N x_i$ is the mean of the dataset, m_i represents the mean of the class i , $\Phi(\mathbf{x}) = \text{sigmoid}(\mathbf{x}) = 1/(1 + e^{-x})$ which is the non-linear squashing function and $\vartheta(\cdot)$ denotes a distance metric which in our case is the $L2$ norm. The objective of the non-linear discriminant analysis is to find a set of directions on a manifold in which the between-class distance is maximized while minimizing the within-class distance as:

$$C(W, x_j) = (C_w(W, x_j)/N) + \lambda(N/C_b(W)) \quad (5.3)$$

where λ is the regularizing parameter in the cost function between the within-class and between-class distances and it can be learnt via validation dataset. Our aim is to learn the projection W via optimization.

5.3.2 Discriminant Co-Entropy Criterion (DCC)

Recall the issues in LDA-MCC that are discussed in Section 5.2. We propose a novel discriminative cost formulation to overcome those issues. From the viewpoint of Information Theoretic Learning (ITL), the Discriminant Correntropy Criterion (DCC) proposed in this section can be considered as an extension of LDA with the following appealing advantages:

- The proposed DCC is derived with explicit interpretations and it is resilient to outliers as well as rotationally invariant.

- An efficient and robust algorithm based on the boosted gradient descent method is derived to solve the proposed DCC.
- Superiority in performance is confirmed after comparisons with the existing state of the art related algorithms.

The typical MCC cost formulation, the novel DCC formulation, and the boosted gradient descent method for efficient optimization of the proposed DCC cost formulation are described in details in the following sections.

5.3.2.1 Maximum Correntropy Criterion

The concept of correntropy (Liu *et al.*, 2007) which is derived from the generalized correlation function of random processes was proposed for Information Theoretic Learning (ITL). It is directly related to the Information Potential (IP) of Renyis quadratic entropy (Principe *et al.*, 2000) in which a Parzen windowing method is used to estimate the probability distribution of data (Santamaría *et al.*, 2006; Liu *et al.*, 2007). The correntropy is a local similarity measure between two arbitrary random variables A and B, defined by:

$$V_{\sigma}(A, B) = E[k_{\sigma}(A - B)] \quad (5.4)$$

where $E[.]$ denotes the mathematical expectation while k_{σ} is the kernel function that satisfies Mercers theory (Vapnik, 1995). The advantage of the kernel trick is that non-linear maps from the input space to a higher dimensional space is utilized in the above formulation. In contrast to traditional kernel methods, it works independently with pairwise samples. It has a clear theoretical foundation and is symmetric, positive, and bounded. It is often the case in practice that only a finite number of data are available and the joined probability density function of variables A and B is unknown. Thus the sample estimator of correntropy is defined as:

$$\hat{V}_{n,\sigma}(A, B) = \frac{1}{n} \sum_{i=1}^n k_{\sigma}(a_i - b_i) \quad (5.5)$$

where $k_\sigma(x) = \exp(-x^2/2\sigma^2)$ is the Gaussian kernel with the kernel size parameter σ . In Liu *et al.* (2007), the maximum of correntropy of error in Eq.5.5 is named as the maximum correntropy criterion (MCC). In contrast to the globally measured mean square error (MSE), MCC is local. The value of correntropy is mainly decided by the kernel function along the line $A = B$ (Liu *et al.*, 2007). Furthermore, the correntropy has a close relationship with m -estimators (Huber, 2011). The robust formulation of Welsch m -estimator (Liu *et al.*, 2007) for Eq.5.5 can be defined as $p_\sigma(x) = 1 - k_\sigma(x)$. The kernel size parameter of the correntropy criterion controls all the properties of the correntropy (Liu *et al.*, 2007). In Liu *et al.* (2007) a close relationship between the m -estimation and methods of ITL is established while a practical way is provided to choose an appropriate kernel size.

5.3.2.2 DCC Formulation

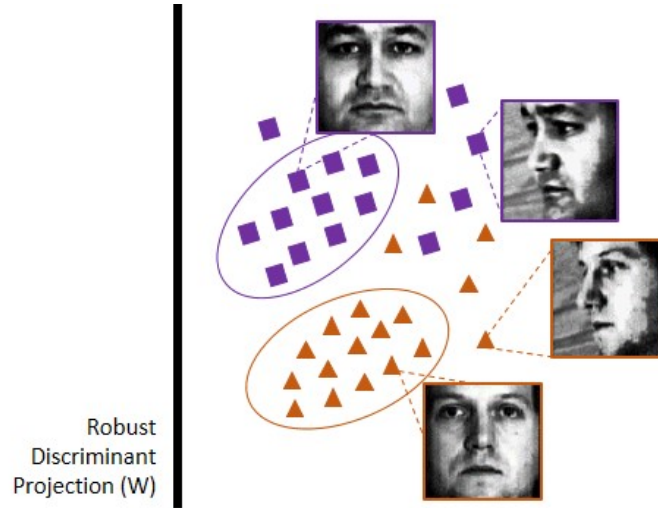


Figure 5.1: The set of outliers that can potentially affect the estimation of class means. Black line indicates the robust projection (W) that is learnt via DCC. Triangles and Squares indicate the samples that belong to two different classes.

We propose a novel discriminant correntropy criterion (DCC) to add an extra level of flexibility in learning a discriminant subspace. MCC is utilized in our approach, but the kernel size parameters are defined separately for each class in DCC to facilitate the outlier removal that may vary from one class to another. Hence the learned subspace will ideally be robust to large outliers that exist in different class samples as shown in Figure 5.1. Besides DCC preserves the class discriminatory information by ignoring the outliers as much as possible while reducing the feature dimensionality.

Assume that the training data is given as $X = [x_1, x_2, \dots, x_{N_i}]$ where $x_j \in \mathbb{R}^d (j = 1, 2, \dots, N_i)$ represents the image of a subject as a column vector and N_i is the total number of images for the

i^{th} class where $i = 1, 2, \dots, C$. We define the intra-class (within-class) cost J_w and inter-class (between-class) cost J_b respectively as follows:

$$J_w(X, W, \mu) = \frac{1}{C} \sum_{i=1}^C \left(1 - \frac{1}{N_i} \sum_{j=1}^{N_i} k_{\sigma_i}(\|W^T(x_j^i - \mu^i)\|_2) \right) \quad (5.6)$$

$$J_b(X, W, \mu) = \frac{1}{C} \sum_{i=1}^C \left(1 - k_{\sigma_C} \left(\frac{1}{\|W^T(\mu^i - \mu^0)\|_2} \right) \right) \quad (5.7)$$

$$\mu^0 = \frac{1}{C} \sum_{i=1}^C \left(\frac{1}{N_i} \sum_{j=1}^{N_i} x_j^i \right) \quad (5.8)$$

where μ^0 is the total mean of the dataset, μ^i represents the mean of class i , $k_{\sigma_i}(\cdot)$ denotes the kernel function for the i^{th} class with the kernel size parameter σ_i , $k_{\sigma_C}(\cdot)$ denotes the kernel function for the total dataset with the kernel size parameter σ_C . These kernel functions are utilized both for intra-class and inter-class cost calculations. For simplicity of notation, we can assume that $N_i = N, i \in C$. The matrix form of the above cost formulations can then be derived as:

$$J_w(X, W, U) = \frac{1}{C \times N} \left[C \times N - \text{sum}(g(\text{cnorm}(W^T(X - U))^2 \times \text{diag}(k_I))) \right] \quad (5.9)$$

$$J_b(X, W, U, U^0) = \frac{1}{C \times N} \left[C \times N - \text{sum}(g(\frac{1}{\text{cnorm}(W^T(U - U^0))^2} \times \text{diag}(k_C))) \right] \quad (5.10)$$

where

$$U^0 = [m_1 \ m_2 \ m_3 \ m_4 \dots m_i \dots m_{C \times N}],$$

$$m_i = m = \frac{1}{C \times N} \sum_{i=1}^{C \times N} x_i, x_i \in X \quad (5.11)$$

$$k_I = [\phi_1 \dots \phi_C], \ i \in C, \ \phi_i = \left[\frac{1}{\sigma_{i,1}^2} \dots \frac{1}{\sigma_{i,N}^2} \right] \quad (5.12)$$

$$k_C = [\psi_1 \dots \psi_{C \times N}], \psi_i = \left[\frac{1}{\sigma_c^2} \right] \quad (5.13)$$

$$g(x) = e^{-x/2} \quad (5.14)$$

$X \in \mathbb{R}^{d \times (C \times N)}$ is the complete data matrix, $U \in \mathbb{R}^{d \times (C \times N)}$ is the corresponding class means while $U^0 \in \mathbb{R}^{d \times (C \times N)}$ denotes the total mean of the dataset. Functions $cnorm(.)$ and $sum(.)$ calculate the column wise L2-norm and the summation of vector of elements respectively. For simplicity we follow the work in He *et al.* (2011) to initialize the kernel sizes as follows:

$$\sigma_{i,.}^2 = std(error_i) = std(\|W^T(x_{.}^i - \mu^i)\|_2^2) \quad (5.15)$$

$$\sigma_C^2 = std(error_C) = std(1/\|W^T(\mu^{\cdot} - \mu^0)\|_2^2) \quad (5.16)$$

Eq.5.15 denotes the kernel size for the i^{th} class while Eq.5.16 shows the kernel size for the total class means.

In summary, we can denote the final cost function as follows:

$$\begin{aligned} J(X, W, U, U^0) &= (1 - \lambda) \times J_w(W, U) + \lambda \times J_b(W, U, U^0) \\ s.t \ W^T W &= I \end{aligned} \quad (5.17)$$

where λ is a hyper-parameter to be learned against the validation dataset. In this formulation both the total data mean and each single class mean are variables and therefore can be adjusted to reduce the side effect when large outliers are present. Finally, the overall objective function can be written as:

$$\begin{aligned} [W^*, U^*] &= argmin_{W, U} J(X, W, U, U^0) \\ &= (1 - \lambda) J_w(X, W, U) + \lambda J_b(X, W, U, U^0) + \tau \|W^T W - I\|_2^2 \end{aligned} \quad (5.18)$$

where τ is the hyper parameter for the orthogonality constraint. We optimize W and U jointly through an iterative process (Algorithm 5.1). This optimization problem is similar to the non-linear discriminant analysis Section 5.3.1 where the optimization is performed with respect to W and the non-linear feature $\phi(X)$ observed via a deep net (Section 5.4). In contrary to the LDA-MCC method proposed in Zhou and Kamata (2012), DCC adjusts both the class means and

Algorithm 5.1 Training Algorithm:

```
1: Inputs :  $\ell U, \eta U, \delta U, t\delta U, gU$  (defined below)
2:  $\ell W, \eta W, \delta W, t\delta W, gW$  (defined below)
3:  $X = [x_1^1 \dots x_1^{N_1} x_2^1 \dots x_2^{N_2} \dots x_C^1 \dots x_C^{N_C}]$ 
4:  $U = \text{class means}$ 
5:  $W = W_0$ 
6: Outputs :  $W^*, U^*$ 
7: while (true) do
8:    $C_t = J(X, W^t, U^t, U^0)$ 
9:    $dU = \partial J / \partial U^t$ 
10:   $\delta U, t\delta U, gU = BGrad(\ell U, \eta U, \delta U, t\delta U, gU)$ 
11:   $U^{t+1} = U^t - \delta U$ 
12:  Calculate  $\{U^0\}^{t+1}$  with  $U^{t+1}$  according to Eq.5.11
13:   $CU^{t+1} = J(X, W^t, U^{t+1}, \{U^0\}^{t+1})$ 
14:  if ( $C - CU^{t+1} < \epsilon$ ) then
15:    break;
16:  end if
17:   $dW = \partial J / \partial W^t$ 
18:   $\delta W, t\delta W, gW = BGrad(\ell W, \eta W, \delta W, t\delta W, gW)$ 
19:   $W^{t+1} = W^t - \delta W$ 
20:   $CW^{t+1} = J(X, W^{t+1}, U^{t+1}, \{U^0\}^{t+1})$ 
21:  if ( $CU^{t+1} - CW^{t+1} < \epsilon$ ) then
22:    break;
23:  end if
24: end while
25: {where  $W_0$  is a random projection matrix,  $X$  is the dataset,  $t$  is the iteration index,  $\ell, \eta, \delta, t\delta, g$ 
denotes the learning rate, momentum rate, gradient, cumulated gradient, gradient gain respec-
tively. These parameters are defined for the mean data matrix ( $U$ ) and the projection matrix
( $W$ ) while  $\epsilon$  is a small positive value.  $BGrad(\cdot)$  denotes the boosted gradient descent function
as shown in Algo.5.2}
```

Algorithm 5.2 Boost Gradient (BGrad) Function:

```
1: Inputs :  $\ell, \eta, \delta, t\delta, g$ 
2: Outputs :  $\delta, t\delta, g$ 
3:  $\delta = \ell \times \delta$ ;
4: if ( $mr > 0$ ) then
5:    $same = (sign(\delta) == sign(t\delta));$ 
6:    $diff = (sign(\delta) != sign(t\delta));$ 
7:    $g = (g + .3) * same + (g * .7) * diff$ ;
8:    $t\delta = \eta * t\delta + (g * \delta)$ ;
9:    $\delta = t\delta$ ;
10: end if
11: {where  $\ell, \eta, \delta, t\delta, g$  denotes the learning rate, momentum rate, gradient, cumulated gradient,
    gradient gain in each function call.}
```

the total data mean during the optimization process. Furthermore, it utilizes class-specific kernel bandwidths to learn a more robust subspace for all classes.

5.3.2.3 DCC Optimization

In this section, we utilize an efficient implementation of the gradient descent algorithm on the cost functions to find the optimal parameters. It adjusts the rate of gain of the descent by a certain factor depending on the current and the previous directions of the gradient (Algorithm 5.2). The gradients of the cost functions denoted by Eq.5.9, Eq.5.10, Eq.5.18, and the orthogonality constraint can be derived as follows:

$$\frac{\partial J_w}{\partial U} = \frac{1}{C \times N} \left[WW^T (U - U^0) \text{diag}(mcc) \text{diag}(k_I) \right] \quad (5.19)$$

$$\frac{\partial J_b}{\partial U} = \frac{1}{C \times N} \left[WW^T (U - U^0) \text{diag}(frac) \text{diag}(k_C) \right] \quad (5.20)$$

$$\frac{\partial J}{\partial U} = (1 - \lambda) \frac{\partial J_w}{\partial U} + \lambda \frac{\partial J_b}{\partial U} \quad (5.21)$$

$$\frac{\partial J_w}{\partial W} = \frac{1}{C \times N} \left[(X - U) \text{diag}(mcc) \text{diag}(k_I) (X - U)^T W \right] \quad (5.22)$$

$$\frac{\partial J_b}{\partial W} = \frac{-1}{C \times N} \left[(U - U^0) \text{diag}(\text{frac}) \text{diag}(k_C) (U - U^0)^T W \right] \quad (5.23)$$

$$\frac{\partial J}{\partial W} = (1 - \lambda) \frac{\partial J_w}{\partial W} + \lambda \frac{\partial J_b}{\partial W} + \tau \frac{\partial \|W^T W - I\|_2^2}{\partial W} \quad (5.24)$$

where,

$$\frac{\partial \|W^T W - I\|_2^2}{\partial W} = 4(W^T W - I)W^T \quad (5.25)$$

$$mcc = g(\text{cnorm}(W^T (X - U))^2 \times \text{diag}(k_I)) \quad (5.26)$$

$$\text{frac} = \frac{g((1/\text{cnorm}(W^T (U - U^0))^2) \times \text{diag}(k_C))}{\text{cnorm}(W^T (U - U^0))^4} \quad (5.27)$$

There is no theoretical proof for the convergence of the proposed optimization process. However, we observed that irrespective of the initial conditions, the optimal solution for the above objective always performs better than the state of the art related methods. Thus its convergence can be justified by the evidence shown via the extensive experiments described in the Section 5.5.2.

5.4 Deep Discriminant Analysis (DDA) Framework

Deep Learning (DL) methods have shown impressive progress over conventional linear techniques due to the immense power of learning data adaptive features for problems that possess highly non-linear characteristics. Despite the improvements made by deep models, the parameter learning in deep models requires a high quantity of training data in the form of matching pairs while some machine learning problems are facing the small sample size problem (Chen *et al.*, 2000; Zhang *et al.*, 2016). Typically, only a few hundreds of training samples are available due to the difficulties of collecting matching pairs. Deep models, in general, utilize a loss function at the single sample level. The cross-entropy loss is a popular choice that can maximize the probability of each element. The gradients computed from the class-membership probabilities with cross-entropy may help to enlarge the inter-class differences but is unable to reduce the intra-class variations.

Futhermore, the structure of a dataset that can be described with the inter-class and intra-class

evaluation criteria was generally not incorporated into the objective of learning discriminative features. We are motivated to develop an architecture that is more suitable to perform discriminant analysis with a moderate number of data samples. At the same time, the proposed network is expected to produce feature representations which could be easily separable by a linear model in its latent space such that variation within the same class is minimized while differences between classes are maximized.

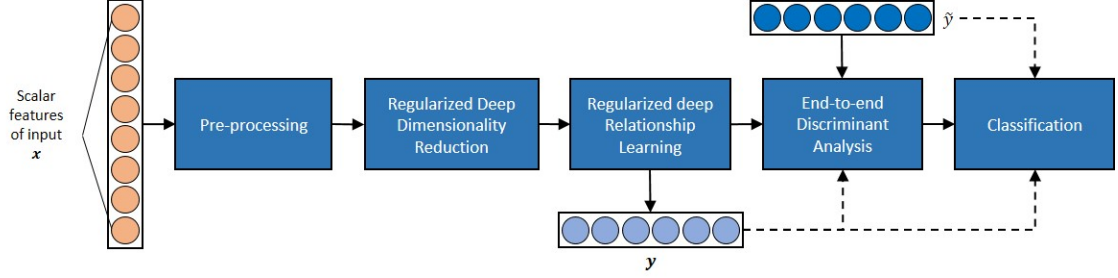


Figure 5.2: Highlevel view of the proposed discriminant framework.

In general, Autoencoder (AE) based models are relatively easy to train and can be used to subdivide the global non-linearity involved in a particular problem domain into a series of sub-objectives where each sub-objective is modeled via a simple AE. Hence it promotes a more straightforward design to reduce the complexity involved in CNN. We present an autoencoder based hybrid architecture (extension to the model introduced in chapter 4) for powerful feature learning with labeled datasets, which is comprised of dimensionality reduction layers and multiple supervised layers.

The novel non-linear discriminant error criterions help to perform effective feature learning from raw pixels and can be applied on almost all deep learning models that perform classification tasks (such as Zhu *et al.* (2014) and Kan *et al.* (2014)). We embed the non-linear discriminant error criterion discussed in Section 5.3.1 in a deep learning network to form the deep discriminant analysis framework. The non-linear error criterion Section 5.3.1 is chosen over the robust alternative formulation to ease the training process of the deep learning network. This is possible due to its simplicity of gradient calculations and compatibility with the backpropagation algorithm. The hierarchical mapping of features (dimensionality reduction component) in the proposed framework would decrease the effects of outliers and add robustness to a certain extent thus become useful in performing non-linear discriminant analysis with the novel cost formulation discussed in Section 5.3.1. Unlike many existing methods which assume the problem to be linear in nature, the proposed DDA framework makes no prior assumptions thus exploiting the full potential of learning a highly non-linear transformation. Furthermore, the network is trained with the gradients of the novel discriminant criterion to approximate inter-class and intra-class variations (without discarding any discriminant information in inter-class and intra-class as mentioned above) in the discriminant analysis component (Figure 5.2) and finds the projection Wc (Figure 5.3) to maximize the ratio between them in end-to-end fashion. It enforces the promotion of feature distributions which

have a low variance within the same class and high variance between classes. As a result, the computed deep non-linear features become linearly separable in the resulting latent space. The high-level representations learned via the proposed model are highly supervised and can help to boost the performance of subsequent classifiers such as LDA. We describe the beneficial properties of deep latent space induced by non-linear discriminant criterion in terms of low intra-class variance, high inter-class variance, and optimal decision boundaries, to gain plausible improvements for the tasks of classification. The high-level overview of the framework is shown in Figure 5.2. More importantly, our method requires no prior information about the data and can be trained with any database of interest in pre-training and fine-tuning stages.

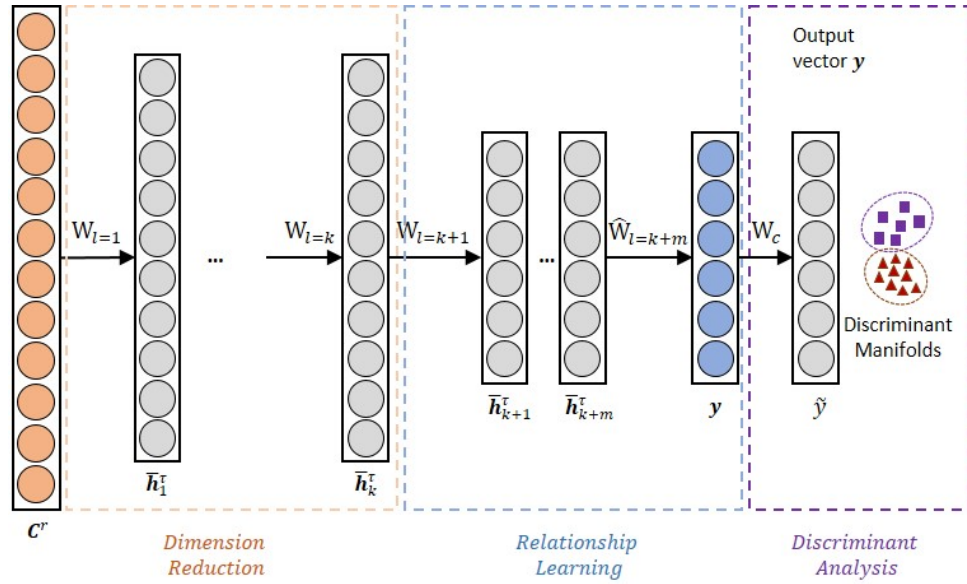


Figure 5.3: Architecture of the proposed discriminant framework.

The proposed DDA framework consists of three interconnected learning process: the progressive non-linear dimension reduction process, relationship learning process, and a discrimination process as shown in Figure 5.3.

As mentioned in the previous chapter, similar to AutoDNet framework, the layers in dimensionality reduction component of DDA framework perform non-linear dimension reduction while the layers in relationship learning component perform the mapping between the reduced dimensional feature and the output. Additionally in the proposed DDA framework, the layers in the discriminant analysis component as per the Figure 5.3, perform the discriminant analysis based on a single representative class instance thus ensures the features observed in the reconstruction layer are highly discriminative. The input \mathbf{c}_i^r of the proposed DDA framework is defined as follows:

$$\mathbf{c}_i^r = [q_{i,1}^r, \dots, q_{i,n}^r] \quad (5.28)$$

\mathbf{c}_i^r represents the combined high dimensional input vector where $q_{i,p}^r$ is the p^{th} ($p = 1 \dots n$) scalar feature of r^{th} sample that belongs to the i^{th} class.

Pre-training for layers in dimensionality reduction and relationship learning component is performed as mentioned in the Section 3.3. Finally, we utilize the DDA layer on the output layer to learn the initial projection matrix W_c during the pre-training process. The objective function of the layer in the discriminant analysis could be written as:

Algorithm 5.3 Training Algorithm:

- 1: $X^j = [c_{11}^j \dots c_{1N_1}^j c_{21}^j \dots c_{2N_2}^j \dots c_{S1}^j \dots c_{SN_S}^j]$;
 - 2: $V^j = \text{validation patch set}$;
 - 3: $\{W^j\}_0 = \text{weights obtained after pretraining stage}$;
 - 4: $\{ve^j\}_0 = +\infty$ //temporary variables;
 - 5: $count_{epoch} = 0$;
 - 6: $error_{validation} = +\infty$;
 - 7: **while** ($error_{validation} > 0$) AND ($count_{epoch} < t$) **do**
 - 8: $F^j = DDA_4(X^j, \{W^j\}_{idx})$
 - 9: $e^j = C(\{W_c^j\}_{idx}, F^j)$
 - 10: $\nabla\{W_c^j\}_{idx} = \frac{\partial e^j}{\partial\{W_c^j\}_{idx}}$
 - 11: $\nabla F^j = \frac{\partial e^j}{\partial F^j}$
 - 12: $\nabla\{W^j\}_{idx} = \nabla F^j \frac{\partial DDA_4(X^j, \{W^j\}_{idx})}{\partial\{W^j\}_{idx}}$
 - 13: $\{W_c^j\}_{idx+1} = \{W_c^j\}_{idx} - \ell (\nabla\{W_c^j\}_{idx})$
 - 14: $\{W^j\}_{idx+1} = \{W^j\}_{idx} - \ell (\nabla\{W^j\}_{idx})$
 - 15: $VF^j = DDA_4(V^j, \{W^j\}_{idx+1})$
 - 16: $\{ve^j\}_{idx+1} = C(\{W_c^j\}_{idx+1}, VF^j)$
 - 17: $error_{validation} = \{ve^j\}_{idx} - \{ve^j\}_{idx+1}$
 - 18: $count_{epoch} = count_{epoch} + 1$
 - 19: **end while**
 - 20: {where idx is the iteration index, ℓ denotes the learning rate, t is a large positive integer to denote the maximum epoch count and $DDA_4(\cdot)$ represents the features obtained at the reconstruction layer (Layer 4) and W_c is the projection matrix to the DDA space. In training, the breaking condition that mostly occurs is the validation error criterion.}
-

$$W_c^* = \underset{W_c}{\operatorname{argmin}} C(W_c, g_q(h_{i,q}^r)) \quad (5.29)$$

where $h_{i,q}^r$ is the learned representation of the last layer in the relationship learning component ($q = k + m + 1$ as mentioned in Section 4.2.2) for the r^{th} sample that belongs to the i^{th} class and $C(\cdot)$ is the cost formulation described in Eq. 5.3. In order to refine the features further, we perform training on the database of interest by stacking the pre-initialized layers one after another to jointly optimize the objective as below:

$$\begin{aligned} & \left[W_l^*|_{l=1}^L, b_l^*|_{l=1}^L, \widehat{W}_L^*, \widehat{b}_L^*, W_c^* \right] = \\ & \underset{W_l|_{l=1}^L, b_l|_{l=1}^L, \widehat{W}_L, \widehat{b}_L, W_c}{\operatorname{argmin}} C(W_c, p(c_i^r)) \end{aligned} \quad (5.30)$$

where $p(c_i^r) = g_L(f_L(\dots(f_1(c_i^r))))$ with $L = k+m$; and $W_l|_{l=1}^L$ denotes the encoders weights, \widehat{W}_L denotes the decoder weights and W_c is the projection matrix learned via optimizing the non-linear discriminant error criteria $C(\cdot)$. The training algorithm is shown in Algorithm 5.3.

5.5 Applications

The novel cost formulations discussed above are evaluated with an extensive set of experiments in the following sections. Section 5.5.1 evaluates the effectiveness of non-linear discriminant error criterion in face recognition domain while Section 5.5.2 demonstrate the performance of utilizing the discriminant co-entropy error criterion where outliers become a major issue in computer vision related problems.

The SHM problem in civil engineering domain mainly falls under a non-linear regression task where the inputs (mode shapes and frequencies) need to be regressed against the outputs (elemental stiffness reductions). It is important to predict the location as well as the magnitude of the stiffness reduction in the structure in SHM problem. Since the number of stiffness reductions patterns with varying magnitude could be infinite, it is impossible to define a class label for each pattern of stiffness reduction. The proposed discriminant analysis techniques and the outlier robust cost formulations are solely based on the class label information of the data, Hence the direct applicability of those methods on SHM problem is intractable. For the time being we exclude the experiments in SHM using the proposed DDA framework, but will continue to explore alternative ways to reorganize SHM data to utilize the proposed methods.

5.5.1 DDA Framework For Face Recognition

Although many face recognition methods were proven to achieve impressively good performance in the constrained environments, their performance is still unsatisfactory in unconstrained environments, mainly due to the large variations in face images caused by pose, expression, lighting, occlusion etc. In fact, a face with such variations can be regarded as a neutral face exposed to different kind of noises. De-noising such noises while extracting robust and discriminative features to enlarge the inter-personal variations and shrink the intra-personal margins at the same time remains one of the most challenging problems in face recognition.

We utilize the proposed DDA framework to perform discriminant learning from raw pixels to enhance the performance in face recognition in general. High-level representations learned via the proposed DDA framework are highly supervised and can help to boost the performance of subsequent classifiers such as LDA. The value of utilizing non-linear discriminant error criterion as a tractable objective to guide the learning process for efficient high-level features is demonstrated in various face related problems.

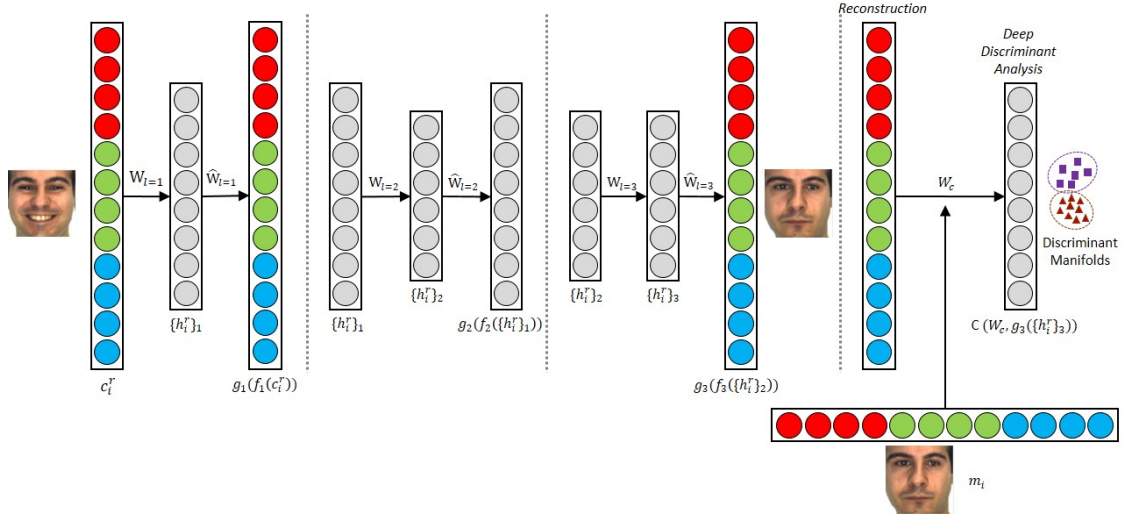


Figure 5.4: Progressive pre-training of respective layers to achieve better initial weights prior to the training phase. Layers 1 and 2 perform the progressive non-linear dimension reduction while de-noising happens at Layer 3. The non-linear discriminant criteria is utilized on the reconstructed neutral face (DDA layer). The mean face of class i is denoted by m_i .

The proposed DDA framework is configured with two hidden layers are chosen for dimensionality reduction of the input while the relationship learning component consists of one hidden layer for simplicity. A face with variations such pose, expression and etc is denoted as a noisy face which is, in our case the input, while a frontal face with neutral expression is denoted as a canonical face which is the output.

The first two layers of the framework perform progressive non-linear dimension reduction and yield a low dimensional feature whose effective dimension is half the dimension of the original RGB features. The nodes in the higher layer learn the statistical dependencies among the nodes in the adjacent lower layer so that the higher layer can discover more complex patterns (abstract) in the input by eliminating the noise and reducing dimensions. The 3rd layer performs the de-noising based on a strong supervisory signal which is the neutral frontal face in our case. The last layer performs the discriminant analysis based on a single representative class instance thus ensures the features observed in the reconstruction layer are highly discriminative. We use the neutral face image of each class as the representative image instead of using the actual mean image. This will ensure that the proposed DDA framework will learn a forceful projection for each noisy face image to approximate its frontal neutral face. The stacked R, G, B channel features are combined and used as the input to the framework. In order to reduce the effects of illumination and color contrasting, histogram equalization is performed on the V-channel after converting the RGB image to the HSV color space. The pixel intensities are then normalized from the range 0 – 255 to 0 – 1 to be compatible with the DDA’s operating range. This setup ensures that the proposed DDA framework learns the optimal combinations of weights for pixel based on the global objective function.

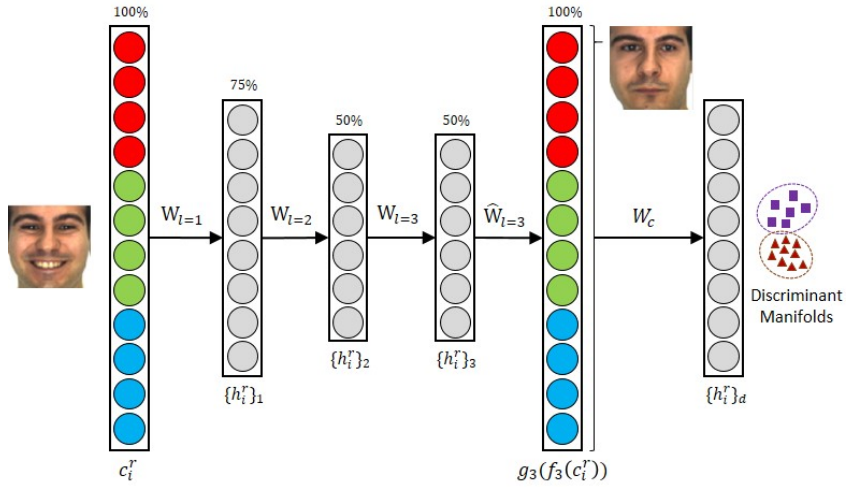


Figure 5.5: DDA framework where $\{h_i^r\}_1, \{h_i^r\}_2$ is the low dimensional noisy feature learned at Layer 1 and 2 for the r_{th} input image of class i , which is c_i^r . $\{h_i^r\}_3$ denotes the noise-less feature learned at Layer 3 (de-noising layer) in the observed low dimensional space. $g_3(\cdot)$ represents the decoder function. Hence the discriminant layer where $\{h_i^r\}_d \in \mathbb{R}^{class\ count-1}$ is shown as the right most layer.

The layer-wise pre-training procedure is performed as shown in Figure 5.4. In order to refine the features further, we perform fine-tuning (Section 3.3.3) on the database of interest by stacking the pre-initialized layers one after another to jointly optimize the objectives as shown in Figure 5.5. We employ the full batch gradient descent algorithm and the backpropagation mechanism on the

cost functions to find the optimal parameters.

After the fine-tuning phase of the DDA framework, the feature representations learned at the denoising layer h_3 and the reconstruction layer h_4 will ensure that they are highly discriminative and suitable for recognition purpose. Since the low dimensional feature (h_3) and the features at the reconstruction layer (h_4) are obtained via a strong supervised non-linear discriminant criterion, these features are highly favorable with LDA analysis followed by Nearest Neighbour (NN) classifier for recognition. In addition, we utilize the DDA induced features followed by PCA and SRC separately to observe the properties of the learned features. The experimental results are reported in the following section.

5.5.1.1 Experiments

Face images in AR (Martínez and Benavente, 1998), Curtin (Li *et al.*, 2013a) and MultiPIE (Sim *et al.*, 2002) databases are used for experiments. All images are cropped, aligned and resized to 33x33 resolution. Experiments are conducted on expressions and pose separately to evaluate the effectiveness of the proposed approach. A validation dataset is used in every test case to select the optimal parameters for the objective functions. Furthermore, we compare our results with the AutoDNet framework introduced in Chapter 4.

5.5.1.2 Facial Expression Experiments



Figure 5.6: Different expressions and the corresponding indices.

Three distinct experiments were conducted in regard to the expression problem, each consisting of 8 test cases (TC) to evaluate the performance of the proposed DDA framework . Figure 5.6 shows

the images used with different expressions including extremely opened mouth.

5.5.1.3 Same Identity Experiment

In this setup, experiments were conducted on the AR database in isolation, and training and testing were performed on different images of the same subject. One out of the 8 images from each identity was taken for testing while the other 7 images from the same identities were used for training. The test cases were formed as shown below:

- For each test case i : Test on the i^{th} image of each identity and train on the remaining 7 images of each identity.

The data split for the training, validation and testing process of the 100 identities in the AR database is described as follows. One of the 8 images from 75 identities was used for testing in each test case. The remaining 7 images from the 75 identities were used for training. Images of the same indices from the other 25 identities were used for validation. Results of face recognition on the same identity experiment are shown in Table 5.1.

Table 5.1: Results of the same identity experiments on AR database.

Test case Index	PCA	LDA	KPCA (Gaussian)	KDA (Gaussian)	SRC	AutoDNet			DDA		
						PCA	LDA	SRC	PCA	LDA	SRC
1	94.0	100.0	94.6	100.0	100.0	98.7	100.0	100.0	98.7	100.0	100.0
2	93.3	100.0	94.6	100.0	98.7	98.7	100.0	100.0	98.7	100.0	100.0
3	88.0	100.0	90.6	100.0	98.7	94.7	100.0	100.0	96.0	100.0	100.0
4	92.0	98.7	93.3	100.0	97.3	97.3	100.0	100.0	96.0	100.0	100.0
5	84.0	97.3	86.6	98.7	96.0	94.7	100.0	97.3	94.7	100.0	100.0
6	94.7	100.0	96.0	100.0	97.3	97.3	100.0	98.7	98.7	100.0	100.0
7	74.7	93.3	77.3	97.3	94.7	80.0	96.0	98.7	84.0	100.0	100.0
8	78.7	92.0	81.3	97.3	96.0	79.0	93.0	96.0	84.0	99.2	100.0

5.5.1.4 Cross Identity Experiment

In this experiment, training and testing are performed on mutually exclusive datasets from the AR database. This setting exploits the DDA's generalization ability on mutually exclusive datasets that were observed under the same environmental conditions.

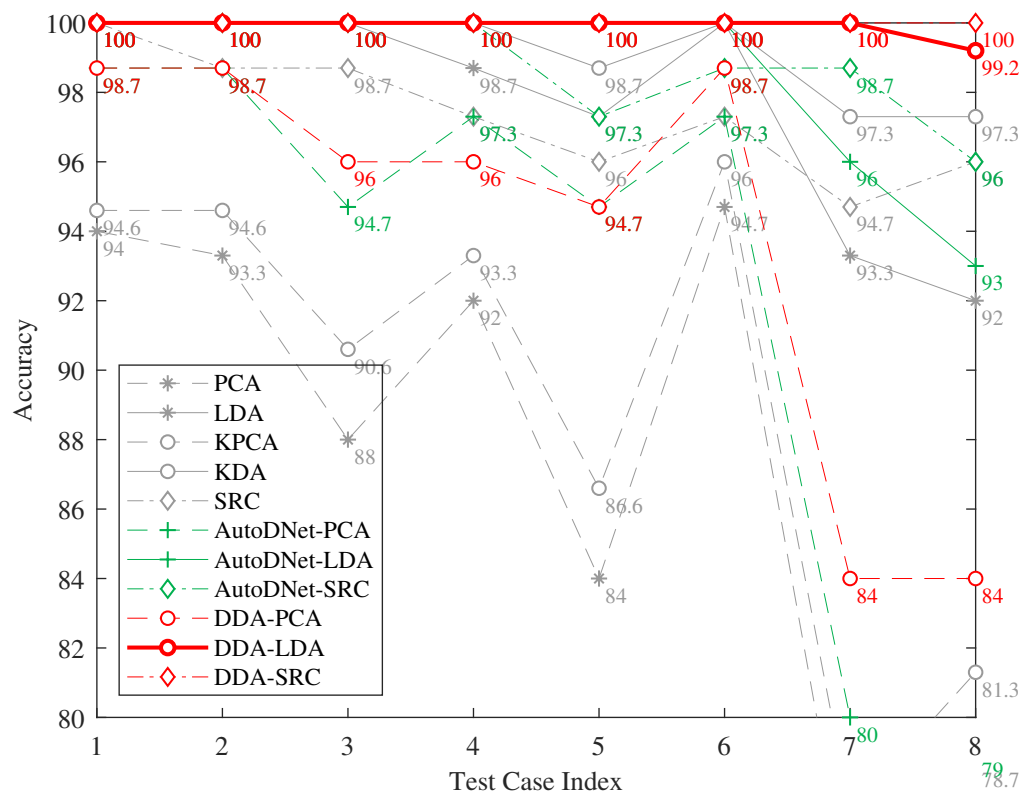


Figure 5.7: Results of the same identity experiments on AR database.

All 8 images of 50 identities from the AR database were taken for training and another 25 identities were used for validation. Images of the remaining 25 identities were split into gallery and test image. The i^{th} test case follow the same format as described in the previous setup, i.e. in Test Case i ($i = 1, \dots, 8$), Image i was used for testing and the other 7 images formed the gallery. Results are shown in Table 5.2:

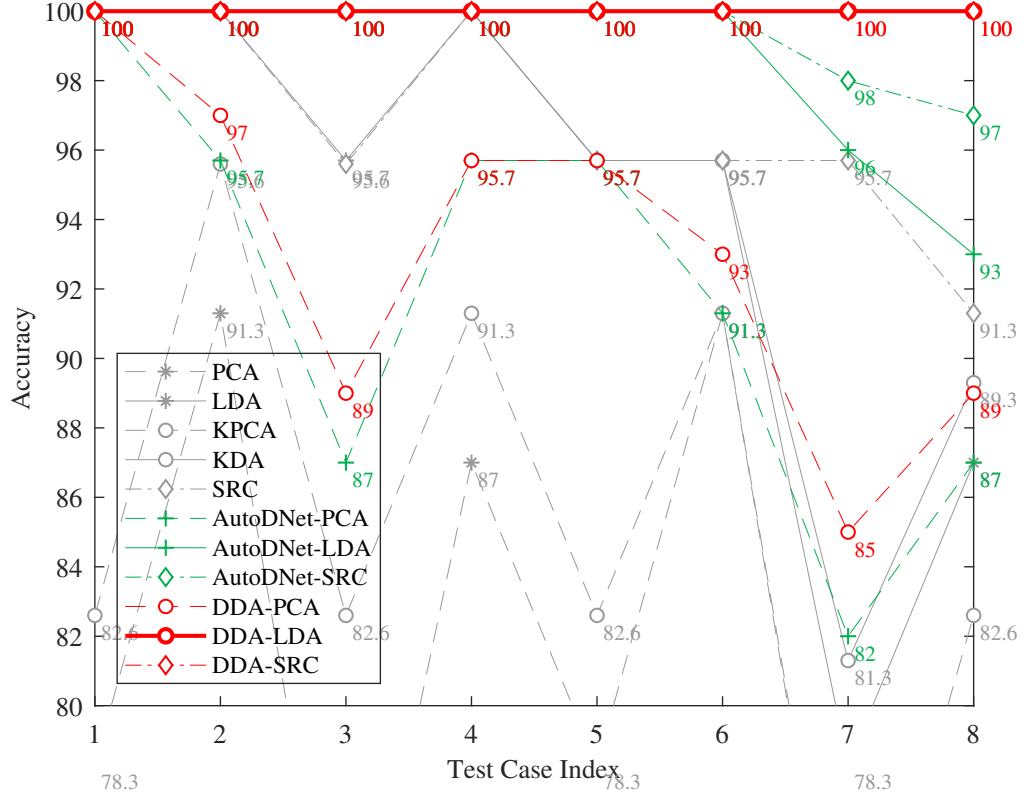


Figure 5.8: Results of the cross identity experiments on AR database.

5.5.1.5 Cross database experiment

In this experiment, training and testing were performed on different databases that are captured under different environmental conditions. This is a challenging task for the proposed framework with an aim to test the generalization ability for face images captured under the various environmental conditions. The i^{th} test case follows the same formats described in the previous setup. The training set consist of 75 subjects from the AR database and another 25 subjects from AR were taken for validation. 50 subjects from the Curtin database were used for testing. Experimental results in terms of face recognition rates are shown in Table 5.3:

Table 5.2: Results of the cross identity experiments on AR database.

Test case Index	PCA	LDA	KPCA (Gaussian)	KDA (Gaussian)	SRC	AutoDNet			DDA		
						PCA	LDA	SRC	PCA	LDA	SRC
1	78.3	100.0	82.6	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
2	91.3	100.0	95.6	100.0	100.0	95.7	100.0	100.0	97.0	100.0	100.0
3	70.0	95.7	82.6	100.0	95.6	87.0	100.0	100.0	89.0	100.0	100.0
4	87.0	100.0	91.3	100.0	100.0	95.7	100.0	100.0	95.7	100.0	100.0
5	78.3	95.7	82.6	95.7	95.7	95.7	100.0	100.0	95.7	100.0	100.0
6	91.3	95.7	91.3	95.7	95.7	91.3	100.0	100.0	93.0	100.0	100.0
7	69.6	78.3	70.0	81.3	95.7	82.0	96.0	98.0	85.0	100.0	100.0
8	74.0	87.0	82.6	89.3	91.3	87.0	93.0	97.0	89.0	100.0	100.0

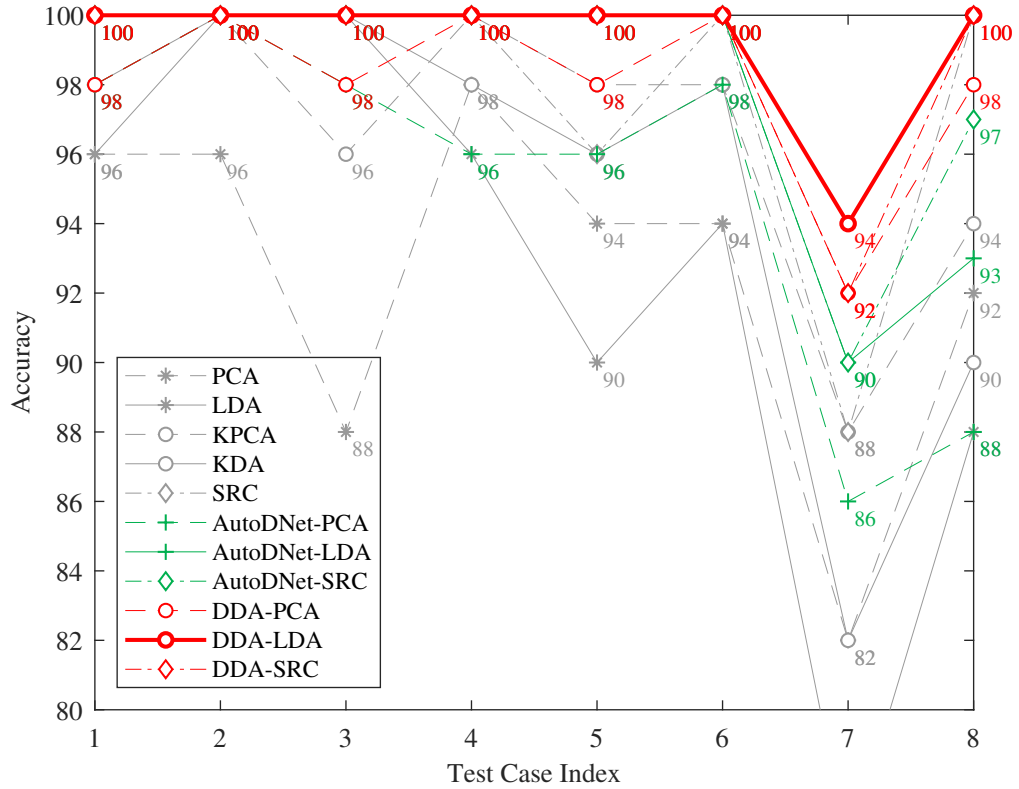


Figure 5.9: Results of the cross database experiments on AR and Curtin database.

Table 5.3: Results of the cross database experiments on AR and Curtin database.

Test case Index	PCA	LDA	KPCA (Gaussian)	KDA (Gaussian)	SRC	AutoDNet			DDA		
						PCA	LDA	SRC	PCA	LDA	SRC
1	96.0	96.0	98.0	98.0	100.0	98.0	100.0	100.0	98.0	100.0	100.0
2	96.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
3	88.0	100.0	96.0	100.0	100.0	98.0	100.0	100.0	98.0	100.0	100.0
4	98.0	96.0	100.0	98.0	100.0	96.0	100.0	100.0	100.0	100.0	100.0
5	94.0	90.0	98.0	96.0	96.0	96.0	100.0	100.0	98.0	100.0	100.0
6	94.0	94.0	98.0	98.0	100.0	98.0	100.0	100.0	100.0	100.0	100.0
7	82.0	76.0	88.0	82.0	88.0	86.0	90.0	90.0	92.0	94.0	92.0
8	92.0	88.0	94.0	90.0	100.0	88.0	93.0	97.0	98.0	100.0	100.0

Discussion. As shown in Table 5.1, Table 5.2 and Table 5.3, DDA framework performs consistently better compared to other methods for face images with different types of facial expression including extremely opened mouth. The differences in DDA-PCA vs AutoDNet-PCA and KPCA Scholkopf *et al.* (2012), DDA-LDA vs AutoDNet-LDA and KDA B. Alacam (2012), DDA-SRC vs AutoDNet-SRC and SRC clearly show the improvement by utilizing the non-linear discriminant error criterion. Comparing DDA-LDA and DDA-PCA, it is conceivable that the opened mouth faces lie in a LDA structure (Gaussian) in the observed low dimensional space. It also shows that learned features favour linear approximation of the face images in the observed space (refer DDA-SRC performance). Hence DDA framework outperforms the other linear dimensionality reduction techniques and the shallow models (kernels). Irrespective of the identities at training and the environmental conditions in which the images are acquired, DDA framework is highly invariant to the opened mouth noise (Figure 5.7, Figure 5.8 and Figure 5.9) thus claims its immense generalization ability with better low dimensional features for robust face recognition.

5.5.1.6 Face Pose Testing

The pose related experiments were carried out in the same fashion as mentioned in the previous section. The grayscale images showing various face pose from the popular MultiPIE database (Gross *et al.*, 2010) as well as the Curtin database (Li *et al.*, 2013a) are used in these experiments. There are 6 poses per subject in the MultiPIE database, as shown in Figure 5.10. Test cases are defined as in the previous experiments, except that there are now only 6 of them (excluding the frontal pose) which are labeled by the pose angles. We follow the usual test setups of other pose related methods where the same identity experiment is excluded since the training needs to contain

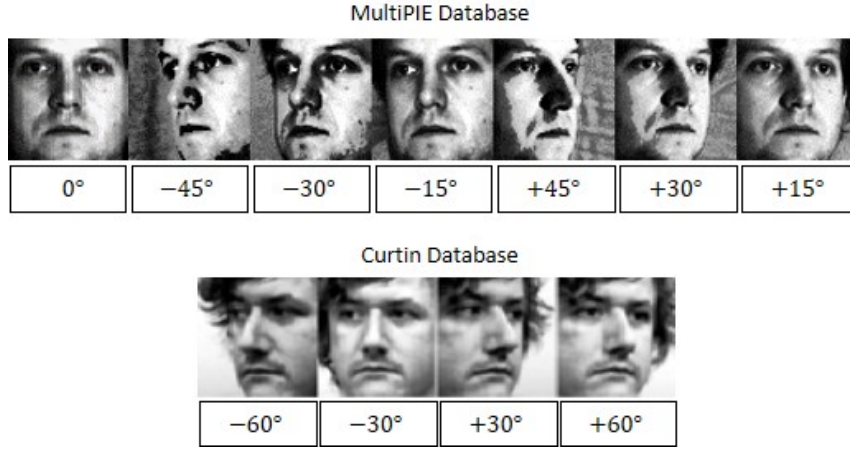


Figure 5.10: Images with different poses and the indices.

the pose to be recognized. The other two experiments were conducted instead, as detailed below.

5.5.1.7 Cross identity experiment

In this setup, training and testing was performed on mutually exclusive datasets from the MultiPIE database. 6 different test cases (excluding the frontal case) on different face pose of varying degree $45^\circ - 15^\circ$ are conducted. We follow the same dataset configuration used in Kan *et al.* (2014) to compare and contrast our method with the existing state of the arts 2D and 3D methods presented in Kan *et al.* (2014). The images of all 337 subjects from MultiPIE at 7 poses with neutral expression and frontal illumination are used. We use images from the first 200 subjects (subject ID 001 to 200) for training, which consisted of 4,207 images in total. The images from the remaining 137 subjects are used for testing, with 1,879 images in total. Out of them, the frontal face images from the earliest photo-taking session for the 137 subjects are used as gallery images (137 in total), and images from the other poses are used as probe images (1,742 in total). SRC, LDA, PCA, KDA, KPCA results were excluded since these techniques generally failed (or not significant) and not robust under large pose variations (beyond the range of -15° to $+15^\circ$). Experimental results using all the other existing techniques in terms of face recognition rates are shown in Table 5.4:

5.5.1.8 Cross database experiment

This setup aims to evaluate the proposed DDA framework against different databases captured under different environmental conditions. This is a novel experiment setting to assess the general-

Table 5.4: Results of the cross identity experiments on MultiPIE.

Method	-45°	-30°	-15°	+15°	+30°	+45°	Average
DDA-SRC	73	92	98.5	98.5	93	74.1	88.2
DDA-LDA	63	90	99.2	99.2	92	64.4	84.6
DDA-PCA	60	86	98.5	98.5	88	60.7	82
SPAE (Kan <i>et al.</i> , 2014)	84.9	92.6	96.3	95.7	94.3	84.4	91.4
DAE (Bengio, 2009)	69.9	81.2	91	91.9	86.5	74.3	82.5
GMA (Sharma <i>et al.</i> , 2012)	75	74.5	82.7	92.6	87.5	65.2	79.6
CCA (Florin <i>et al.</i> , 2012)	53.3	74.2	90	90	85.5	48.2	73.5
PLS (Sharma and Jacobs, 2011)	51.1	76.9	88.3	88.3	78.5	56.5	73.3
MDF (Li <i>et al.</i> , 2012b)	78.7	94	99	98.7	92.2	81.8	90.7
Asthana11 (Asthana <i>et al.</i> , 2011)	74.1	91	95.7	95.7	89.5	74.8	86.8

ization ability of the proposed DDA framework with respect to the pose problem. None, but one, of the previous related work has reported their ability to work in this kind of setting, hence only that work will be compared with. The images of all 337 subjects at pose variation (+45° to -45°) in the MultiPIE database were taken as the training set while all images of the 50 subjects at the same pose range from the Curtin database were used as the test set. Note that the Curtin database only consists of face images at pose angles +/-60°, +/-30° and 0° (Figure 5.10). The frontal face images from the 50 subjects in the Curtin database were used as gallery images and images of the other poses as probe images (each pose consists of 3 images). SRC, LDA, PCA results were excluded as mentioned above. Experimental results are shown in Table 5.5:

Table 5.5: Results of the cross database experiments on MultiPIE and Curtin database.

Method	-60°	-30°	+30°	+60°
DDA-SRC	33	70	71	31
DDA-LDA	28	77	77.8	27.6
DDA-PCA	24	69.2	71	23
AutoDNet-SRC	20	60	62	19
AutoDNet-LDA	19	68	69.7	16
AutoDNet-PCA	17	53.8	54.4	15

Discussion. As shown in Table 5.4, the proposed DDA framework outperforms many 2D methods such as DAE (Bengio, 2009), GMA (Sharma *et al.*, 2012), CCA (Florin *et al.*, 2012), PLS (Sharma and Jacobs, 2011) and 3D method (Asthana *et al.*, 2011) as displayed in the average column. It

outperforms the best state of the art method SPAE (Kan *et al.*, 2014) in small pose variations ($\pm 15^\circ$). Note that in this case, DDA framework learns the same transformation from $\pm 15^\circ$ to 0° like SPAE, but with a non-linear discriminant criteria which leads to better performance. Although it does not perform as well as SPAE in large poses (beyond 30°), unlike SPAE, the proposed DDA framework can be used with various kinds of machine learning problems to learn effective features while preserving the class structure of data to facilitate recognition. More importantly, the proposed DDA framework can be trained with no prior information about the pose or expression while such information is necessary for training the SPAE (Kan *et al.*, 2014) model. Moreover, the non-linear discriminant error criterion combined with the deep structure can successfully be applied in complex problem domains where the regular LDA criteria fail. As shown by Table 5.5, the proposed DDA framework demonstrates a decent level of tolerance for recognizing faces captured under different environmental conditions within the pose range (-30° to $+30^\circ$), while other existing methods fail in such a challenging setting. Although the results for larger pose variations are unsatisfactory, the experiments clearly show the ability of the proposed DDA framework to learn effective features across databases irrespective of the environmental conditions, thus unveil a new direction for further improvements.

5.5.2 Discriminant Coentropy in Classification Context

There have been many very useful attempts to extract robust low dimensional descriptive or discriminative features of data during the past decades. Although notable improvement has been achieved by these methods such as Principal Component Analysis (PCA-L2) (Turk and Pentland, 1991), PCA-L1 (Kwak, 2008), Φ -PCA (Iglesias *et al.*, 2007), LDA (McLachlan, 2004), MCSC (Yang *et al.*, 2017), they do not explicitly and effectively consider the robustness of the extracted features. This is mostly because these approaches adopted metric norms sensitive to large outliers, which is quite common in practical applications. Learning a robust and discriminative feature representations remains a challenging task as any part of data could be corrupted in real-world scenarios and sometimes the magnitude of noise may be significant.

PCA based methods are linear data transformation techniques that have been utilized extensively in image processing and machine learning problems in the last two decades as they can represent high-dimensional data in a low-dimensional subspace. However, PCA itself has severe limitations as large errors will dominate the mean square error (MSE) due to the adoption of L_2 -norm. Thus, the conventional PCA based methods are prone to the presence of outliers that are significantly far away from the rest of the data points. In He *et al.* (2011), the authors have utilized the Maximum Correntropy Criterion (MCC) (Liu *et al.*, 2007) cost formulation to overcome robust issue successfully. As it is only based on the idea of PCA, MCC lacks the ability to exploit class discriminatory information during the subspace learning process. Thus, MCC provides insufficient

discriminative information in learning robust feature representation, especially under tough conditions (Fidler *et al.*, 2006). One key challenge for classification tasks is to extract robust and discriminative features with an aim of reducing intra-class variations while enlarging inter-class differences. Linear Discriminant Analysis (LDA) is a well-known supervised linear feature extraction method for such a purpose. However, the conventional LDA is based on L2-norm, and thus highly sensitive to the presence of outliers.

5.5.2.1 DCC for Computer Vision

A novel feature extraction approach based on the Maximum Correntropy Criterion (MCC), named as Discriminant Correntropy Criterion (DCC) was introduced in Section 5.3.2 to perform effective feature extraction under the presence of outliers. The DCC is formulated precisely with explicit interpretations and its solution is derived via an efficient and robust algorithm. The obtained features are resilient to large outliers and rotations of data and can be effectively utilized to find the best representations of each class. Furthermore, the obtained feature representations in the projected low dimensional space via DCC are expected to preserve the discriminatory information while being robust to outliers. Thus they should be more suitable for recognition purpose. Contrary to the conventional LDA/PCA based approaches in low dimensional subspace learning, the DCC cost formulation is able to learn the robust and discriminative features while estimating class means and the total data mean at the same time. We utilize the estimated class means combined with the NN classifier in our experiments to evaluate the performance of the proposed approach. Extensive experiments in the classification context on popular datasets show the effectiveness and the robustness of the proposed approach in comparison with some p-norm-based methods and the corresponding classification performances are improved significantly.

5.5.2.2 Experiments

In this section, several experiments are conducted on some pattern recognition problems and the performance of the proposed approach is compared with the state-of-the-art subspace learning methods. The MCC-PCA (He *et al.*, 2011) method is chosen to represent all PCA based approaches since it performs superior to all of them. The AutoDNet that was introduced in Chapter 4 AutoDNet-LDA (for non-extreme noise), SRC (Wright *et al.*, 2009) methods are evaluated against the proposed approach to compare the performance and the superiority of the DCC-criteria over outliers. Experiments were performed on a toy dataset as well as four public databases for recognition, namely, the AR (Martínez and Benavente, 1998), ORL (Samaria and Harter, 1994), CMU (Sim *et al.*, 2002), and MNIST (LeCun *et al.*, 1998) databases. We follow the ideas of MCC (Liu

et al., 2007) to estimate the bandwidth σ as discussed above.

5.5.2.3 Toy Data Set

We first construct a toy dataset of 10 samples clustered into two categories with a large outlier included in Class 1 as shown in Figure 5.11 (a). The outlier in Class 1 is intentional in order to intuitively evaluate the effectiveness of DCC. The learned projections are plotted as 1-dimensional signals in Figure 5.11 (b), Figure 5.11 (c) and Figure 5.11 (d) corresponding to LDAL2, LDA-R1, and DCC respectively. In this experiment, DCC is randomly initialized with orthogonal projection matrices.

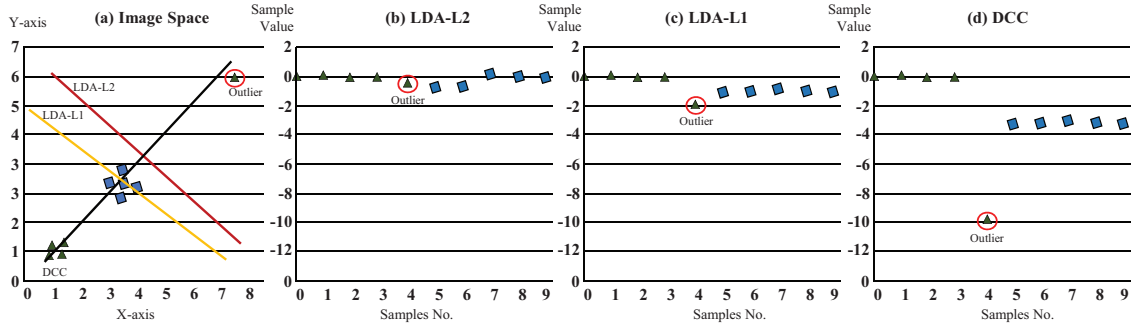


Figure 5.11: (a) Samples in toy set and 1D subspaces. Solid lines indicate, Red: LDA subspace, Amber: LDA-L1 subspace, Black: DCC subspace. (b) Results of LDA-L2 projection. (c) Results of LDA-L1 projection. (d) Results of DCC projection.

Discussion: According to Figure 5.11, the inter-class scatter of the two-class samples except for the outlier sample in Figure 5.11(d) is much larger than those in Figure 5.11(b) and Figure 5.11(c). This is clearly shown in Figure 5.11(a) with representations of the learned 1D subspaces of LDA-L2, LDA-L1, DCC. Hence it shows DCC's superiority and robustness to large outliers in comparison with other subspace learning methods. The random initialization of DCC further demonstrates the robustness of the learned subspace toward outliers.

5.5.2.4 AR Database

The AR (Martínez and Benavente, 1998) dataset consists of over 3,200 color frontal face images of 126 subjects. In our experiment, frontal views with different expressions and varying degree of occlusion (sun-glass and scarf) for each subject are chosen as shown in Figure 5.12. Two experiments are conducted here. In the first experiment, images as shown in the 2nd row in Figure 5.12 with extreme open mouth expression, scarf, and sunglass faces are used for training. The

remaining images as shown in the 1st row in Figure 5.12 are randomly chosen to augment the training or used for testing with a 50% split. All images are cropped, aligned, resized to 33×33 resolution and the histograms are equalized to normalize the illumination. The whole training and testing process is performed 5 times with randomly chosen 75 identities. The average recognition performances are shown in Table. 5.6.



Figure 5.12: Different expressions and occlusions

Table 5.6: Recognition rates on AR dataset.

Method	Recognition Rate
DCC	92.3 ± 0.7
AutoDNet-LDA	85.5 ± 1
SRC (Wright <i>et al.</i> , 2009)	86 ± 1.5
LDA-L2	80 ± 1.8
LDA-L1 (Li <i>et al.</i> , 2010)	87 ± 0.8
MCC-PCA (He <i>et al.</i> , 2011)	83.4 ± 0.4

Discussion: As shown in Table 5.6, the proposed method performs superior to other methods. This experiment shows the effectiveness of DCC in learning a discriminatory robust subspace to perform the recognition.

The second experiment on the AR database is conducted with a setting that is similar to Zhou and Kamata (2012) where 7 random images per identity are chosen for training (60%) and the remaining 5 images are used for testing (40%). The average recognition rates in this experiment are shown in Table. 5.7.

Discussion: As shown in Table 5.7, the proposed method outperforms the other state of art related methods as expected. This experiment is composed of a tougher setting than the previous controlled setting. It further demonstrates the effectiveness of DCC in learning a discriminatory robust subspace to solve facial expressions or occlusions issues in face datasets.

Table 5.7: Recognition rates on AR dataset.

Method	Recognition Rate
DCC	86
LDA-MCC (Zhou and Kamata, 2012)	83.7
LDA-L2	69.7
LDA-L1 (Li <i>et al.</i> , 2010)	76.1
MCC-PCA (He <i>et al.</i> , 2011)	67.2

5.5.2.5 ORL Database

The ORL (Samaria and Harter, 1994) dataset consists of 40 identities where each identity has 8 frontal illuminated images with different expressions and non-planar rotations. We use 3 images from each identity along with their corrupted versions as the training set while the remaining 5 images are used as the testing set. Three experiments are conducted here with corruptions occluding 25%, 50% and 100% of the images respectively. Images with total corruptions are also included to evaluate the robustness of the proposed method against the other methods. The training and testing samples are shown in Figure 5.13 and Figure 5.14 respectively. All images are grayscale, normalized to a resolution of 32×32 pixels and histogram equalized to normalize the illumination. The corruptions are made at random locations. Experimental results in terms of recognition rates are shown in Table 5.8, Table 5.9 and Table 5.10.

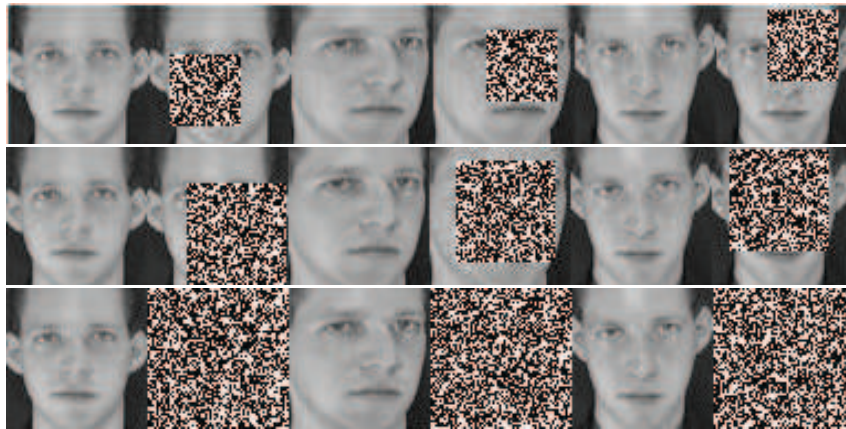


Figure 5.13: Training sets with corruptions

Discussion: As shown in Table 5.8, Table 5.9 and Table 5.10, the proposed DCC method performs consistently better compared to the other state-of-the-art methods for face images with various types of corruption at random locations. DCC, LDA-L1 and MCC-PCA show a constant recognition rate across the three experiments while LDA-L2 shows a gradual fall in recognition rates



Figure 5.14: Testing set

Table 5.8: Recognition rates of Experiment 1 on ORL dataset.

Method	Corruption	Recognition Rate
DCC	30	80 ± 1
LDA-MCC (Zhou and Kamata, 2012)	30	75.7
AutoDNet-LDA	30	78 ± 0.5
SRC (Wright <i>et al.</i> , 2009)	30	77 ± 0.3
LDA-L2	30	65.6 ± 2.3
LDA-L1 (Li <i>et al.</i> , 2010)	30	72 ± 1.3
MCC-PCA (He <i>et al.</i> , 2011)	30	67.5 ± 0.9

Table 5.9: Recognition rates of Experiment 2 on ORL dataset.

Method	Corruption	Recognition Rate
DCC	50	80 ± 1
AutoDNet-LDA	50	76.5 ± 0.5
SRC (Wright <i>et al.</i> , 2009)	50	77 ± 0.5
LDA-L2	50	63 ± 1.8
LDA-L1 (Li <i>et al.</i> , 2010)	50	72 ± 1.3
MCC-PCA (He <i>et al.</i> , 2011)	50	67 ± 0.9

Table 5.10: Recognition rates of Experiment 3 on ORL dataset.

Method	Corruption	Recognition Rate
DCC	100	80 ± 1
AutoDNet-LDA	100	75 ± 1
SRC (Wright <i>et al.</i> , 2009)	100	76.5 ± 0.4
LDA-L2	100	49 ± 1.7
LDA-L1 (Li <i>et al.</i> , 2010)	100	72 ± 1.3
MCC-PCA (He <i>et al.</i> , 2011)	100	66 ± 0.4

when the magnitude of the corruption increases. Hence DCC, LDA-L1 and MCC-PCA can be seen as robust to outliers irrespective of the amount of corruption. The performance differences in DCC vs LDA-L2 and DCC vs LDA-L1 clearly show the advantage of utilizing the DCC objective. DCC vs MCC-PCA shows the improvement over utilizing the class discriminatory information in the training phase. The consistent performance rates across Table 5.8, Table 5.9 and Table 5.10 show the robustness of the features learned via the proposed method.

5.5.2.6 MultiPIE Database

The MultiPIE (Gross *et al.*, 2010) dataset consists of 337 identities where each identity has 3 non-illuminated, neutral expression images for varying pose angles. We use images from randomly chosen 100 subjects in $-\{75^\circ, 60^\circ, 45^\circ, 30^\circ, 0^\circ\}$ pose angles as the gallery, which means the training set consists of $100 \times 5 \times 3 = 1500$ images. The extreme pose angles are included as outliers in training. The images from the same identities with -15° pose angle are used as the probe images, i.e., the testing set contains $100 \times 3 = 300$ images). All images are cropped, aligned, resized to 33×33 resolution and pixel intensities are normalized to 0 – 1. Experimental results in terms of recognition rates are shown in Table 5.11.

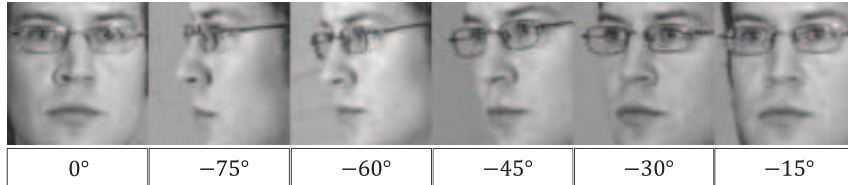


Figure 5.15: Images with different poses.

Table 5.11: Recognition rates on Multi-PIE dataset.

Method	Recognition Rate
DCC	97.3 ± 0.7
AutoDNet-LDA	74.6 ± 1.8
SRC (Wright <i>et al.</i> , 2009)	91.6 ± 1
LDA-L2	45 ± 2.6
LDA-L1 (Li <i>et al.</i> , 2010)	82.7 ± 0.6
MCC-PCA (He <i>et al.</i> , 2011)	70 ± 1.3

Discussion: As shown in Table 5.11, the proposed method outperforms all state of the art methods with a big margin. Since the large pose angles introduce large margin outliers, this experiment showcase the robustness of subspace learning via DCC in such a challenging setting.

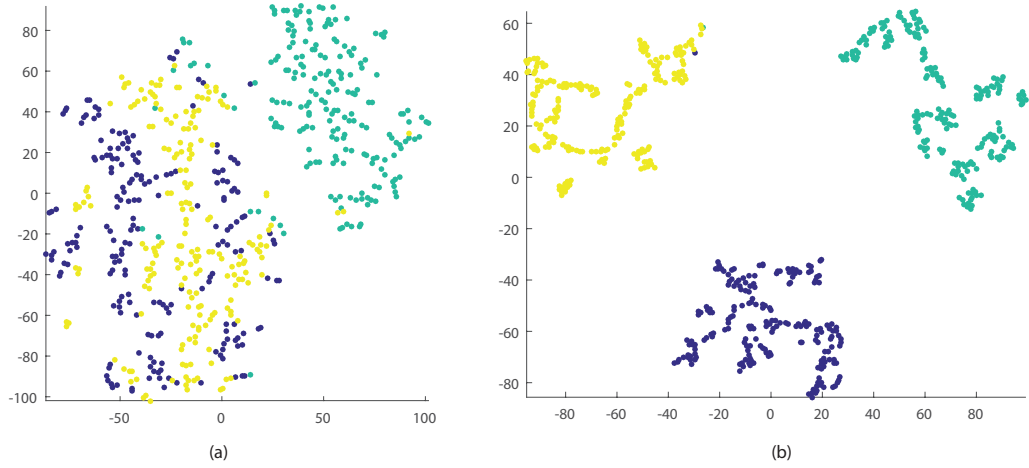


Figure 5.16: Visualizing MNIST digits $\{3, 8, 9\}$ with TSNE. (a) Digits in image space. (b) Representations in subspace.

5.5.2.7 MNIST database

The MNIST (LeCun *et al.*, 1998) database contains 70,000 handwritten digits at 28×28 resolution. It is commonly divided to a training set that consists of 60,000 (*Set A*) samples and a test set of 10,000 (*Set B*) samples. For this database, we choose to work on the digits $\{3, 8, 9\}$ which represent a difficult visual discrimination problem Figure 5.17. Same as in He *et al.* (2011), the $\{3, 8, 9\}$ digits in the first 10,000 samples from *Set A* are taken as our training set and those in *Set B* as our testing set. Three experiments are conducted with subsets of 200, 300, 500 samples per digit randomly selected for training while the testing set is composed of randomly chosen 2700 samples (900 per digit). We also perform random rotations with respect to class centres to the training dataset as shown in Figure 5.18. This is to showcase the L2-norm inherited property of rotational invariance of the proposed method. Experiments are conducted with different subsets of the data and the results in terms of recognition rates are shown in Table 5.11. The TSNE (Maaten and Hinton, 2008) visualized figures of the MNIST dataset for digits $\{3, 8, 9\}$, before and after subspace learning via DCC, are also shown in Figure 5.16.

Table 5.12: Recognition rates on MNIST dataset.

Tr. Samples	DCC	LDA-L2	LDA-L1 (Li <i>et al.</i> , 2010)	MCC-PCA (He <i>et al.</i> , 2011)
200×3	91.8 \pm 0.4	69.1 \pm 1	89.3 \pm 0.6	88.8 \pm 0.2
300×3	93.2 \pm 0.4	75.3 \pm 0.9	90.7 \pm 0.3	90.0 \pm 0.1
500×3	94.1 \pm 0.2	80.2 \pm 0.8	91 \pm 0.4	87.8 \pm 0.1

Discussion: As shown in Table. 5.12, the proposed method performs about 1 – 4% better than

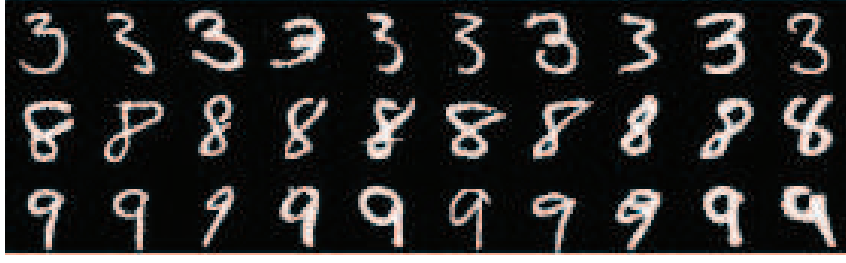


Figure 5.17: Testing set

the other methods on the MNIST dataset in all cases. The improvement is marginal against LDA-L1 due to the less amount of noise in the MNIST digit images compared to other face datasets. MNIST is originally a black and white dataset which is composed of a limited number of gray levels. In contrast, grayscale face images spread across a broad range of gray levels and can incorporate a varying degree of noise thus may lead to large outliers due to face noise and other possible corruptions.

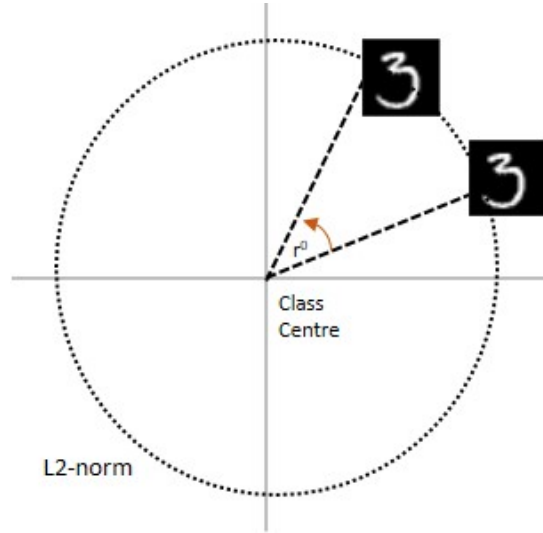


Figure 5.18: Rotation of the data in the image space. Rotations are performed within the sphere denoted by L2 norm. Hence the rotations are performed with respect to class centers.

5.6 Summary

In this chapter, we proposed the DDA framework along with non-linear discriminant error criteria that could be easily embedded in any deep learning network of similar nature. The novel discriminant error criteria are alternatives to the typical squared error cost function that does not consider the labels in the data. These proposed cost formulations are highly beneficial in the

context of supervised learning where the class labels of the data are available.

The proposed deep discriminant analysis (DDA) framework is an extension to the AutoDNet framework introduced in Chapter 5 where it performs progressive non-linear dimension reduction while learning the mapping between the input observed low dimensional space and the output by the non-linear discriminant error criteria introduced in this chapter. We follow a similar experimental setup to Chapter 3 to evaluate the proposed DDA framework under different setting varying from typical case to the most challenging case. In addition, face-pose related experiments are conducted to evaluate the efficiency of the proposed framework against both the 3D and 2D image based state-of-the-art methods. Experiments show that the proposed DDA framework can learn effective features in various face related problem domains and show a good level of generalizability.

Furthermore, the alternative DCC cost formulation is evaluated for its effectiveness in the non-linear discriminant analysis. It performs better and more robust for classification related tasks, especially with outliers. The proposed DCC objective function is robust to outliers (both simple and extreme) and can be efficiently optimized via the suggested gradient-based method. It is also rotation invariant and can correctly update the class means and thereby the total data mean. Experimental results showcase that the proposed method can outperform the other robust PCA/LDA based approaches which are based on 1 or 2-norm.

Although the proposed cost formulations are not directly applicable in the civil engineering domain due to the infinite number of stiffness reduction patterns, the SHM problem could be redefined to identify possible damage patterns that can occur which are localized to certain areas of the structure. This way the damages could be grouped into classes to denote the areas efficiently, for example, damages on the left, middle and right side of the structure. Furthermore, a set of appropriate sub-classes could be defined inside each of these main classes to characterize the damage pattern sufficiently. Such class information could then be utilized to perform the proposed discriminant analysis to learn a useful and robust mapping from input to output. We will work on such applications in the near future.

The complexity of a machine learning problem can change from mild to extreme due to the non-linear nature involved in the problem domain. The proposed approaches so far perform this simplification of dealing with complex tasks via the carefully designed dimensionality reduction and relationship learning components. However, utilizing a deep learning network in isolation may not be efficient to divide the global objective that involves the total complexity into tractable sub-objectives. Complex deep learning system design techniques would help to absorb such complexities involved in these tasks by utilizing multiple deep networks efficiently. A smart deep learning system design is proposed in Chapter 6 using the strategy of dividing the global complexity into

sub-objectives to design efficient and effective deep learning systems for complex tasks.

Chapter 6

Complex Deep Learning Systems

6.1 Introduction

Learning is essential for building intelligent systems thus utilizing simple deep learning models in many machine learning problems has been successful during recent times. Nevertheless, difficult tasks cannot be solved in a single step but rather require multiple processing stages. Careful analysis is required for complex problems, and it is often beneficial to define sub-objectives to address the complexities involved in stages. The proposed approaches in the previous chapters are capable to a certain extent of simplifying these complexities via a carefully designed dimensionality reduction and relationship learning components. However, they might not always be sufficient due to the limited capacity of the frameworks and the nature of the problem.

As more data becomes available and more complex problems are required to be solved, the need for complex deep learning methods or designs of complete deep learning systems has become a necessity. These systems need to be carefully designed in a way that it simplifies the problem complexity in the respective domain. Some systems such as Alrjebi *et al.* (2016); Liang *et al.* (2016) incorporate deep learning models into their system pipeline to enhance the performance of intermediate steps. We justify that multiple deep learning frameworks addressing different segments of a complex problem together are a necessity to resolve large variability in the input data for complex problems such as robust face recognition under occlusion, pose and expression.

As an example, the face pose problem is considered. Identifying subjects with variations caused by poses is one of the most challenging tasks in face recognition, since the difference in appearances caused by poses may be even larger than the difference of identity. In order to model the complicated transforms from the non-frontal pose to frontal pose, the framework introduced in Chapter 3 with multiple layers is preferred attributed to its larger capacity compared to the shallow network. A straightforward implementation is to use the non-frontal face images and frontal face images as the input and output of the AutoDNet framework (Chapter 4) respectively. However, it may be intractable due to the following factors. When the AutoDNet framework is used to transform the non-frontal face images to the frontal ones directly, the objective is highly non-linear leading to a broader search region. Therefore it is prone to be trapped in local minima that deviates far from the

true one, especially given the pre-training strategy for dimensionality reduction and relationship learning stages. Thus a careful division of the complex objective of transforming a pose to the frontal pose is necessary to solve the problem efficiently. Complex deep learning system design techniques would help to absorb such complexities involved in these tasks by utilizing multiple deep networks efficiently.

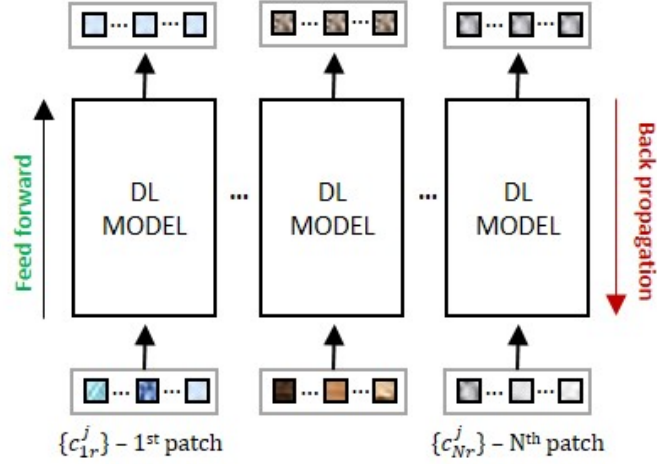


Figure 6.1: Illustration of a patch-based deep learning system.

Localized information processing (patch-based) such as a patch in an image, is one of the typical strategies for the design of deep learning systems by characterizing the data effectively. Patch-based division (Figure 6.1) divides the global non-linear objective into small but tractable goals for the proposed autoencoder based framework in previous chapters by limiting the number of parameters (complexity) that the model should learn in each layer. Processing a low-resolution patch itself requires less computing resources (in terms of CPU and memory) in training a DL model due to its low complexity. Hence it becomes possible to train DL models for each patch in a parallel cluster environment efficiently. The patch-based division helps to detect local structures in input space while measuring the information content of neighborhoods of each feature in the input. In computer vision, multiple resolution images can also be handled in this way. Furthermore, for issues arising due to image alignment, an overlapping patch scheme can be integrated into the training process. Another important fact is the consideration of training all patches independently from one another. The cost function involves the predicted output patch and the corresponding target patch that is a part of the ground truth. Due to the formulation of each DL model (per patch), it is conceivable that the data patches should be treated as a whole to calculate the cost. Thus it preserves the correlation between each patch at the output layer and promotes learning the relationships between the patches. Hence it conceptually relates the features found to the high-level semantics of the data. The choice of good input patch size is a vital factor in this system design. Also, it yields the opportunity of capturing more dependencies (Vincent *et al.*, 2010) between dimensions by producing a high dimensional input. Hence the patch size can be decided

via cross-validation.

Despite all the advantages of utilizing a patch based system design, the patch-based frameworks would not be sufficient to tackle complex non-linearities in certain problem domains especially when the location of a patch affects the whole objective of the problem. If the locations were selected manually by a human user, the accuracy cannot be guaranteed. If the features were automatically selected, it would rely on the accuracy of the feature-based approach. Furthermore, a typical patch-based system design as shown in Figure 6.1 could easily fail if a complex problem like face recognition against varying patterns of occlusion is considered, since the occlusion patterns can occur in arbitrary locations and the information from a patch would not be useful compared to the whole face image. In the context of classification, utilizing the complete input have the advantage of distinctly capturing the most prominent features within the inputs used, to uniquely identify label amongst a gallery set, and to automatically find features. Thus we propose a carefully engineered deep learning system design scheme for complex tasks in the following section.

6.2 Proposed System Design

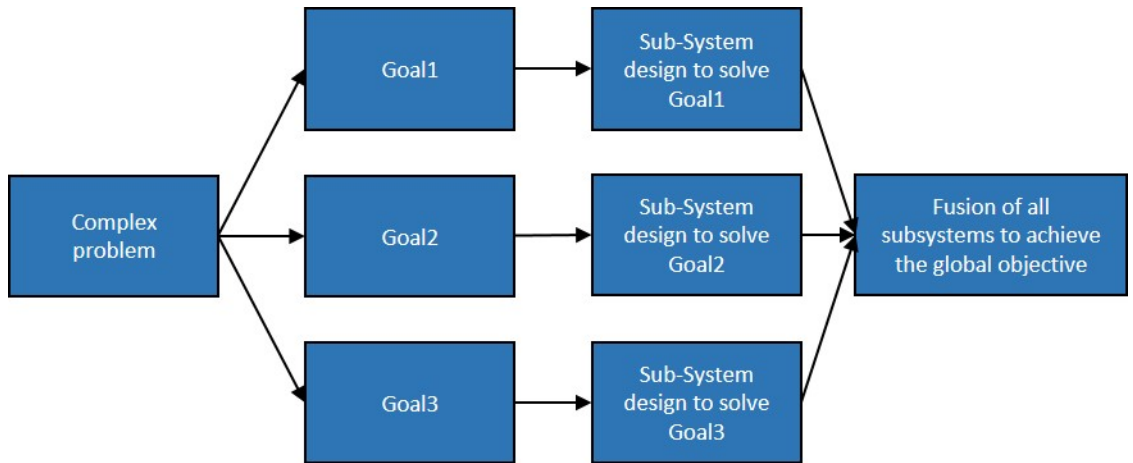


Figure 6.2: Illustration of system designs for complex problems.

A complex objective can be effectively divided into sub-objectives where each sub-objective is tackled with a carefully designed system as shown in Figure 6.2. These sub-systems can be powered by the generic autoencoder based frameworks (Figure 6.3) introduced in Chapter 3, 4, 5 depending on the context. We propose a sub-system design approach that we found effective in computer vision related problems.

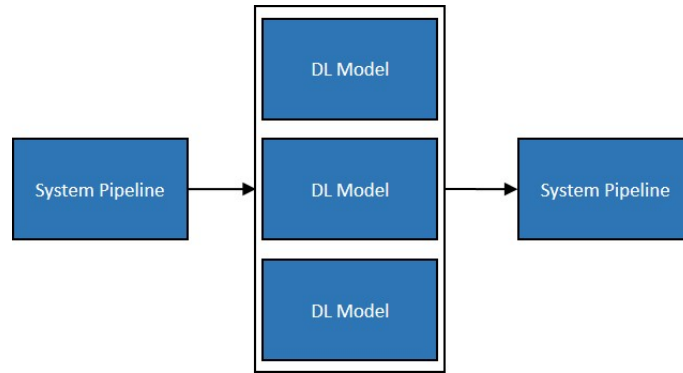


Figure 6.3: Illustration of sub-system design with autoencoder based models.

6.2.1 Deep Model Fusion System

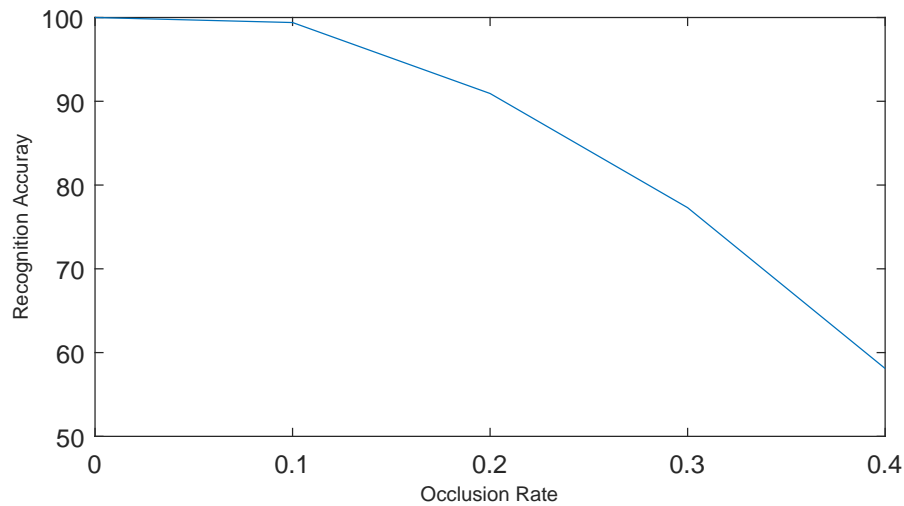


Figure 6.4: Single autoencoder recognition performance on occluded images of different magnitude.

In general, a complex problem can be modeled into a few simpler but tractable objectives where each of these objectives can be effectively addressed via a set of deep learning models. Deep model fusion system design performs model fusion according to the problem description in order to achieve the complex objective overall.

Consider the problem of robust face recognition against occlusion. In practice, as the size and location of the occluded facial areas are unpredictable, face recognition with occlusions remains very difficult. We consider the occlusions that can possibly occur on a face image in various orientations as a complex problem that needs to be divided into many sub-objectives. Reconstruction-based occlusion elimination followed by recognition has shown a proven pathway in dealing with the

complexities that are imposed due to occlusion. In particular, the reconstruction process itself is a problem of highly non-linear nature. Occlusion removal process with a single autoencoder will fail if the occluded portion becomes significant as shown in Figure 6.4. Inspired by the observation that the reconstruction process can be performed in a few stages where each stage performs reconstruction for a smaller occluded region, we intend to utilize shallow autoencoders to perform this task in each stage, considering its impressive ability to handle non-linearity (Vincent *et al.*, 2010).

Direct application of deep autoencoder (Bengio, 2009) to achieve this goal is intractable due to the high complexity of the reconstruction problem, especially when only a limited number of training samples are available. We propose a novel DL system design based on Figure 6.5, which consists of a line of proposed AutoDNet models where each model is utilized in each subsystem to optimize its assigned sub-objective. Furthermore as mentioned in previous chapters, each shallow autoencoder in these models is designed to achieve limited but tractable goal (part of the global non-linearity). With such a reconstruction strategy, the deep network is enforced to approximate the eventual goals layer by layer along the occluded face manifold (Figure 6.6). The trained deep network gradually eliminates the occlusions step by step and reconstructs the whole image at the output (Figure 6.7).

We utilize such trained deep networks, jointly tackling different occlusion patterns to produce the reconstruction of the faces that are occluded with various patterns in varying magnitude as shown in Figure 6.5. For example, the occlusion patterns (Figure 6.8) that were unseen in the training stage could be addressed via the occlusion patterns defined in the training stage as shown in Figure 6.9. In this way, no prior information of the magnitude or the direction of occlusion is required in the proposed approach. The contributions here can be summarized as below:

- Using the shallow autoencoders to remove occlusion progressively to yield a non-occluded face reconstruction via the AutoDNet framework introduced in the previous chapter.
- Utilize the deep model fusion system design to eliminate occlusions in varying degree and magnitude to achieve high performance for Face recognition against occlusion.

Furthermore, An efficient data expansion strategy is also proposed, which will be described in the following section, to generate more data and to minimize the complexities caused by alignment, camera setting, etc.

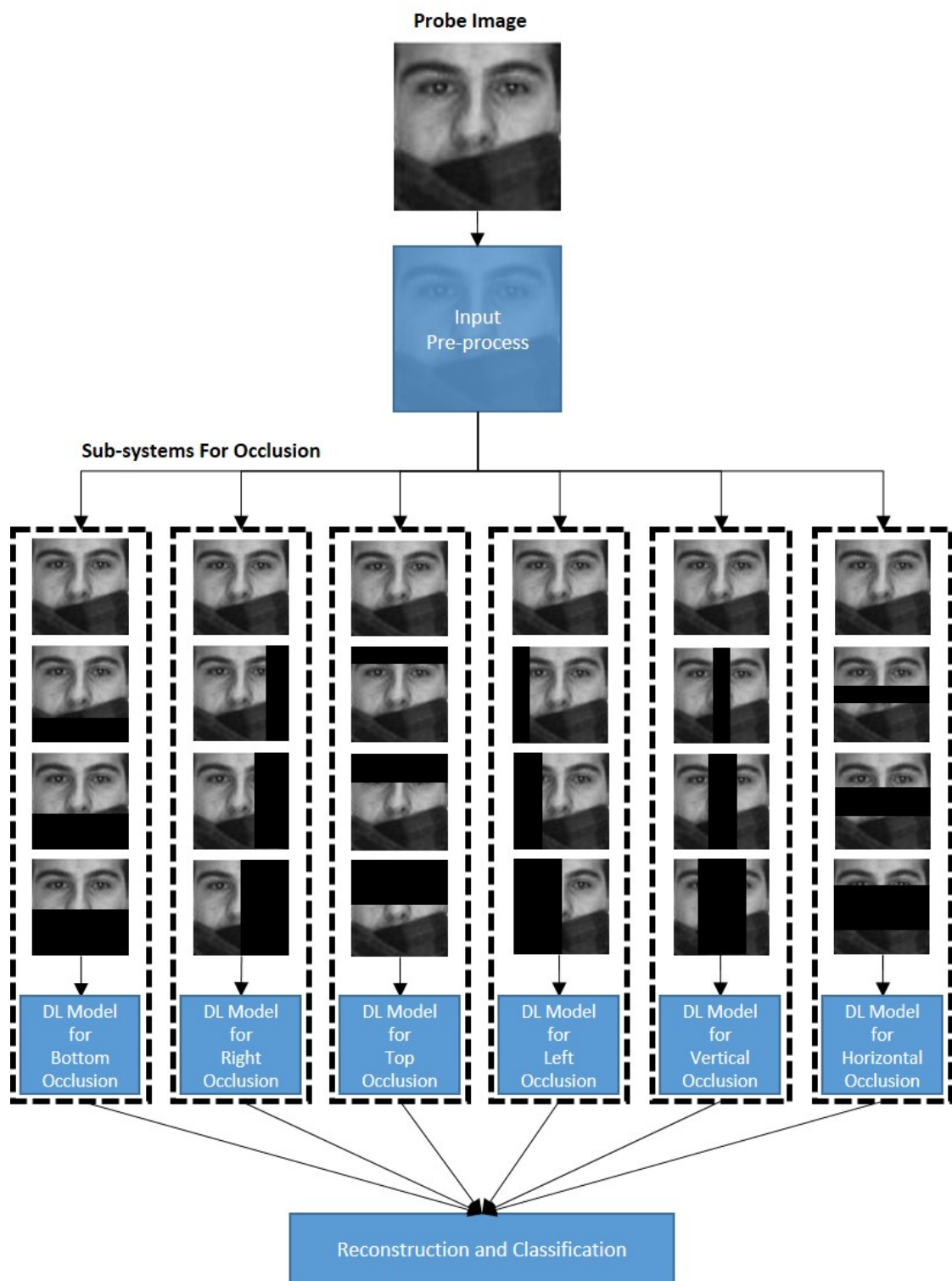


Figure 6.5: Illustration of a deep model fusion system.

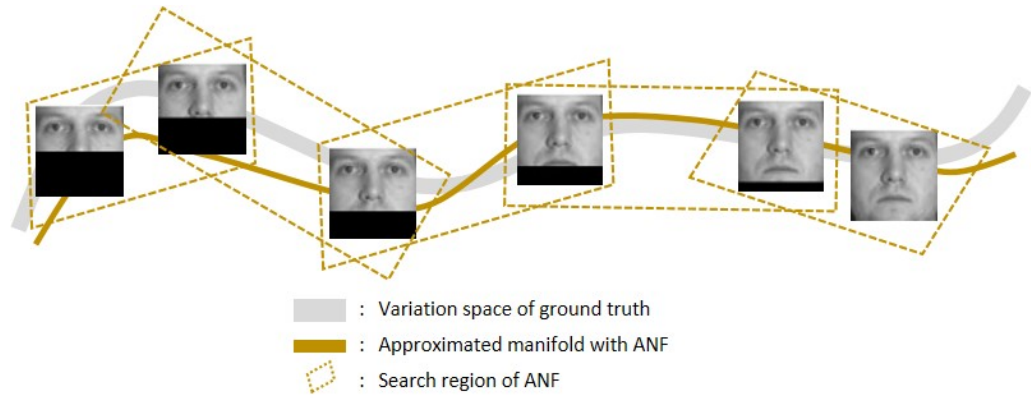


Figure 6.6: Illustration of the occluded face manifold, and the limited but tractable goals that are set during the training of each layer of ANF.



Figure 6.7: (a) Examples of reconstructed faces of MultiPIE database with simulated occlusion; (b) Examples of reconstructed faces of AR database with real occlusion. The top row shows the faces with no occlusion and the bottom row shows the reconstructed faces.

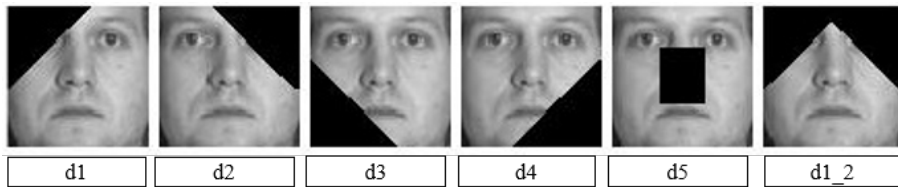


Figure 6.8: Illustration of simulated occlusion in various directions. The alias for each occlusion is shown at the bottom of each column.

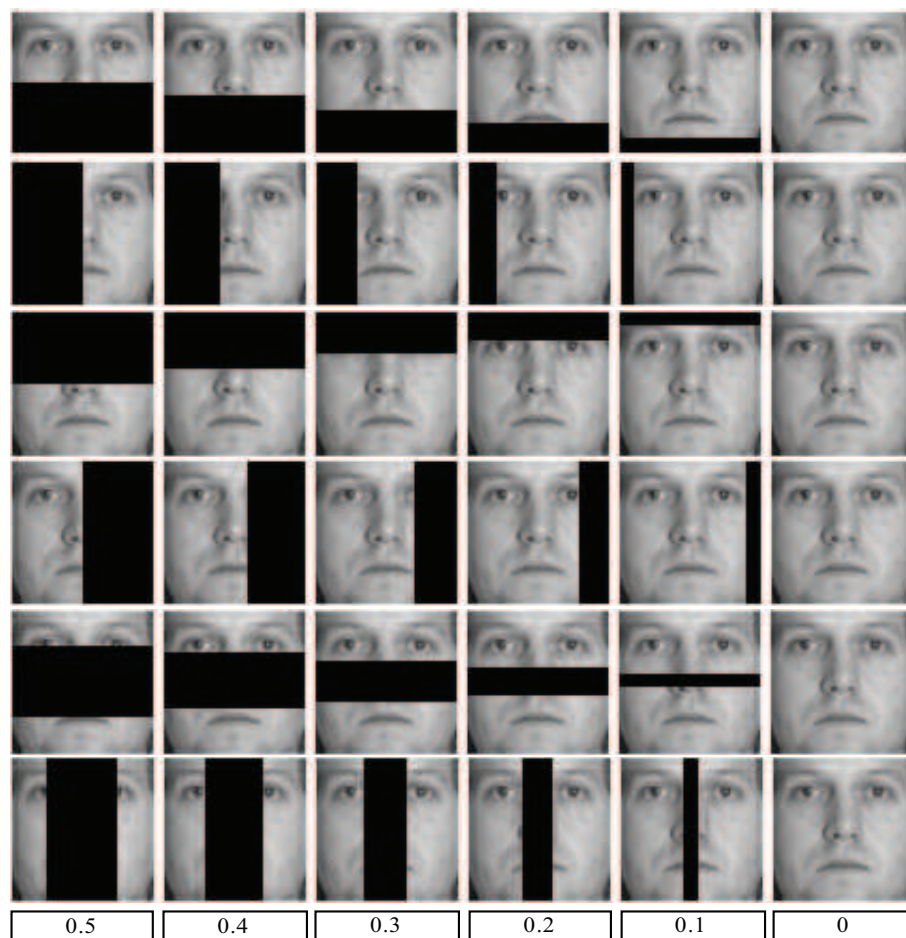


Figure 6.9: Illustration of simulated occlusion in various directions and magnitudes for a frontal face. Rows from top shows the bottom, left, top, right, middle (horizontal), middle (vertical) occlusions with varying ratios respectively. The occlusion ratios are shown at the bottom of each column.

6.2.1.1 Dataset Expansion Strategy

Consider a subset of all frontal face images per identity in a database. For example, Figure 6.10 shows the 3 frontal face image subset from the MultiPIE database. An extended dataset D is defined as below:

$$\begin{aligned}
D &= [D^0, D^1, D^2, D^3, D^4, D^5] \\
D^0 &= [S^{10}, S^{10}, S^{20}, S^{20}, S^{30}, S^{30}] \\
D^1 &= [S^{21}, S^{31}, S^{11}, S^{31}, S^{11}, S^{21}] \\
D^2 &= [S^{32}, S^{22}, S^{32}, S^{12}, S^{22}, S^{12}] \\
D^3 &= [S^{13}, S^{13}, S^{23}, S^{23}, S^{33}, S^{33}] \\
D^4 &= [S^{24}, S^{34}, S^{14}, S^{34}, S^{14}, S^{24}] \\
D^5 &= [S^{35}, S^{25}, S^{35}, S^{15}, S^{25}, S^{15}]
\end{aligned} \tag{6.1}$$

where

$$\begin{aligned}
S^{p0} &= [x_{p0.5}^0, x_{p0.4}^0, x_{p0.3}^0, x_{p0.2}^0, x_{p0.1}^0, x_{p0.0}^0] \\
S^{p1} &= [x_{p0.4}^1, x_{p0.4}^1, x_{p0.3}^1, x_{p0.2}^1, x_{p0.1}^1, x_{p0.0}^1] \\
S^{p2} &= [x_{p0.3}^2, x_{p0.3}^2, x_{p0.3}^2, x_{p0.2}^2, x_{p0.1}^2, x_{p0.0}^2] \\
S^{p3} &= [x_{p0.2}^3, x_{p0.2}^3, x_{p0.2}^3, x_{p0.2}^3, x_{p0.1}^3, x_{p0.0}^3] \\
S^{p4} &= [x_{p0.1}^4, x_{p0.1}^4, x_{p0.1}^4, x_{p0.1}^4, x_{p0.1}^4, x_{p0.0}^4] \\
S^{p5} &= [x_{p0.0}^5, x_{p0.0}^5, x_{p0.0}^5, x_{p0.0}^5, x_{p0.0}^5, x_{p0.0}^5]
\end{aligned} \tag{6.2}$$

D^0 denotes the input dataset for the 1st layer while D^l , ($l > 0$) denotes the target dataset for the l^{th} layer, e.g., D^0 and D^1 are the input and the target dataset to the first shallow autoencoder respectively. S^{pl} is an array of face images for layer l , constructed with the p^{th} ($p = 1, 2, 3$) image as indexed in Figure 6.10. x_{pj}^l is the face image with index p , for layer l , and with the occlusion rate j . This schema promotes reconstructing smaller occlusion region of different face images that belong to the same identity. It is also beneficial in maximizing the learning of useful

transformation from the non-occluded part of the image, mapping the non-occluded segments of different face images that belong to the same identity. Learning this mapping of non-occluded segments can minimize the complexities caused by alignment, camera setting, etc.

6.3 Applications

In this section, the proposed system design is evaluated in applications. Section 6.3.1 evaluates the effectiveness of the proposed system design with respect to a computer vision problem due to its seamless applicability in the problem domain. In the context of civil engineering, the SHM problem of learning a mapping between the inputs and the outputs needs to be reformulated carefully in order to divide the global complexity into simpler and tractable goals. The current form of the SHM problem definition is not directly applicable for the proposed deep model fusion system design thus the experiments for SHM problem are excluded.

6.3.1 Deep Model Fusion System Design For Occlusion Removal

Face recognition (FR) against occlusions is a popular research topic in computer vision and pattern recognition. One of the challenges in automatic FR is the extreme difficulty to avoid different forms of occlusions during image capturing especially when the probe image is acquired in uncontrolled environments. In fact, occlusions pose one of the most significant negative impacts on recognition performance. The magnitude of the loss of facial information, the unpredictable nature of possible occlusion shapes, and the color of the occluded area, etc. would highly degrade the performance of face recognition systems.

Traditional holistic methods such as Principal Component Analysis (PCA) (Turk and Pentland, 1991), Linear Discriminant Analysis (LDA) (Belhumeur *et al.*, 1997) and Locality Preserving Projections (LPP) (He *et al.*, 2005) fail in such challenging settings. This is due to the fact that after the feature projection, the corrupted information in the occluded areas could pollute each feature element. In the past decade, numerous possible solutions have been reported in the literature to handle occlusions.

Saito *et al.* (1999) proposes to restore occluded segments of images before recognition by interpolation using principal component analysis (PCA) while a morphable face model was utilized in Hwang and Lee (2003). A local non-negative matrix factorization (LNMF) based model was proposed in Li *et al.* (2001) for partial occlusion recognition. A spatially localized, parts-based sub-

space representation of face patterns was learned in this model. The Sparse Representation Coding (SRC) approach for occluded face recognition is proposed in Wright *et al.* (2009). It exploits a function of sparsity in recognition and shows effective performance as a novel breakthrough on occluded face recognition. SRC strategy utilizes a simple linear subspace distance metric to decide the subject of the probe image. The linear subspace is determined mainly according to the subject that could linearly represent the probe image most closely and sparsely. In this approach, the most well-known and valuable discriminant information, such as non-linear manifold-ing, are never considered thus limiting the performance of recognition. In Zhang *et al.* (2011), it is argued whether the sparsity is really helpful in the context of recognition. The PCA based reconstruction algorithm proposed in Li and Feng (2013) detects the occlusion via utilizing SRC and reconstructing the occluded region by solving an over-determined system of equations in learning the probe image's representation in PCA subspace. Once the reconstruction is performed, a classifier can be adapted for face recognition. Even though this approach performs superior to other methods by reconstructing the occluded region, the recognition accuracy will be significantly reduced if a high percentage of occlusion is present. Besides, the occlusion region needs to be detected prior to the reconstruction phase and any detection error can negatively impact the reconstruction process. An approach has been proposed in Alrjebi *et al.* (2017) by combining 2D-MCF representation followed by partition-SRC (P-SRC) classifier for face recognition with occlusions. It divides the face image into non-overlapping blocks with 2D-MCF model and then the SRC classifier is used on each block with majority voting. The localized processing of face images in this approach would be highly sensitive to the alignment of face images thus affects the performance of recognition at the final stage. Ou *et al.* (2014) proposes a structured dictionary learning method to learn an occlusion dictionary from the data instead of an identity matrix. In addition, a mutual incoherence of dictionary regularization term is introduced into the cost formulation of the dictionary learning which encourages the occlusion dictionary to be as independent as possible of the training sample dictionary. The occlusion can then be sparsely represented by the linear combination of the atoms from the learned occlusion dictionary and effectively separated from the occluded face image. The classification is then carried out on the recovered non-occluded face images and the size of the expanded dictionary is also much smaller than the one used in the original SRC (Wright *et al.*, 2009). In order to perform effective dictionary learning, this method utilizes various occlusions that can possibly be approximated the probe occlusion sparsely. If black pixels are used to occlude the images (when no prior information about occlusion is assumed) and to learn occlusion dictionary, the learned occlusion dictionary leads to poor performance. Furthermore, researchers have pointed out that SRC based methods suffer from two drawbacks. Firstly, the computational cost of SRC via L1 norm constraints is too high. Secondly, the performance drops heavily when the training sample per subject is insufficient, which makes it impractical in practical usage.

The inherited linear nature in all the above methods negatively influence the complexities involved in images captured in an uncontrolled environment such as alignment, camera setting, etc. Deep

learning has emerged as a successful domain to address such complexities effectively (Arel *et al.*, 2010; Najafabadi *et al.*, 2015). Deep network pre-trained with Restricted Boltzmann Machine (RBM) (Murtaza *et al.*, 2013) or denoising autoencoder (DAE) can capture complicated statistical patterns, and stacked denoising autoencoder (SDAE) can harness serious corruptions and distortions (Hinton, 2012; Vincent *et al.*, 2010; Bengio, 2009; Erhan *et al.*, 2010; Hinton *et al.*, 2006; Vincent *et al.*, 2008). These aspects of autoencoders were taken into consideration in Zhang *et al.* (2013) to propose a stack autoencoder based strategy to detect occluded areas via a mapping-autoencoder (MAE-shallow autoencoder) that requires no prior knowledge of occlusion types and positions. When occluded faces are restored, a deep neural network (DNN) is used for robust face recognition. Their proposed method assumes the occlusion as an additive noise and detects the occlusion prior to the removal phase. In occlusion detection phase, the face image is divided into patches and an occlusion map is learned for an individual patch with a shallow autoencoder pre-trained with Restricted Boltzmann Machines (RBMs). This information is utilized to refine the reconstruction of the image that is produced from a typical Stack Denoising Autoencoder (SDAE) in an iterative fashion. The experiments are only performed by utilizing a single database that contains images under different occlusion conditions but acquired in the same environment setting. To the best of our knowledge, there are no other comparative research work having done on DL context to address the complexities in occlusion specifically, followed by a recognition phase.

We follow a deep model fusion system design that consists of a set of deep learning models to tackle the global occlusion problem as discussed in Section 6.2.1. The proposed system (**ANF** - AutoNet Fusion) is evaluated against the state-of-the-art methods on three publicly available color face databases (MultiPIE (Gross *et al.*, 2010), AR (Martínez and Benavente, 1998), Curtin (Li *et al.*, 2013a)) where each of the setups is based on frontal faces with different facial expressions in uniform illumination. For simplicity in presentation, grayscale images were utilized in order to minimize the number of learnable parameters and the computational power required. The system can be easily extended to handle color face images. All images were cropped and resized into 30x30 resolution. The evaluation of the system was done through three different experiments, each of them consisting of six (6) test cases to evaluate the system's invariability on six (6) different occlusion conditions in six (6) different orientations as shown in Figure 6.9. Frontal face with no occlusion is included in these test cases to evaluate the systems performance over generic FR scenarios. A non-linear squashing function 'sigmoid' with 800 hidden nodes is utilized at each layer in the proposed system. To compare our results, we use the state-of-the-art method presented in Li and Feng (2013)(PCA-R) for reconstruction and to perform classifications with SRC, PCA, Regularized LDA, and Ou *et al.* (2014)(SSRC) that utilizes a dictionary learning approach for occlusion followed by sparse coding for classification. Furthermore, the extensive set of experiments that were performed in the following sections shows the performance gain in such a system design to fully utilize the potential of autoencoder based models.

6.3.1.1 Cross Identity Experiment



Figure 6.10: Frontal face images with their corresponding indices.

The MultiPIE (Gross *et al.*, 2010) database consists of 337 identities where each identity has 3 non-illuminated, neutral expression images for varying pose angles. The 3 grayscale frontal face images as shown in Figure 6.10 are chosen for this experiment. Training was performed with images that belong to a set of identities and testing was performed with images that belong to another set of identities in this database, with no overlap between the identities used for training and testing. Specifically, the training dataset consists of 3 frontal face images that belong to randomly chosen ($200 \approx 337 * 60\%$) identities and the validation dataset consists of face images of non-overlapping ($67 \approx 337 * 20\%$) identities. The testing dataset consists of 2 randomly chosen frontal face images per gallery dataset and 1 frontal face image for the probe from the remaining identities (70). This experiment configuration evaluates the system's generalization ability on mutually exclusive datasets created under the same environmental conditions such as lighting, reflection, camera, etc. The results are shown in Table 6.1.

Table 6.1: Accuracy on Cross Identity.

Occlusion	PCA-R			SSRC	ANF		
	PCA	LDA	SRC		PCA	LDA	SRC
0.5	38.6	80.0	88.2	97.4	90.4	99.2	97.1
0.4	39.6	85.4	89.3	98.2	92.1	100.0	99.2
0.3	43.2	91.4	91.4	98.6	93.6	100.0	100.0
0.2	49.3	95.4	92.9	100.0	94.3	100.0	100.0
0.1	52.1	96.1	95.7	100.0	93.9	100.0	100.0
0	74.3	100.0	100.0	100.0	100.0	100.0	100.0

Discussion: As shown in Table 6.1, The proposed ANF performs consistently higher and more stable compared to other methods with varying rates of occlusion. The differences in ANF-SRC vs. (PCA-R)SRC, ANF-LDA vs. (PCA-R)LDA, and ANF-PCA vs. (PCA-R)PCA and SSRC clearly show the improvement of the proposed approach over the existing methods. The proposed system can be used as an effective nonlinear reconstruction framework when the occlusion rate is not known prior to the reconstruction process. Furthermore, ANF-LDA and ANF-SRC perform

better in comparison to ANF-PCA, showing the discriminative-ness of the reconstructed frontal face image.

6.3.1.2 With Various Occlusion Orientation Experiments

To show the effectiveness of the proposed approach on robust face recognition against various occlusion patterns, we evaluate system against different occlusion orientations as shown in Figure 6.8. We utilize the same training and validation data splits as described in Section 6.3.1.1 to train the system and testing dataset consists of 2 randomly chosen frontal face images per gallery dataset and 1 frontal face image for the probe from the remaining identities (70). The occlusion orientations as shown in Figure 6.8 are then applied to the probe image to evaluates the system’s generalization ability on different occlusion orientations that were unforeseen in the training stage.

Table 6.2: Accuracy on Cross Identity with various occlusion orientations.

Method	Occlusion Orientation Alias (Figure 6.8)					
	$d1$	$d2$	$d3$	$d4$	$d5$	$d1_2$
ANF	100	100	100	100	100	99.2

Discussion: As shown in Table 6.2, the proposed ANF performs consistently high enough as expected and reflects the effectiveness of the progressive occlusion removal strategy. Due to the training that is performed over possible occlusion orientations with varying magnitude (as shown in Figure 6.9), the proposed approach could successfully address other occlusion patterns such as shown in Figure 6.8 that were unseen in the training stage. Thus the proposed system can be used as an effective nonlinear reconstruction framework when the occlusion rate and the orientation are pre-unknown to the reconstruction process.

6.3.1.3 Isolated Database Experiment (real occlusion)

In this experiment, training and testing were also performed on the same database with real occlusions. AR (Martínez and Benavente, 1998) and Curtin (Li *et al.*, 2013a) databases were used in isolation for this experiment. Two types of occlusion exist in these databases. The AR database contains face images occluded by scarf (100 identities) while the Curtin database contains face images occluded by hand (52 identities). These occlusions are real (not simulated) occlusions. To evaluate the performance of the proposed framework on the AR and Curtin datasets in isolation, an additional test case is performed in addition to the 6 simulated test cases as described in Section 6.3.1.1. We utilize images with indexes 1, 3, 4, 5, 6 of each identity (AR-Figure 6.11, Curtin-

Figure 6.12) as the training dataset. In the testing phase, we choose the frontal non-occluded face images as the gallery dataset while the occluded face images are chosen as the probe dataset (AR- Figure 6.11, Curtin-Figure 6.12). For the test cases with simulated occlusion, we utilized images at index 1, 3, 4, 5, 6 for the gallery and the image at index 2 as the probe image. For the test case with real occlusion, we use images at index 1, 2, 3, 4, 5, 6 for the gallery. For the experiment on the AR dataset, images at index 7, 8 are utilized as the probes (Figure 6.11). For the experiment on the Curtin dataset, the image with index 7 (the only frontal face with occlusion) is utilized as the probe (Figure 6.12). The experimental results on the AR and Curtin datasets are summarized in Table 6.3 and Table 6.4 respectively.

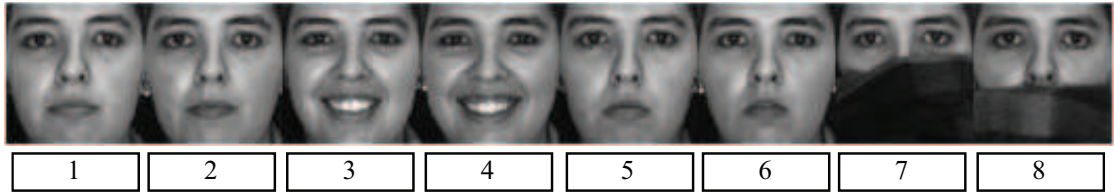


Figure 6.11: Face image subset of the AR database including real occlusion.



Figure 6.12: Face image subset of Curtin database including real occlusion.

Table 6.3: Accuracy on the AR Database.

Occlusion	PCA-R			SSRC	ISDAE	ANF		
	PCA	LDA	SRC			PCA	LDA	SRC
Scarf	23.2	76.7	54.3	89.9	93.8	83.8	94.9	96.8
0.5	31.3	83.8	52.5	91.2	91.1	75.7	96.0	95.5
0.4	33.3	90.9	65.7	93.2	92.1	92.9	98.0	99.0
0.3	34.3	92.1	75.7	94.6	94.0	93.9	98.0	99.0
0.2	75.7	94.3	87.8	96.3	96.0	93.9	98.0	99.0
0.1	82.4	96.2	94.9	97.4	99.2	94.9	98.0	99.0
0.0	87.3	98.0	98.0	98.0	100.0	94.9	100.0	100.0

Discussion: As shown in Table 6.3 and Table 6.4, ANF performs consistently good across different occlusions while outperforming the existing methods. This shows the robust quality of the reconstruction process for different occlusions of face images that are acquired in the same environmental conditions. As mentioned above, the different performances of ANF-LDA, ANF-SRC

Table 6.4: Accuracy on the Curtin Database.

Occlusion	PCA-R			SSRC	ANF		
	PCA	LDA	SRC		PCA	LDA	SRC
Hand	55.8	84.6	57.7	66.7	86.5	91.4	92.3
0.5	87.5	90.1	66.2	93.4	97.3	94.3	96.4
0.4	93.6	92.5	73.7	96.7	97.3	95.7	96.2
0.3	97.4	94.2	95.5	99.0	97.3	99.3	99.6
0.2	97.4	94.9	96.2	99.3	99.4	100.0	100.0
0.1	98.1	97.4	98.1	99.6	99.4	100.0	100.0
0.0	100.0	100.0	98.1	100.0	100.0	100.0	100.0

vs. ANF-PCA demonstrates the discriminant quality of the reconstructed face via the proposed approach as expected. We include ISDAE (Zhang *et al.*, 2013) performance in the AR database experiment to cross compare the proposed autoencoder strategy with their method. The proposed method performs consistently high in both simulated and real occlusion. Furthermore, in the experiment with the AR database, SSRC utilizes one of the scarfed faces for each identity for occlusion dictionary learning (Ou *et al.*, 2014). In contrary, the proposed approach does not require any prior information on occlusions, yet still performs better compared to SSRC. Note that in the experiment with the Curtin database using SSRC, no real occluded faces are used in occlusion dictionary learning. Simulated occlusions are used instead, thus resulting in a drop in classification accuracy.

6.3.1.4 Cross Database Experiment

In order to further demonstrate the robustness of the proposed approach, we apply ANF on two mutually exclusive databases that were acquired in different environmental conditions. The MultiPIE (Gross *et al.*, 2010) database is utilized for training and the AR database is used for testing in this experiment. The same setting as described in Section 6.3.1.1 is followed in the training phase. The testing is conducted on the AR (Martínez and Benavente, 1998) dataset.

The AR database consists of over 3,200 color frontal face images of 126 subjects. Greyscale images of 100 subjects are used in our experiment. As shown in Figure 6.11, 8 frontal views with different expressions and occlusions (scarf) for each subject is considered in the testing phase. Unlike MultiPie, the AR dataset contains face images (occluded by a scarf) with real occlusions (not simulated). Therefore, an additional test case is performed in addition to the 6 simulated test cases described in Section 6.3.1.1. For the additional test case, the frontal non-occluded face

images are used as the gallery dataset while the 2 face images with the scarf are used as the probes (Figure 6.11). For the test cases with simulated occlusions, images with indexes 1, 3, 4, 5, 6 are used as the gallery images and the image with index 2 is used as the probe image (Figure 6.11). For the test case with real occlusions, we use images with indexes 1, 2, 3, 4, 5, 6 for the gallery and the images with indexes 7, 8 as probes (Figure 6.11). The results are summarized in Table 6.5.

Table 6.5: Accuracy on Cross Database.

Occlusion	PCA-R			SSRC	ANF		
	PCA	LDA	SRC		PCA	LDA	SRC
Scarf	25.0	57.1	44.9	70.1	57.1	75.3	75.8
0.5	25.0	63.7	57.3	82.3	58.3	87.9	87.9
0.4	30.6	70.5	75.3	87.9	64.7	92.9	89.9
0.3	38.9	72.5	84.8	89.9	66.0	92.9	97.2
0.2	56.9	74.5	86.1	93.7	69.7	96.5	98.2
0.1	60.6	78.2	86.4	94.3	85.9	98.2	99.0
0.0	91.9	94.9	96.0	96.5	88.0	99.0	99.0

Discussion: As shown in Table 6.5, ANF depicts a very strong generalization ability towards unseen subjects and expressions in training, even if the training and testing images are taken in different environmental settings. It consistently outperforms other methods, showing the true invariability towards reconstructing occluded face regions. Furthermore, ANF-LDA vs. ANF-PCA performance rates indicates the discriminant quality of the reconstructed face via the proposed approach. It can be observed that some scarf faces in AR occlude more than 50% of the face (e.g., Image 7 in Figure 5). Such occurrences affect the performance of the reconstruction process since the proposed approach is trained for a maximum of 0.5 occlusion rate. The impact of the limit is obvious in the recognition results as shown in (Table 6.5) in the rows that correspond to "Scarf" and 0.5 occlusion rate.

6.4 Summary

In this chapter, we focus on generic design principals for a complex deep learning system design for sub-divide the complex tasks into more simpler and tractable objectives. It is often beneficial when a typical deep learning network could not handle the global complexity of the problem as a whole. Also, it is highly useful in terms of computational efficiency, performance, and implementational aspects. We discuss an effective deep learning system design where the AutoDNet can be utilized for the sub-objectives defined in a given complex problem. The proposed deep model

fusion system design is useful in learning complex non-linearities exist in a problem while fusing the features learned at the end.

We show that a carefully followed deep model fusion system design can be effectively utilized to tackle a complex problem in computer vision which is face recognition against occlusion that occurs in various direction and magnitudes. The main objective of the task is further divided into sub-objectives where each sub-objective was addressed with the AutoDNet models proposed in the previous chapter. It eliminates the occlusion progressively for reconstructing the original (non-occluded) face images step by step. As shown in the experiments, the proposed system improved face recognition performance significantly compared to the existing state-of-the-art methods. The proposed system performs consistently well and stable even when large occlusions are present.

The proposed design scheme is discussed with respect to the computer vision domain due to the nature of the problem definition. The design scheme can be extended to other domains such as civil engineering, where the problem needs to be redefined in a way that the global complexity can be divided into simpler and tractable goals. Currently, we utilize the mode shapes and the frequencies to describe the structural properties while there also exist some other descriptors to describe the structural properties. Each of these descriptors can be processed via a set of AutoDNet frameworks and fused them together to produce the stiffness reductions. This way it could be able to characterize the structure for damage identification effectively. We will investigate such applications in the near future.

Chapter 7

Conclusions and Future Directions

The primary focus of this thesis is the study of novel deep learning techniques that can be utilized both in computer vision and civil engineering domain generically. The proposed techniques are based on the basic building block of deep learning which is autoencoders, described in Section 2.4.1. Our main contributions reside in the context of the generality of such novel autoencoder based frameworks while improving their accuracy and efficiency in active feature learning, classification and regression tasks. A typical discriminant analysis was taken a step further with deep learning to perform non-linear discriminant analysis in end-to-end fashion. Lastly, we integrate all the proposed frameworks into a complex deep learning system design to solve complex problems efficiently via sub-dividing the global objective into many tractable sub-objectives.

The research begun with an introduction to a carefully designed autoencoder based deep learning framework named AutoNet that can generically address machine learning problems in vastly different problem domains. It is discussed in details in Chapter 3. The notions behind the proposed AutoNet framework lead to broad applicability and easy adaptivity aspects thus potentially providing better feature learning in the respective domains. Performance evaluations are conducted in both the computer vision and civil engineering domains with some state-of-the-art approaches based on the classification accuracy and metrics like regression value (R-value). The experiment results reveal a significant overall improvement by our proposed framework thus building a strong foundation towards learning useful features generically. Considering the significant performance improvement of the proposed framework against the state-of-the-art methods and its proven adaptivity toward vastly different problem domains we conclude that the novel AutoNet framework can be efficiently utilized to perform both classification and regression tasks in general with reduced complexity and ease of training.

We then proposed an extension to the AutoNet framework to perform robust feature learning when various types of noise in the data are considered. As AutoNet can only perform in cases where noise in data is not significant, the necessity to work under various noise effects drawn our attention towards introducing an extended framework named AutoDNet. AutoDNet was also utilized both in computer vision and civil engineering domain to exploit its potential towards dealing with the additional complexity that arose from the noise in the data. The impressive de-noising capability of the proposed AutoDNet framework in both classification and regression

contexts were observed during our experiments that were formed with different settings across varying types and magnitudes of noise. Furthermore, special attention is given to the effects of enforcing sparsity via the proposed extended framework (SAF), and the experiments demonstrated it is more robust and stable in comparison to the non-sparse AutoDNet. Overall, the significant performance figures of the proposed SAF against the state-of-the-art methods conclude that it is efficient in feature learning under varying degree of noise thus could be effectively utilized with noisy data acquired directory from sensors.

The central focus of attention was then changed to discriminant analysis in Chapter 5 where we proposed a novel deep discriminant analysis (DDA) framework based on AutoDNet framework introduced previously. Furthermore, we present two novel cost formulations associated with discriminant analysis namely: non-linear discriminant error criterion and discriminant co-entropy (DCC) error criterion where latter is the robust variant of the former cost model. Both of these cost formulations could easily be embedded into deep networks replacing the typical squared error criterion to benefit from non-linear feature extractions via the deep structures, especially in a classification context. The proposed DDA framework is a hybrid architecture which combines the non-linear discriminant error criterion and AutoDNet (the deep learning framework introduced previously) to learn non-linear representations of data to a latent space where data can be linearly separable. Hence it can be trained in an end-to-end fashion. Experiments show that the proposed DDA framework is able to learn effective features in various face related problem thus making it highly efficient in utilizing the class information of the data. Additionally, we evaluate the effectiveness of the DCC cost formulation in the classification context when there exist large outliers in data. The experiments conducted on various image databases with corrupted images show that DCC cost formulation is resilient to outliers and performs better and stable for classification related tasks in general.

Lastly, we discuss highly efficient deep learning system design to divide a complex problem into more straightforward tasks further while utilizing the frameworks proposed in previous chapters. In this way, the complicated global non-linearity involved in a problem can be divided into pieces of more tractable objectives thus can be modeled by multiple instances of the previously defined frameworks to work in harmony on specific simpler goals. A complex problem in the computer vision domain is considered that would be hard to solve if a single deep network is utilized in isolation. We propose a deep fusion based system design that utilizes a line of deep models to effectively address the complications in a complex problem domain. The experiments show the superiority of the proposed system against the existing state-of-the-art methods due to the efficiency of the proposed system design. The evaluation was performed with both genuine and simulated added complexities to compare and contrast the robustness of the system. The proposed system performs consistently well and stable. The experiments prove that an efficient system design for a complex problem can reveal the full potential of the proposed frameworks while

significantly improving the performance.

7.1 Future Study

Despite the highly satisfactory performances achieved by all of our proposed frameworks, the following possible problems are identified to be solved in the near future.

- The proposed AutoNet framework in Chapter 3 consists of two main components where dimensionality reduction and relationship learning are the main objectives. Currently we chose the simplest deep learning model (autoencoder) as the generic building block in our framework. We will explore the usage of convolutional autoencoders in the framework.
- We believe that the proposed AutoDNet in Chapter 4 is a complete framework for dealing with various types of noise. However further investigations are deserved on noise removal strategies via enforcing sparsity. Robust activation functions could also contribute towards reducing the noise in the process of training the whole framework in an end-to-end fashion.
- It is shown that deep discriminant analysis is an effective way to discriminate information that belongs to different classes. We also mention an alternative cost formulation to perform robust discriminant analysis (DCC). Something worth further investigation is the deep latent space that would be formed with DCC embedded into a deep structure of non-linear mappings. New frontiers need to be drawn towards defining the SHM problem to incorporate class information in the problem domain, which will enable utilizing the proposed outlier robust discriminant analysis methods for applications in SHM domain.
- A further exploration on complex deep learning system designs can be performed to address extreme complexities that exist in vastly different domains efficiently. The work that has been completed for face recognition against occlusion can be extended by incorporating more discriminative information into the design of the framework for better performances with extreme occlusions (more than 50%). Parallel architecture designs could also be considered as another system design alternative to address complex problems. A parallel architecture design could also be beneficial in incorporating data from various sources in describing the properties of a structure in the SHM problem domain.
- Last but not least, our proposed approaches are based on the autoencoder model which is, in fact, the simplest deep learning network. Different deep network architectures can be explored with new optimization strategies and activation functions. New strategies can also

be investigated to learn many levels of abstractions via deep learning networks to tackle the complex non-linearities involved in widely different problem domains.

Bibliography

- Ahonen, T., Hadid, A., and Pietikäinen, M. (2004). Face recognition with local binary patterns. *European Conference on Computer Vision*, pages 469–481.
- a.K. Qin, Suganthan, P., and Loog, M. (2006). Generalized null space uncorrelated Fisher discriminant analysis for linear dimensionality reduction. *Pattern Recognition*, **39**(9), 1805–1808.
- Alrjebi, M. M., Liu, W., and Li, L. (2016). Face recognition against pose variations using multi-resolution multiple colour fusion. *International Journal of Machine Intelligence and Sensory Signal Processing*, **1**(4), 304–320.
- Alrjebi, M. M., Pathirage, N., Liu, W., and Li, L. (2017). Face recognition against occlusions via colour fusion using 2d-mcf model and src. *Pattern Recognition Letters*.
- An, G. Y. and Ruan, Q. (2006). Novel mathematical model for enhanced fisher’s linear discriminant and its application to face recognition. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 2, pages 524–527. IEEE.
- Arel, I., Rose, D. C., and Karnowski, T. P. (2010). Deep machine learning-a new frontier in artificial intelligence research [research frontier]. *IEEE computational intelligence magazine*, **5**(4), 13–18.
- Asthana, A., Marks, T. K., Jones, M. J., Tieu, K. H., and Rohith, M. (2011). Fully automatic pose-invariant face recognition via 3D pose normalization. In *2011 International Conference on Computer Vision*, pages 937–944.
- B. Alacam, B. Yazici, N. B. (2012). Fisher discriminant analysis with kernels. *Image Processing, IEEE Transactions on*, **1**(1), 1.
- Bakhary, N., Hao, H., and Deeks, A. J. (2007). Damage detection using artificial neural network with consideration of uncertainties. *Engineering Structures*, **29**(11), 2806–2815.
- Belhumeur, P. N., Hespanha, J. P., and Kriegman, D. J. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**(7), 711–720.
- Bengio, Y. (2009). Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, **2**(1), 1–127.
- Bengio, Y. (2011). *Unsupervised and transfer learning challenges in machine learning*, volume 7.

- Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade*, pages 437–478. Springer.
- Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, **5**(2), 157–166.
- Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. (2007a). Greedy layer-wise training of deep networks. In *Advances in neural information processing systems*, pages 153–160.
- Bengio, Y., LeCun, Y., and Others (2007b). Scaling learning algorithms towards AI. *Large-scale kernel machines*, **34**(5), 1–41.
- Blumer, A., Ehrenfeucht, A., Haussler, D., and Warmuth, M. K. (1989). Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, **36**(4), 929–965.
- Boureau, Y.-l., Cun, Y. L., and Others (2008). Sparse feature learning for deep belief networks. In *Advances in neural information processing systems*, pages 1185–1192.
- Brownjohn, J. M. W. (2007). Structural health monitoring of civil infrastructure. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, **365**(1851), 589–622.
- Candes, E. J. and Tao, T. (2006). Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory*, **52**(12), 5406–5425.
- Candès, E. J., Romberg, J. K., and Tao, T. (2006). Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, **59**(8), 1207–1223.
- Chen, L. F., Liao, H. Y. M., Ko, M. T., Lin, J. C., and Yu, G. J. (2000). New LDA-based face recognition system which can solve the small sample size problem. *Pattern Recognition*, **33**(10), 1713–1726.
- Cootes, T. F., Edwards, G. J., and Taylor, C. J. (2001). Active appearance models. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6), 681–685.
- Dai, D. Q. and Yuen, P. C. (2007). Face recognition by regularized discriminant analysis. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, **37**(4), 1080–1085.
- Ding, Z. H., Yao, R. Z., Huang, J. L., Huang, M., and Lu, Z. R. (2017). Structural damage detection based on residual force vector and imperialist competitive algorithm. *Structural Engineering and Mechanics*, **62**(6), 709–717.
- Donoho, D. L. (2006). For most large underdetermined systems of linear equations the minimal 1-norm solution is also the sparsest solution. *Communications on Pure and Applied Mathematics*, **59**(6), 797–829.

- Drira, H., Ben Amor, B., Srivastava, A., Daoudi, M., and Slama, R. (2013). 3D Face recognition under expressions, occlusions, and pose variations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **35**(9), 2270–2283.
- Duan, J., Pan, Z., Zhang, B., Liu, W., and Tai, X.-C. (2015). Fast algorithm for color texture image inpainting using the non-local ctv model. *Journal of Global Optimization*, **62**(4), 853–876.
- Duda, R. O., Hart, P. E., and Stork, D. G. (2000). *Pattern Classification (2Nd Edition)*. Wiley-Interscience, New York, NY, USA.
- Erhan, D., Manzagol, P.-A., Bengio, Y., Bengio, S., and Vincent, P. (2009). The Difficulty of Training Deep Architectures and the Effect of Unsupervised Pre-Training. *AISTATS*, **5**, 153–160.
- Erhan, D., Courville, A., and Vincent, P. (2010). Why Does Unsupervised Pre-training Help Deep Learning ? *Journal of Machine Learning Research*, **11**, 625–660.
- Fidler, S., Skocaj, D., and Leonardis, A. (2006). Combining reconstructive and discriminative subspace methods for robust classification and regression by subsampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**(3), 337–350.
- Fisher, R. A. (1938). The Statistical Utilization of Multiple Measurements. *Annals of Eugenics*, **8**(4), 376–386.
- Fletcher, R. and Reeves, C. M. (1964). Function minimization by conjugate gradients. *The Computer Journal*, **7**(2), 149–154.
- Florin, E., Gross, J., Pfeifer, J., Fink, G. R., Timmermann, L., HOTELLING, H., Ewald, A., Marzetti, L., Zappasodi, F., Meinecke, F. C., Nolte, G., Fries, P., Crick, F., Koch, C., Siegel, M., Donner, T. H., Engel, A. K., Biswal, B., Yetkin, F. Z., Haughton, V. M., and Hyde, J. S. (2012). Relations Between Two Sets of Variates.
- Friedman, J. H. (1989). Regularized discriminant analysis. *Journal of the American Statistical Association*, **84**(405), 165–175.
- Friswell, M. and Mottershead, J. E. (2013). *Finite element model updating in structural dynamics*, volume 38. Springer Science & Business Media.
- Fukunaga, K. (2013). *Introduction to statistical pattern recognition*. Academic press.
- Galton, F. (1889). Personal identification and description. *Journal of Anthropological Institute of Great Britain and Ireland*, pages 177–191.
- Gan, M., Wang, C., and Others (2016). Construction of hierarchical diagnosis network based on deep learning and its application in the fault pattern recognition of rolling element bearings. *Mechanical Systems and Signal Processing*, **72**, 92–104.

- Gao, W., Cao, B., Shan, S., Chen, X., Zhou, D., Zhang, X., and Zhao, D. (2008). The cas-peal large-scale chinese face database and baseline evaluations. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, **38**(1), 149–161.
- Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014a). Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 580–587.
- Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014b). Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 580–587.
- Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep Sparse Rectifier Neural Networks. In *Aistats*, volume 15, page 275.
- Gross, R. (2010). The CMU Multi-PIE Face Database. <http://www.cs.cmu.edu/afs/cs/project/PIE/MultiPie/Multi-Pie/Home.html>.
- Gross, R., Matthews, I., Cohn, J., Kanade, T., and Baker, S. (2010). Multi-pie. *Image Vision Computing*, **28**(5), 807–813.
- Hagan, M. (2007). Neural Network Design. *Network*, **4120**, 45308–45308.
- Hao, H. and Xia, Y. (2002). Vibration-based damage detection of structures by genetic algorithm. *Journal of computing in civil engineering*, **16**(3), 222–229.
- Hastie, T., Tibshirani, R., and Buja, A. (1994). Flexible discriminant analysis by optimal scoring. *Journal of the American Statistical Association*, **89**(428), 1255–1270.
- Hastie, T., Buja, A., and Tibshirani, R. (1995). Penalized Discriminant Analysis.
- He, R., Hu, B.-G., Zheng, W.-S., and Kong, X.-W. (2011). Robust principal component analysis based on maximum correntropy criterion. *IEEE Transactions on Image Processing*, **20**(6), 1485–1494.
- He, X., Yan, S., Hu, Y., Niyogi, P., and Zhang, H.-J. (2005). Face recognition using laplacianfaces. *IEEE transactions on pattern analysis and machine intelligence*, **27**(3), 328–340.
- Heo, J., Kong, S. G., Abidi, B. R., Abidi, M., *et al.* (2004). Fusion of visual and thermal signatures with eyeglass removal for robust face recognition. In *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on*, pages 122–122. IEEE.
- Hinton, G. E. (2012). A practical guide to training restricted boltzmann machines. In *Neural networks: Tricks of the trade*, pages 599–619. Springer.

- Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, **313**(5786), 504–507.
- Hinton, G. E., Osindero, S., and Teh, Y.-W. (2006). A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, **18**(7), 1527–1554.
- Hochreiter, S., Bengio, Y., Frasconi, P., and Schmidhuber, J. (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.
- Huber, P. J. (2011). *Robust statistics*. Springer.
- Hwang, B.-W. and Lee, S.-W. (2003). Reconstruction of partially damaged face images based on a morphable face model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **25**(3), 365–372.
- Iglesias, J. E., De Bruijne, M., Loog, M., Lauze, F., and Nielsen, M. (2007). A family of principal component analyses for dealing with outliers. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 178–185. Springer.
- Jeong, K. H., Liu, W., Han, S., Hasanbelliu, E., and Principe, J. C. (2009). The correntropy MACE filter. *Pattern Recognition*, **42**(5), 871–885.
- Jia, F., Lei, Y., Lin, J., Zhou, X., and Lu, N. (2016). Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data. *Mechanical Systems and Signal Processing*, **72**, 303–315.
- Jiang, X., Binkert, M., Achermann, B., and Bunke, H. (2000). Towards detection of glasses in facial images. *Pattern Analysis & Applications*, **3**(1), 9–18.
- Jiang, X., Mandal, B., and Kot, A. (2008). Eigenfeature regularization and extraction in face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **30**(3), 383–394.
- Jing, Z., Mariani, R., and Wang, J. (2000). Glasses detection for face recognition using bayes rules. In *Advances in Multimodal InterfacesICMI 2000*, pages 127–134. Springer.
- Kan, M., Shan, S., Chang, H., and Chen, X. (2014). Stacked progressive auto-encoders (spae) for face recognition across poses. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1883–1890. IEEE.
- Karami, E., Prasad, S., and Shehata, M. S. (2017). Image matching using sift, surf, BRIEF and ORB: performance comparison for distorted images. *CoRR*, **abs/1710.02726**.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc.

- Kumar, N., Agrawal, R., and Jaiswal, A. (2014). A comparative study of linear discriminant and linear regression based methods for expression invariant face recognition. In *Advances in Signal Processing and Intelligent Recognition Systems*, pages 23–32. Springer.
- Kwak, N. (2008). Principal component analysis based on l1-norm maximization. *IEEE transactions on pattern analysis and machine intelligence*, **30**(9), 1672–1680.
- Larochelle, H., Bengio, Y., Louradour, J., and Lamblin, P. (2009). Exploring Strategies for Training Deep Neural Networks. *Journal of Machine Learning Research*, **1**, 1–40.
- Le, Q. V., Ranzato, M. A., Monga, R., Devin, M., Chen, K., Corrado, G. S., Dean, J., and Ng, A. Y. (2013). Building High-level Features Using Large Scale Unsupervised Learning. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8595–8598.
- LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, **86**(11), 2278–2324.
- Lee, J. J., Lee, J. W., Yi, J. H., Yun, C. B., and Jung, H. Y. (2005). Neural networks-based damage detection for bridges considering errors in baseline finite element models. *Journal of Sound and Vibration*, **280**(3), 555–578.
- Li, B. Y., Mian, A. S., Liu, W., and Krishna, A. (2013a). Using kinect for face recognition under varying poses, expressions, illumination and disguise. In *Applications of Computer Vision (WACV), 2013 IEEE Workshop on*, pages 186–192. IEEE.
- Li, B. Y., Mian, A. S., Liu, W., and Krishna, A. (2013b). Using Kinect for face recognition under varying poses, expressions, illumination and disguise. In *Proceedings of IEEE Workshop on Applications of Computer Vision*, pages 186–192.
- Li, J. and Hao, H. (2014). Substructure damage identification based on wavelet-domain response reconstruction. *Structural Health Monitoring*, **13**(4), 389–405.
- Li, J. and Hao, H. (2016). A review of recent research advances on structural health monitoring in Western Australia. *Structural Monitoring and Maintenance*, **3**(1), 33–49.
- Li, J., Dackermann, U., Xu, Y.-L., and Samali, B. (2011). Damage identification in civil engineering structures utilizing PCA-compressed residual frequency response functions and neural network ensembles. *Structural Control and Health Monitoring*, **18**(2), 207–226.
- Li, J., Law, S. S., and Ding, Y. (2012a). Substructure damage identification based on response reconstruction in frequency domain and model updating. *Engineering Structures*, **41**, 270–284.
- Li, J., Law, S. S., and Hao, H. (2013c). Improved damage identification in bridge structures subject to moving loads: Numerical and experimental studies. *International Journal of Mechanical Sciences*, **74**, 99–111.

- Li, S., Liu, X., Chai, X., and Zhang, H. (2012b). Morphable displacement field based image matching for face recognition across pose. *Eccv 2012*, pages 102–115.
- Li, S. Z., Hou, X. W., Zhang, H. J., and Cheng, Q. S. (2001). Learning spatially localized, parts-based representation. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE.
- Li, X., Hu, W., Wang, H., and Zhang, Z. (2010). Linear discriminant analysis using rotational invariant L1norm. *Neurocomputing*, **73**(13-15), 2571–2579.
- Li, Y. and Feng, J. (2013). Reconstruction based face occlusion elimination for recognition. *Neurocomputing*, **101**, 68–72.
- Liang, A. (2017). *Face Image Retrieval with Landmark Detection and Semantic Concepts Extraction*. Ph.D. thesis, Curtin University.
- Liang, A., Liu, W., Li, L., Farid, M. R., and Le, V. (2014). Accurate facial landmarks detection for frontal faces with extended tree-structured models. In *Proceedings - International Conference on Pattern Recognition*, pages 538–543.
- Liang, A., Pathirage, C. S. N., Wang, C., Liu, W., Li, L., and Duan, J. (2016). Face Recognition Despite Wearing Glasses. In *2015 International Conference on Digital Image Computing: Techniques and Applications, DICTA 2015*.
- Liu, K., Cheng, Y. Q., and Yang, J. Y. (1992). A generalized optimal set of discriminant vectors. *Pattern Recognition*, **25**(7), 731–739.
- Liu, W., Pokharel, P. P., and Príncipe, J. C. (2007). Correntropy: properties and applications in non-gaussian signal processing. *IEEE Transactions on Signal Processing*, **55**(11), 5286–5298.
- Loog, M., Duin, R. P., and Haeb-Umbach, R. (2001). Multiclass linear dimension reduction by weighted pairwise Fisher criteria. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **23**(7), 762–766.
- Lu, J., Plataniotis, K. N., and Venetsanopoulos, A. N. (2003). Face recognition using LDA-based algorithms. *IEEE Transactions on Neural Networks*, **14**(1), 195–200.
- Lu, Z. R. and Wang, L. (2017). An enhanced response sensitivity approach for structural damage identification: convergence and performance. *International Journal for Numerical Methods in Engineering*, **111**(13), 1231–1251.
- Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, **9**(Nov), 2579–2605.

- Martínez, A. and Benavente, R. (1998). The ar face database. Technical Report 24, Computer Vision Center, Bellatera. Cites in Scholar Google: <http://scholar.google.com/scholar?hl=en&lr=&client=firefox-a&cites=1504264687621469812>.
- Martínez, A. M. (1998). AR face database. <http://www2.ece.ohio-state.edu/~aleix/ARdatabase.html>.
- McLachlan, G. J. (2004). *Discriminant analysis and statistical pattern recognition*.
- Mian, A. S. (2013). Databases. <http://staffhome.ecm.uwa.edu.au/~00053650/databases.html>.
- Møller, M. F. (1993). A scaled conjugate gradient algorithm for fast supervised learning. *Neural networks*, **6**(4), 525–533.
- Multiple, A., Analysis, R., and Regression, M. L. (2012). Multiple Linear Regression. *Handbook of Psychology*, pages 1–32.
- Murtaza, M., Sharif, M., Raza, M., and Shah, J. H. (2013). Analysis of Face Recognition under varying facial expression: A survey. *International Arab Journal of Information Technology*, **10**(5).
- Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., and Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. *Journal of Big Data*, **2**(1), 1.
- Naseem, I., Togneri, R., and Bennamoun, M. (2010). Linear regression for face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **32**(11), 2106–2112.
- Ni, P., Xia, Y., Li, J., and Hao, H. (2018). Improved decentralized structural identification with output-only measurements. *Measurement: Journal of the International Measurement Confederation*, **122**, 597–610.
- Ni, Y. Q., Wang, B. S., and Ko, J. M. (2002). Constructing input vectors to neural networks for structural damage identification. *Smart Materials and Structures*, **11**(6), 825.
- Ou, W., You, X., Tao, D., Zhang, P., Tang, Y., and Zhu, Z. (2014). Robust face recognition via occlusion dictionary learning. *Pattern Recognition*, **47**(4), 1559–1572.
- Padil, K. H., Bakhary, N., and Hao, H. (2017). The use of a non-probabilistic artificial neural network to consider uncertainties in vibration-based-damage detection. *Mechanical Systems and Signal Processing*, **83**, 194–209.
- Pathirage, C. S. N., Li, L., Liu, W., and Zhang, M. (2016). Stacked Face De-Noising Auto Encoders for Expression-Robust Face Recognition. In *2015 International Conference on Digital Image Computing: Techniques and Applications, DICTA 2015*.

- Pedagadi, S., Orwell, J., Velastin, S., and Boghossian, B. (2013). Local fisher discriminant analysis for pedestrian re-identification. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3318–3325.
- Pokharel, P. P., Liu, W., and Principe, J. C. (2009). A low complexity robust detector in impulsive noise. *Signal Processing*, **89**(10), 1902–1909.
- Principe, J. C., Xu, D., and Fisher, J. (2000). Information theoretic learning. *Unsupervised adaptive filtering*, **1**, 265–319.
- Ranzato, M., Poultney, C., Chopra, S., and LeCun, Y. (2006). Efficient learning of sparse representations with an energy-based model. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*, pages 1137–1144. MIT Press.
- Ranzato, M. A. (2007). A Unified Energy-Based Framework for Unsupervised Learning. In *Proc. Conference on AI and Statistics (AISTATS)*, **2**, 860–867.
- Rao, C. (1948). The Utilization of Multiple Measurements in Problems of Biological Classification. In *Journal of the Royal Statistical Society. Series B (Methodological)*, volume 10, pages 159–203.
- Rasmussen, C. E. and Ghahramani, Z. (2001). Occam’s Razor. *Advances in Neural Information Processing Systems*, **13**, 294–300.
- Raudys, S. and Duin, R. P. (1998). Expected classification error of the Fisher linear classifier with pseudo-inverse covariance matrix. *Pattern Recognition Letters*, **19**(5-6), 385–392.
- Righi, G., Peissig, J. J., and Tarr, M. J. (2012). Recognizing disguised faces. *Visual Cognition*, **20**(2), 143–169.
- Roweis, S. (1996). Levenberg-Marquardt Optimization. *Notes, University Of Toronto*.
- Saito, Y., Kenmochi, Y., and Kotani, K. (1999). Estimation of eyeglassless facial images using principal component analysis. In *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*, volume 4, pages 197–201. IEEE.
- Samaria, F. S. and Harter, A. C. (1994). Parameterisation of a stochastic model for human face identification. In *Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on*, pages 138–142. IEEE.
- Santamaría, I., Pokharel, P. P., and Principe, J. C. (2006). Generalized correlation function: definition, properties, and application to blind equalization. *IEEE Transactions on Signal Processing*, **54**(6), 2187–2197.
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, **61**, 85–117.

- Scholkopf, B., Smola, A. J., and Müller, K.-R. (2012). Kernel Principal Component Analysis. *Computer Vision And Mathematical Methods In Medical And Biomedical Image Analysis*, **1327**(3), 583–588.
- Shan, S. (2008). CAS-PEAL face database. <http://www.jdl.ac.cn/peal/>.
- Sharma, A. and Jacobs, D. W. (2011). Bypassing synthesis: PLS for face recognition with pose, low-resolution and sketch. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 593–600.
- Sharma, A., Kumar, A., Daume, H., and Jacobs, D. W. (2012). Generalized Multiview Analysis: A discriminative latent space. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2160–2167.
- Sharon, Y., Wright, J., and Ma, Y. (2009). Minimum sum of distances estimator: Robustness and stability. In *Proceedings of the American Control Conference*, pages 524–530.
- Sim, T., Baker, S., and Bsat, M. (2002). The cmu pose, illumination, and expression (pie) database. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pages 46–51. IEEE.
- Subbarao, R. and Meer, P. (2006). Subspace estimation using projection based M-estimators over grassmann manifolds. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 3951 LNCS, pages 301–312.
- Sun, Y., Wang, X., and Tang, X. (2014). Deep Learning Face Representation by Joint Identification-Verification. pages 1–9.
- Tao, D., Tang, X., and Wu, X. (2006). Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**(7), 1088–1099.
- Tibshirani, R. (1996). Regression Selection and Shrinkage via the Lasso.
- Torkkola, K. (2004). Discriminative features for text document classification. *Pattern Analysis and Applications*, **6**(4), 301–308.
- Torrey, L. and Shavlik, J. (2009). Transfer Learning. *Machine Learning*, pages 1–22.
- Tsai, P. and Jan, T. (2005). Expression-Invariant Face Recognition System Using Subspace Model Analysis. *IEEE International Conference on Systems, Man and Cybernetics*, **2**, 1712–1717.
- Turk, M. and Pentland, A. (1991). Eigenfaces for Recognition. *Journal of Cognitive Neuroscience*, **3**(1), 71–86.

- Vapnik, V. N. (1995). The Nature of Statistical Learning Theory.
- Vapnik, V. N. (2006). *Estimation of dependences based on empirical data ; Empirical inference science : afterword of 2006*.
- Vapnik, V. N. and Chervonenkis, A. Y. (2015). On the uniform convergence of relative frequencies of events to their probabilities. In *Measures of Complexity: Festschrift for Alexey Chervonenkis*, pages 11–30.
- Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM.
- Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P.-A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *The Journal of Machine Learning Research*, **11**, 3371–3408.
- Wang, X., Ruan, Q., Jin, Y., and An, G. (2014). Three-dimensional face recognition under expression variation. *Eurasip Journal on Image and Video Processing*, **2014**(1).
- Wang, Y.-K., Jang, J.-H., Tsai, L.-W., and Fan, K.-C. (2010). Improvement of face recognition by eyeglass removal. In *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2010 Sixth International Conference on*, pages 228–231. IEEE.
- Wolpert, D. H. (1996). The Lack of a Priori Distinctions between Learning Algorithms. *Neural Computation*, **8**(7), 1341–1390.
- Wong, W. K. and Zhao, H. (2013). Eyeglasses removal of thermal image based on visible information. *Information Fusion*, **14**(2), 163–176.
- Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S., and Ma, Y. (2009). Robust face recognition via sparse representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **31**(2), 210–227.
- Wu, C., Liu, C., Shum, H.-Y., Xy, Y.-Q., and Zhang, Z. (2004). Automatic eyeglasses removal from face images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **26**(3), 322–336.
- Xia, Y., Hao, H., Brownjohn, J. M. W., and Xia, P.-Q. (2002). Damage identification of structures with uncertain frequency and mode shape data. *Earthquake engineering & structural dynamics*, **31**(5), 1053–1066.
- Xiujuan Chai, Shiguang Shan, Xilin Chen, and Wen Gao (2007). Locally Linear Regression for Pose-Invariant Face Recognition. *IEEE Transactions on Image Processing*, **16**(7), 1716–1725.

- Xu, D. (1999). *Energy, entropy and information potential for neural computation*. Ph.D. thesis, Citeseer.
- Xu, H. J., Ding, Z. H., Lu, Z. R., and Liu, J. K. (2015). Structural damage detection based on Chaotic Artificial Bee Colony algorithm. *Structural Engineering and Mechanics*, **55**(6), 1223–1239.
- Yang, L., Li, C., Han, J., Chen, C., Ye, Q., Zhang, B., Cao, X., and Liu, W. (2017). Image reconstruction via manifold constrained convolutional sparse coding for image sets. *IEEE Journal of Selected Topics in Signal Processing*, **11**(7), 1072–1081.
- Ye, J. and Li, Q. (2005). A two-stage linear discriminant analysis via QR-decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**(6), 929–941.
- Yeung, W. T. and Smith, J. W. (2005). Damage detection in bridges using neural networks for pattern recognition of vibration signatures. *Engineering Structures*, **27**(5), 685–698.
- Yu, H. and Yang, J. (2001). A direct LDA algorithm for high-dimensional data with application to face recognition. *Pattern Recognition*, **34**(10), 2067–2070.
- Yuan, X.-t. and Hu, B.-g. (2009). Robust Feature Extraction via Information Theoretic Learning. *Proceedings of the 26th International Conference on Machine Learning*, (m), 1193–1200.
- Yun, C.-B., Yi, J.-H., and Bahng, E. Y. (2001). Joint damage assessment of framed structures using a neural networks technique. *Engineering structures*, **23**(5), 425–435.
- Zang, C. and Imregun, M. (2001). Structural damage detection using artificial neural networks and measured FRF data reduced via principal component projection. *Journal of Sound and Vibration*, **242**(5), 813–827.
- Zhang, L., Yang, M., and Feng, X. (2011). Sparse representation or collaborative representation: Which helps face recognition? In *Computer vision (ICCV), 2011 IEEE international conference on*, pages 471–478. IEEE.
- Zhang, L., Xiang, T., and Gong, S. (2016). Learning a Discriminative Null Space for Person Re-identification. *Cvpr*, pages 1239–1248.
- Zhang, T., Fang, B., Tang, Y. Y., Shang, Z., and Xu, B. (2010). Generalized discriminant analysis: A matrix exponential approach. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, **40**(1), 186–197.
- Zhang, Y., Liu, R., Zhang, S., and Zhu, M. (2013). Occlusion-robust face recognition using iterative stacked denoising autoencoder. In *International Conference on Neural Information Processing*, pages 352–359. Springer.

- Zhao, W., With, A., Center for Automation Research, U. o. M., Krishnaswamy, A., Chellappa, R., Swets, D. L., and Weng, J. (1998). *Discriminant Analysis of Principal Components of Facial Recognition.*, volume 163.
- Zhou, W. and Kamata, S.-i. (2012). Linear discriminant analysis with maximum correntropy criterion. In *Asian Conference on Computer Vision*, pages 500–511. Springer.
- Zhu, Z., Luo, P., Wang, X., and Tang, X. (2013). Deep learning identity-preserving face space. *Proceedings of the IEEE International Conference on Computer Vision*, pages 113–120.
- Zhu, Z., Luo, P., Wang, X., and Tang, X. (2014). Recover Canonical-View Faces in the Wild with Deep Neural Networks. pages 1–10.
- Zifeng, C., Baowen, X., Weifeng, Z., Dawei, J., and Junling, X. (2007). CLDA: Feature selection for text categorization based on constrained LDA. In *ICSC 2007 International Conference on Semantic Computing*, pages 702–709.

Every reasonable effort has been made to acknowledge the owners of copyright material. I would be pleased to hear from any copyright owner who has been omitted or incorrectly acknowledged.