Beyond PCA: Deep Learning Approaches for Face Modeling and Aging

Chi Nhan Duong

A Thesis

in

The Department

of

Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy (Computer Science) at

Concordia University

Montréal, Québec, Canada

November 2017

Concordia University

School of Graduate Studies

This is to certify that the thesis prepared

By:	**Mr. Chi Nhan Duong**
Entitled:	**Beyond PCA: Deep Learning Approaches for Face Modeling and Aging**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Computer Science)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality. Signed by the Final Examining Committee:

______________________________ Chair
Dr. Hua Ge

______________________________ External Examiner
Dr. B.V.K. Vijaya Kumar

______________________________ Examiner
Dr. Tiberiu Popa

______________________________ Examiner
Dr. Adam Krzyzak

______________________________ Examiner
Dr. Hassan Rivaz

______________________________ Supervisor
Dr. Tien D. Bui

______________________________ Co-supervisor
Dr. Khoa Luu

Approved by ______________________________
Dr. Sudhir Mudur, Chair
Department of Computer Science and Software Engineering

November 09, 2017 ______________________________
Dr. Amir Asif, Dean
Faculty of Engineering and Computer Science

Abstract

Beyond PCA: Deep Learning Approaches for Face Modeling and Aging

Chi Nhan Duong, Ph.D.

Concordia University, 2017

Modeling faces with large variations has been a challenging task in computer vision. These variations such as expressions, poses and occlusions are usually complex and non-linear. Moreover, new facial images also come with their own characteristic artifacts greatly diverse. Therefore, a good face modeling approach needs to be carefully designed for flexibly adapting to these challenging issues. Recently, Deep Learning approach has gained significant attention as one of the emerging research topics in both higher-level representation of data and the distribution of observations. Thanks to the nonlinear structure of deep learning models and the strength of latent variables organized in hidden layers, it can efficiently capture variations and structures in complex data.

Inspired by this motivation, we present two novel approaches, i.e. Deep Appearance Models (DAM) and Robust Deep Appearance Models (RDAM), to accurately capture both shape and texture of face images under large variations. In DAM, three crucial components represented in hierarchical layers are modeled using Deep Boltzmann Machines (DBM) to robustly capture the variations of facial shapes and appearances. DAM has shown its potential in inferencing a representation for new face images under various challenging conditions. An improved version of DAM, named Robust DAM (RDAM), is also introduced to better handle the occluded face areas and, therefore, produces more plausible reconstruction results. These proposed approaches are evaluated in various applications to demonstrate their robustness and capabilities, e.g. facial super-resolution reconstruction, facial off-angle reconstruction, facial occlusion removal and age estimation using challenging face databases: Labeled Face Parts in the Wild (LFPW), Helen and FG-NET. Comparing to classical and other deep learning based approaches, the proposed DAM and RDAM achieve competitive results in those applications, thus this showed their advantages in handling occlusions, facial representation, and reconstruction.

In addition to DAM and RDAM that are mainly used for modeling single facial image, the second part of the thesis focuses on novel deep models, i.e. Temporal Restricted Boltzmann Machines (TRBM) and tractable Temporal Non-volume Preserving (TNVP) approaches, to further model face sequences. By exploiting the additional temporal relationships presented in sequence data, the proposed models have their advantages in predicting the future of a sequence from its past. In the application of face age progression, age regression, and age-invariant face recognition, these models have shown their potential not only in efficiently capturing the non-linear age related variance but also producing a smooth synthesis in age progression across faces. Moreover, the structure of TNVP can be transformed into a deep convolutional network while keeping the advantages of probabilistic models with tractable log-likelihood density estimation. The proposed approach is evaluated in terms of synthesizing age-progressed faces and cross-age face verification. It consistently shows the state-of-the-art results in various face aging databases, i.e. FG-NET, MORPH, our collected large-scale aging database named AginG Faces in the Wild (AGFW), and Cross-Age Celebrity Dataset (CACD). A large-scale face verification on Megaface challenge 1 is also performed to further show the advantages of our proposed approach.

Acknowledgments

I would like to express my gratitude to all those who have supported, influenced and helped me in the process which ultimately resulted in this thesis. First and foremost, I would like to thank my advisors, Prof. Tien D. Bui and Dr. Khoa Luu, for being amazing advisors, sharing your incredible expertise, and providing me invaluable guidance and encouragement. Thank you, Prof. Bui, for strengthening my background in Image Processing and providing me a great balance of guidance, support, and freedom. Dr. Luu, thank you for bringing me to deep learning research field and helping me to approach the problems and solve them in a systematic way. I would also like to thank my committee members, Prof. B.V.K. Vijaya Kumar, Dr. Tiberiu Popa, Prof. Adam Krzyzak, and Dr. Hassan Rivaz for their valuable feedback and suggestions.

I am thankful to Prof. Marios Savvides for his great supports and providing me the opportunity to work in his lab, the Cylab Biometrics Center at Carnegie Mellon University. I have learned and enhanced lots of skills while working in the friendly and supportive environment here. I would like to thank all my colleagues at Concordia University and Cylab Biometrics Center: Kha Gia Quach, Ngan Le and Sekhar Bhagavatula for their fruitful comments and inspiring discussions. Thank you for spending lots of time with me to brainstorm, teamwork, and proof-read my papers. I would like to express my appreciation for Halina Monkiewicz, Tina Yankovich, Brittany Frost for their excellent administrative supports that have provided me more time for my study and research.

Finally, my special thanks go to my parents, Chi Nghia Duong and Thi Kim Hoa Vo, and my wife, Dan Thanh Duong Thi, for your incredible supports, encouragement and love. Thank you for always being by my side and getting me lots of motivations and values that led me to overcome my hard time and find the passion through my PhD.

Contributions of Author

This section presents the list of journal and conference papers completed since I joint Concordia University.

Journal Publications

[1] **Chi Nhan Duong**, Khoa Luu, Kha Gia Quach, Tien D. Bui, "**Deep Appearance Models: A Deep Boltzmann Machine Approach for Face Modeling**", *International Journal of Computer Vision (**IJCV**)*, 2016. **(Under review - 2nd round)** (***Impact factor***: 4.27)

[2] Kha Gia Quach, **Chi Nhan Duong**, Khoa Luu, Tien D. Bui. "**Non-convex Online Robust PCA: Enhance Sparsity via ℓ_p-norm Minimization**". *Computer Vision and Image Understanding (**CVIU**)*, 2017. (***Impact factor***: 1.54)

Conference Publications

[3] **Chi Nhan Duong**, Kha Gia Quach, Khoa Luu, T. Hoang Ngan Le, Marios Savvides, "**Temporal Non-Volume Preserving Approach to Facial Age-Progression and Age-Invariant Face Recognition**", *The IEEE International Conference on Computer Vision (**ICCV**)*, Italy, 2017. (ORAL)

[4] **Chi Nhan Duong**, Khoa Luu, Kha Gia Quach, Tien D. Bui, "**Longitudinal Face Modeling via Temporal Deep Restricted Boltzmann Machines**", *The IEEE International Conference on Computer Vision and Patern Recognition (**CVPR**)*, Las Vegas, 2016, pp. 5272 - 5780. (Acceptance rate 29.9%)

[5] Kha Gia Quach*, **Chi Nhan Duong***, Khoa Luu, Tien D. Bui, "**Robust Deep Appearance**

Models", *The 23rd International Conference on Patern Recognition (**ICPR**)*, Cancun, 2016, pp. 390 - 395. (***equal contribution)**

[6] **Chi Nhan Duong**, Khoa Luu, Kha Gia Quach, Tien D. Bui, "**Beyond Principal Components: Deep Boltzmann Machines for Face Modeling**", *The IEEE International Conference on Computer Vision and Patern Recognition (**CVPR**)*, Boston, 2015, pp. 1786 - 1794. (Acceptance rate 28.4%)

[7] **Chi Nhan Duong**, Kha Gia Quach, Tien D. Bui, "**Are Sparse Representation and Dictionary Learning Good for Handwritten Character Recognition?**", *The 14th International Conference on Frontiers in Handwriting Recognition (**ICFHR**)*, Crete, Greece, 2014, pp. 575 - 580. (ORAL Acceptance rate 20.8%)

[8] Kha Gia Quach, **Chi Nhan Duong**, Khoa Luu, Tien D. Bui, "**Depth-based 3D Hand Pose Tracking**", *The 23rd International Conference on Patern Recognition (**ICPR**)*, Cancun, 2016, pp. 2746 - 2751.

[9] Kha Gia Quach, Khoa Luu, **Chi Nhan Duong**, Tien D. Bui. **Robust ℓ_p-norm Singular Value Decomposition**. *NIPS Workshop on Non-convex Optimization for Machine Learning: Theory and Practice* (**NIPSW**), December 2015.

[10] Kha Gia Quach, **Chi Nhan Duong**, Tien D. Bui, "**Sparse representation and Low-rank approximation for Robust Face Recognition**", *The 22nd International Conference on Pattern Recognition (**ICPR**)*, Stockholm, Sweden, 2014, pp. 1330 - 1335.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Modeling faces with large variations has been a challenging task in computer vision. These variations such as expressions, poses and occlusions are usually complex and non-linear. Moreover, new facial images also come with their own characteristic artifacts that greatly diverse. Therefore, a good face modeling approach needs to be carefully designed for flexibly adapting to these challenging issues. Over the last two decades, the "interpretation through synthesis" approach has become one of the most successful and popular face modeling approaches. This approach aims to "describe" a given face image by generating a new synthesized image similar to it as much as possible. This purpose can be achieved by an optimization process on the appearance parameters of the model based *apriori* on constrained solutions. The subspace model then plays a key role that decides the robustness of the whole system. Therefore, in order to be applicable, it must provide a basis for a broad range of variations that are usually unseen.

Among *classical models*, Active Appearance Models (AAM) can be considered as one of the most successful face interpretation methods. This model was first introduced by Cootes et al. in 1998 [23]. Since then, it has been widely applied in many applications such as face recognition [28], facial expression recognition [115], face tracking [146], emotion classification [71], expressive visual text-to-speech [5] and many other tasks. Although the framework of AAM is general and effective, their generalization ability is still limited especially when dealing with unseen variations. Gross et al. [37] showed that AAM perform well in person-specific cases rather than generic ones. Cootes and Taylor [22] pointed out the problem of the pre-computed Jacobian matrix computed

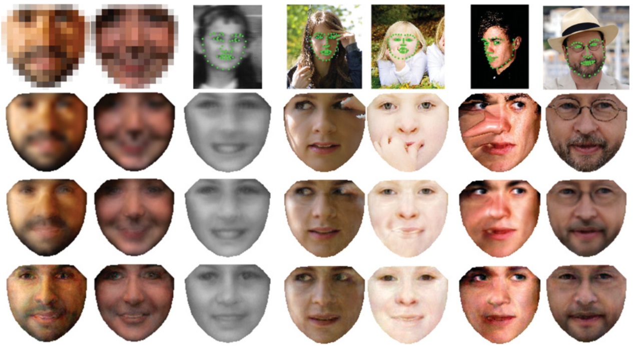


Figure 1.1: An illustration in facial interpretation using the AAM and our DAM approach in real world images, e.g. low resolution, blurred faces, occlusions, off-angle faces, etc. The first row: original images; The second row: shape free images; The third row: facial interpretation using PCA-based AAM; The fourth row: facial interpretation using our proposed DAM approach.

during the training step. Since it is only an approximation for testing image, it may lead to poor convergence when the image is very different from training data. Lighting changes [95] also make AAM difficult to synthesize new images.

To overcome these disadvantages, there have been numerous improvements and adaptations based on the original approach [4, 27, 53, 77]. However, even when these adaptations are taken into account, the capabilities of facial generalization and reconstruction are still highly dependent on the characteristics of training databases. This is because at the heart of AAM, Principal Component Analysis (PCA) is used to provide a subspace to model variations in training data. The limitation of PCA to generalize to illumination and poses, particularly for faces, is very well known. Therefore, it is not surprising that AAM have difficulties in generalizing to new faces under these challenging conditions. On the other hand, the variations in data are not only large but also non-linear. For example, the variations in different facial expressions or poses are non-linear. It apparently violates the linear assumptions of PCA-based models. Thus, single PCA model is unable to interpret the facial variations well. Figure 1.1 presents example faces with various challenging factors, i.e. low-resolution, blurred faces, occlusions, pose faces. The AAM interpretations presented in the third row of the figure have a major negative impact from these wide range of variations.

Recently, *Deep Learning models* such as Deep Boltzmann Machines (DBM) [107] and Convolutional Neural Networks (CNN) [68] have gained significant attention as one of the emerging research topics in both the higher-level representation of data and the distribution of observations.

In the former approach, the deep model is designed following the concepts of probabilistic graphical models. Particularly, in DBM, non-linear latent variables are organized in multiple connected layers in a way that variables in one layer can simultaneously contribute to the probabilities or states of variables in the next layers. Each layer learns a different factor to represent the variations in a given data. Thanks to the nonlinear structure of DBM and the strength of latent variables organized in hidden layers, it efficiently captures variations and structures in complex data that could be higher than second order. Moreover, DBM is shown to be more robust with ambiguous input data [107]. There are some recent works using DBM as prior model [30, 125, 135].

On the other hand, CNN is a biologically-inspired variant of feed-forward artificial neural network where each neuron only responds to a local region, i.e. receptive field, of the visual field. The features extracted from CNN can be also divided into several levels. Features in the first level (i.e. extracted by some first convolutional layers) usually encode simple visual features such as edge, color blobs, etc. In the next level, the extracted features will be the combinations of previous features, i.e. the combinations of edges, the corner. As a result, the more levels a CNN has, the higher-level features can be extracted. In addition, CNN models also enjoy the advantages of tractable back-propagation training process.

Motivating from these approaches, the main aims of this thesis are to exploit the advantages of these two types of deep models, i.e. Deep Boltzmann Machines and Convolutional Neural Networks, for face modeling and aging. The thesis consists of two main streams which focus on modeling (1) single face under large variations, and (2) a face sequence to synthesize the age-progressed faces. In the first stream, a novel deep model, named Deep Appearance Models (DAM), is introduced to overcome the disadvantages of classical linear model such as AAM. This proposed model has shown its potential in both tasks of learning high-level representation and face reconstruction under various challenging conditions. Then an improved version of DAM, named Robust Deep Appearance Models (RDAM), is developed to efficiently handle the occluded face areas, and help to reconstruct more plausible faces.

The second stream of the thesis will concentrate on modeling face sequences by exploiting the temporal relationship between images in these sequences. In particular, a Temporal Restricted Boltzmann Machines (TRBM) based age progression framework is proposed to not only capture the non-linear age related variance of each age group but also be able to embed the aging transformation between age groups. As a result, a smooth synthesis in age progression can be efficiently produced. However, similar to other Boltzmann Machines based approaches, this framework also suffers from the issues of intractable training due to the process of density estimation in probabilistic graphical model. Therefore, in the later part, we further develop a Temporal Non-volume Preserving (TNVP) approach that guarantees a tractable density function, exact inference and evaluation for embedding the feature transformations. Moreover, this structure can be transformed into a deep convolutional network while keeping the advantages of probabilistic models with tractable log-likelihood density estimation.

1.1 Contributions of the Thesis

The main contributions of this thesis are as follows.

(1) We introduce an efficient model, i.e. DAM, that are able to capture the large and non-linear variations presented in face images. Compared to classical models, this model is more advanced in interpreting these variations and show its potential in producing faces with more details.

(2) By extracting high-level representations for both shape and texture of a face, the relationship between them is efficiently exploited in a deeper hidden layer and benefits both reconstruction and discriminative tasks.

(3) We propose a new texture modeling approach on top of DAM structure that is able to distinguish between “good” and “bad” face regions. For example, this new model can take an input face with sunglasses and recover a “clean” face without sunglasses. This modeling step also helps to improve the model fitting process.

(4) In addition to single face modeling, we propose a TRBM based age progression model to embed the temporal relationship between images in a face sequence. Taking the advantages of log-likelihood objective function and avoiding the ℓ_2 reconstruction error during training, the proposed model can synthesize faces with more aging details. Moreover, we also present

a machine learning based approach to learn the aging rules for wrinkle appearance. As a result, our model is more flexible in producing more wrinkle types.

(5) By addressing the intractable issue of RBM model, we propose a novel generative probabilistic models with tractable density function to capture the non-linear age variances. The aging transformation can be effectively modeled using our TNVP. Similar to other probabilistic models, our TNVP is more advanced in term of embedding the complex aging process. Unlike previous aging approaches that suffer from a burdensome preprocessing to produce the dense correspondence between faces, our model is able to synthesize realistic faces given any input face in the wild.

(6) A large-scale aging dataset named AginG Faces in the Wild (AGFW) is collected for analysing the aging effects.

1.2 Summary of remaining chapters

The thesis is organized as follows.

Chapter 2: Background. This chapter provides an overview of classical approach such as Principal Component Analysis (PCA) and Active Appearance Models (AAM); and main theories of Deep learning models such as Restricted Boltzmann Machines (RBM), Deep Boltzmann Machines (DBM), Temporal Restricted Boltzmann Machines, and Convolution Neural Networks.

Chapter 3: Literature review. The first part of this chapter presents a review of single face modeling approaches consisting of AAM based, RBM based, Generative Adversarial Networks based approaches. In the second part, a literature review of longitudinal face modeling approaches is provided.

Chapter 4: Deep Appearance Models for Face Modeling. In this chapter, Deep Appearance Models are introduced for modeling faces under large variations. This model can be considered as an efficient replacement for Active Appearance Models. In contrast to previous models where

DBM is only used as shape prior model or higer-level representation, both shape and texture are modeled using two different DBMs. Further than that, on the top of these two deep models, the higher-level relationships of both shape and texture are exploited in the proposed DAM so that the reconstruction of one can benefit from the information on the other. Three crucial components represented in hierarchical layers are modeled to robustly capture the variations of facial shapes and appearances. DAM is therefore superior to AAM in inferencing a representation for new face images under various challenging conditions.

Chapter 5: Robust Deep Appearance Models for Texture Modeling. This chapter presents the Robust Deep Appearance Models (RDAM) that extends the proposed DAM in its ability of dealing with occluded face regions. In the structure of RDAM, an additional appearance mask is learned and help the DAM to separate corrupted/ occluded pixels in texture modeling process. As a result, those regions are ignored during face reconstruction and model fitting and, therefore, better reconstructed results can be achieved.

Chapter 6: Temporal Restricted Boltzmann Machines for Longitudinal Face Modeling. This chapter introduces a deep model approach for face age progression that can efficiently capture the non-linear aging process and automatically synthesize a series of age-progressed faces in various age ranges. In this approach, we first decompose the long-term age progress into a sequence of short-term changes and model it as a face sequence. The Temporal Restricted Boltzmann Machines based age progression model together with the prototype faces are then constructed to learn the aging transformation between faces in the sequence. In addition, to enhance the wrinkles of faces in the later age ranges, the wrinkle models are further constructed using Restricted Boltzmann Machines to capture their variations in different facial regions. The geometry constraints are also taken into account in the last step for more consistent age-progressed results.

Chapter 7: Temporal Non-volume Preserving Approach. Addressing a limitation of intractable learning process of TRBM based model, in this chapter, a novel generative probabilistic model, named Temporal Non-Volume Preserving (TNVP) transformation, is presented to model the facial aging process at each stage. Unlike Generative Adversarial Networks (GANs), which requires an empirical balance threshold, and Restricted Boltzmann Machines (RBM), an intractable model, our proposed TNVP approach guarantees a tractable density function, exact inference and evaluation for embedding the feature transformations between faces in consecutive stages. Our model shows its advantages not only in capturing the non-linear age related variance in each stage but also producing a smooth synthesis in age progression across faces. Our approach can model any face in the wild provided with only four basic landmark points. Moreover, the structure can be transformed into a deep convolutional network while keeping the advantages of probabilistic models with tractable log-likelihood density estimation.

Chapter 8: Experimental Results. In this chapter, all the experimental results to evaluate the four models are presented. To demonstrate their robustness and capabilities, various applications such as facial super-resolution reconstruction, facial off-angle reconstruction or face frontalization, facial occlusion removal, age estimation and age progression have been taken into account.

Chapter 9: Conclusions and Future Work This chapter provides a summary of the thesis' contributions and discussions for possible future developments of the proposed networks.

Chapter 2

Background

This chapter will first present the background materials needed to understand the principles of Active Appearance Models; Boltzmann Machines (BM) and Restricted Boltzmann Machines (RBM); and Convolutional Neural Networks. Then the state-of-the-art training methods and how models can be built from the training data are introduced. Some extensions of RBM to model real-valued data and temporal dependencies in time-series data as well as a Deep Boltzmann Machines that stacks a sequence of RBMs are presented. A comparison between RBM and Convolutional Neural Networks is also provided for better understanding of the two models.

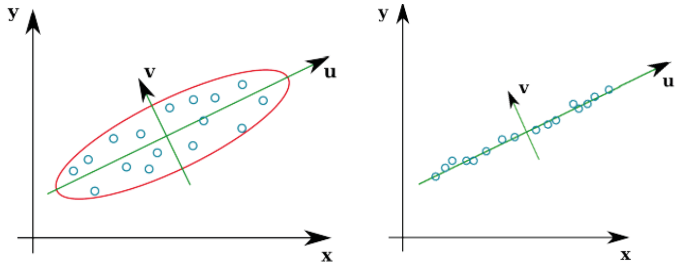
2.1 Active Appearance Models (AAM)

This section briefly reviews the Principal Component Analysis, i.e. the building block of AAM, and two main steps of AAM including modeling and fitting.

2.1.1 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is one of the most common techniques for finding patterns, i.e. low-dimensional representation, of high-dimensional data. This technique has been successfully used for data modeling, compression and visualization for many applications in many fields, e.g., pattern recognition, data compression, image processing, bioinformatics, etc. Given a set of points in n-dimensional space, PCA technique aims to find a d-dimensional linear subspace

($d < n$) that these points mainly lie on this subspace. In other words, PCA tries to find a subspace that captures most of the data variability. For example, given a set of points in two-dimensional space as Figure 2.1(a):



(a) (b)

Figure 2.1: (a) Point set in 2-D space and its principal components; and (b) Projection onto the principal component of the data.

Instead of using the (x, y) coordinate system, we can specify this space by two other orthogonal vectors (u, v) and form a new coordinate system. These two vectors are called principal components and its directions point out how the input data varies. Based on the new coordinate system of the data, the idea of PCA for dimensionality reduction is to approximate the original space by a subspace spanned by d principal components that maximum the variance of the data. For example, in Figure 2.1(b), by projecting all data points onto subspace spanned by u, we can preserve variability of the original data. Therefore, the structure of data can be preserved.

Formally, PCA can be formulated as a statistical problem which is to find principal components (or directions) of a multivariate random variable from sample points $\{\mathbf{x}_i\}$. The d principal components $\mathbf{u}_i \in \mathbb{R}^d$ $(i = 1, \ldots, d)$ is defined as $\mathbf{y}_i = \mathbf{u}_i^T \mathbf{x}$ that maximize the variance of $\mathbf{y}_i$. For example, to find the first principal component, we find a vector u_1^* such that

$$\mathbf{u}_1^* = \arg\max_{\mathbf{u}_1} \text{Var}\left(\mathbf{u}_1^T \mathbf{x}\right) \tag{1}$$

The first d principal components of a multivariate random variable $\mathbf{x}$ are given by the d leading eigenvectors of its covariance matrix $\mathbf{\Sigma}_x = \mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right]$. However, we normally do not know $\mathbf{\Sigma}_x$ and it can only be estimated from n given data points $\mathbf{x}_i$ as follows.

$$\hat{\boldsymbol{\Sigma}}_x = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i\mathbf{x}_i^T = \frac{1}{n}\mathbf{X}\mathbf{X}^T \tag{2}$$

The eigen-vectors of $\hat{\boldsymbol{\Sigma}}_x$ are the "sampled principal components"

$$\hat{\mathbf{y}}_i = \hat{\mathbf{u}}_i^T\mathbf{x} \text{ s.t. } \hat{\boldsymbol{\Sigma}}_x\hat{\mathbf{u}}_i = \lambda\hat{\mathbf{u}}_i \text{ and } \hat{\mathbf{u}}_i^T\hat{\mathbf{u}}_i = \mathbf{1} \tag{3}$$

The matrix $\hat{\boldsymbol{\Sigma}}_x$ containing ordered squared eigenvalues which tell us how much the variability in the data along the corresponding principal components (or directions) is. If $\mathbf{X}$ can be decomposed as $\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ using SVD then we also have the eigenvalue decomposition of covariance matrix as $\mathbf{X}\mathbf{X}^T = \mathbf{U}\boldsymbol{\Sigma}^{\mathbf{2}}\mathbf{U}^T$. If they are ordered, then the first d eigenvectors (corresponding to first d eigenvalues) of $\mathbf{X}\mathbf{X}^T$ will be the first d sample principal components of $\mathbf{X}$.

2.1.2 Active Appearance Models

AAM modeling: The basic AAM [24] employs statistical models to build a unified appearance model describing both shape and texture variation. This appearance model is trained from a set of images with landmarks representing the shape of deformable objects. Let $\mathcal{I} \subset \mathbb{R}^2$ be the image domain and $\mathcal{D} \subset \mathbb{R}^2$ be the texture domain, an image $I(r_{\mathcal{I}})$ is considered as a function of the image domain $\mathcal{I}$, where $(x_I, y_I) = r_{\mathcal{I}} \in \mathcal{I}$. The shape $\mathbf{s} = (r_{\mathcal{D}}^1, ..., r_{\mathcal{D}}^n)$ is a vector consisting of 2D locations $(x_i, y_i) = r_{\mathcal{D}} \in \mathcal{D}$ of the landmarks, while the texture $\mathbf{g}$ in AAM is a group of pixels, i.e. intensities or colors, defined in the texture domain $\mathcal{D}$.

The shape and the texture are represented using two linear PCA models, respectively. First, the shape is linearized as a mean shape $\mathbf{s}_0$ plus a linear combination of shape parameters $\boldsymbol{\alpha}_s$:

$$\mathbf{s}(\boldsymbol{\alpha}_s) = \mathbf{s}_0 + \mathbf{P}_s\boldsymbol{\alpha}_s \tag{4}$$

where $\mathbf{P}_s \in \mathbb{R}^{L_s \times N_s}$ is the matrix consisting of a set of orthonormal base vectors $\mathbf{p}_s^i$ describing the modes of variations learned from training set. Subsequently, all images in the training set are warped onto the mean shape by a model-warp $W(r_{\mathcal{D}}, \mathbf{s})$ and a similarity transformation $N(r_{\mathcal{I}}; \mathbf{q})$

defined in Eqn. (5); and then linearized by applying PCA on the ”shape-free” texture images.

$$\begin{aligned} W(r_{\mathcal{D}}; \mathbf{s}) &= r_{\mathcal{I}} \\ N(r_{\mathcal{I}}; \mathbf{q}) &= \begin{pmatrix} 1+\rho_1 & -\rho_2 \\ \rho_2 & 1+\rho_1 \end{pmatrix} r_{\mathcal{I}} + \tau \end{aligned} \tag{5}$$

where $\mathbf{q} = \{\rho, \tau\}$ composes of the global rotation ρ and the translation τ.

The texture $\mathbf{g}(r_{\mathcal{D}})$ in AAMs is a vectorized image defined over the pixels of $I(N(W(r_{\mathcal{D}}; \mathbf{s}); \mathbf{q}))$ inside the mean shape $\mathbf{s}_0$. The texture $\mathbf{g}(r_{\mathcal{D}}; \boldsymbol{\alpha}_g)$ can be represented as a mean texture $\mathbf{g}_0$ plus a linear combination of texture parameters $\boldsymbol{\alpha}_g$:

$$\mathbf{g}(r_{\mathcal{D}}; \boldsymbol{\alpha}_g) = \mathbf{g}_0(r_{\mathcal{D}}) + \mathbf{P}_g \boldsymbol{\alpha}_g \tag{6}$$

where $\mathbf{P}_g \in \mathbb{R}^{L_g \times N_g}$ is the set of orthonormal base vectors $\mathbf{p}_g^i$ learned from a given training set.

AAM fitting: In order to fit this model to a new testing image, a warping operator $W(r_{\mathcal{D}}; \mathbf{s})$ and a similarity transformation $N(r_{\mathcal{I}}; \mathbf{q})$ defined in Eqn. (5) are employed on that testing image. The parameters of shapes and textures are optimized so that the sum of squared errors between that testing image and the model texture instance are minimized:

$$[\boldsymbol{\alpha}_s^*, \boldsymbol{\alpha}_g^*] = \arg\min_{\boldsymbol{\alpha}_s, \boldsymbol{\alpha}_g} \| [I \circ N \circ W](\boldsymbol{\alpha}_s; \mathbf{q}) - \mathbf{g}(\boldsymbol{\alpha}_g) \|_{\mathcal{D}}^2 \tag{7}$$

where $[I \circ N \circ W](r_{\mathcal{D}}, \boldsymbol{\alpha}_s; \mathbf{q}) = I(N(W(r_{\mathcal{D}}; \boldsymbol{\alpha}_s); \mathbf{q}))$ is the normalized shape-free image warped from the input image I using W and N operators defined in Eqn. (5).

2.2 Boltzmann Machines (BM)

In this section, we first introduce a Product of Experts (PoE), the relationship between PoE and Markov Random Field, and how to train a PoE. Next, the architectures of Boltzmann Machines, Restricted Boltzmann Machines, Deep Boltzmann Machine, and how to train them using the formulations of PoE.

2.2.1 Product of Experts (PoE)

An option to model a high-dimensional and complicated data distribution is to take advantage of a large number of simple probabilistic models and combine the distributions provided by them (see [43]). An example of this type of technique is Mixture of Gaussians, where each simple model is a gaussian and the summation rule is applied to combine them. Product of Experts (PoE) [43] is another way of combining distribution by using the multiplication rule. Given the input data $\mathbf{x}$ and M individual models, called experts, the probability of $\mathbf{x}$ is defined as

$$p(\mathbf{x}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_M) = \frac{1}{Z^{PoE}} \prod_m p_m(\mathbf{x}|\boldsymbol{\theta}_m) \tag{8}$$

where $Z^{PoE} = \sum_{\tilde{\mathbf{x}}} \prod_m p_m(\tilde{\mathbf{x}}|\boldsymbol{\theta}_m)$ denotes the partition function which is the summation of all possible configurations $\tilde{\mathbf{x}}$; $\boldsymbol{\theta}_m$ is the parameters of m-th expert; $p_m(\mathbf{x}|\boldsymbol{\theta}_m)$ is the probability of $\mathbf{x}$ assigned by m-th expert. Notice that all experts are not required to be normalized probabilistic models. However, in order to form a valid PDF, their product needs to be renormalized by partition function Z^{PoE}.

Compared to the mixture model, PoE produces sharper distributions as the result of multiplication. Moreover, a sample receives high overal probability only when all experts assign high probabilities to it. Therefore, no expert can overrule the others even when it assigns very high probability to the sample.

2.2.2 Product of Experts and Markov Random Field (MRF)

Let us consider a particular case when the exponential function is chosen for all experts

$$p_m(\mathbf{x}|\boldsymbol{\theta}_m) = e^{-\frac{1}{T}\psi_m(\mathbf{x})} \tag{9}$$

where $\psi(\mathbf{x})$ stands for the potential function and the temperature T is a regularization parameter. The formulation of Markov Random Field then can be obtained by inserting Eqn. (9) to Eqn. (8)

$$\begin{aligned} p(\mathbf{x}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_M) &= \frac{1}{Z^{MRF}} \prod_m e^{-\frac{1}{T}\psi_m(\mathbf{x})} \\ &= \frac{1}{Z^{MRF}} e^{-\frac{1}{T}\sum_m \psi_m(\mathbf{x})} \\ &= \frac{1}{Z^{MRF}} e^{-\frac{1}{T}E(\mathbf{x})} \end{aligned} \tag{10}$$

where $Z^{MRF} = \sum_{\tilde{\mathbf{x}}} e^{-\frac{1}{T}E(\tilde{\mathbf{x}})}$ is the partition function. Notice that the distribution expressed by this way is also called Boltzmann distribution. From this, we can see that a PoE model with exponential experts is a MRF.

2.2.3 Training a Product of Experts

Several training techniques for PoE have been proposed in literature. A classical and widely used technique is the Maximum Likelihood Estimation (MLE) [14]. However, with the need of computing the partition function, MLE is very limited in term of computational cost particularly when the dimensions of training data become increasingly high. Therefore, Contrastive Divergence (CD) is introduced by Hinton et al. [43] to overcome these limitations. In the following subsections, the main features of both techniques will be presented.

Maximum Likelihood Learning

Given a set of observed data $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ which are assumed to be independent and identically distributed (i.i.d.) random variables and the set of model parameters $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_M\}$, the MLE approach finds the optimal $\boldsymbol{\theta}$ by maximizing the likelihood $p(\mathbf{X}|\boldsymbol{\theta})$ or the log likelihood $\log p(\mathbf{X}|\boldsymbol{\theta})$. Since the training data is i.i.d., the probability distribution can be simplified to the product of probabilities of data samples. Mathematically, the parameters in the model are optimized as follows.

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \log p(\mathbf{X}|\boldsymbol{\theta}) \tag{11}$$

where $\log p(\mathbf{X}|\boldsymbol{\theta}) = \log \prod_{\mathbf{x}} p(\mathbf{x}|\boldsymbol{\theta})$.

The gradient w.r.t each $\boldsymbol{\theta}_m$ is given by

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{\theta}_m} \log p(\mathbf{X}|\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_M) =& \frac{\partial}{\partial \boldsymbol{\theta}_m} \log \prod_{\mathbf{x}} p(\mathbf{x}|\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_M) \\
=& \frac{\partial}{\partial \boldsymbol{\theta}_m} \log \prod_{\mathbf{x}} \frac{\prod_e p_e(\mathbf{x}|\boldsymbol{\theta}_e)}{\sum_{\tilde{\mathbf{x}}} \prod_e p_e(\tilde{\mathbf{x}}|\boldsymbol{\theta}_e)} \\
=& \frac{\partial}{\partial \boldsymbol{\theta}_m} \log \prod_{\mathbf{x}} \prod_{e} p_e(\mathbf{x}|\boldsymbol{\theta}_e) - \frac{\partial}{\partial \boldsymbol{\theta}_m} \log \prod_{\mathbf{x}} \sum_{\tilde{\mathbf{x}}} \prod_{e} p_e(\tilde{\mathbf{x}}|\boldsymbol{\theta}_e) \\
=& \sum_{\mathbf{x}} \frac{\partial \log p_m(\mathbf{x}|\boldsymbol{\theta}_m)}{\partial \boldsymbol{\theta}_m} - N \frac{\partial \log \sum_{\tilde{\mathbf{x}}} \prod_e p_e(\tilde{\mathbf{x}}|\boldsymbol{\theta}_e)}{\partial \boldsymbol{\theta}_m}
\end{aligned} \tag{12}$$

The second term in the RHS can be computed as follows.

$$\begin{aligned}
\frac{\partial \log \sum_{\tilde{\mathbf{x}}} \prod_e p_e(\tilde{\mathbf{x}}|\boldsymbol{\theta}_e)}{\partial \boldsymbol{\theta}_m} =& \frac{1}{\sum_{\tilde{\mathbf{x}}} \prod_e p_e(\tilde{\mathbf{x}}|\boldsymbol{\theta}_e)} \frac{\partial \sum_{\tilde{\mathbf{x}}} \prod_e p_e(\tilde{\mathbf{x}}|\boldsymbol{\theta}_e)}{\partial \boldsymbol{\theta}_m} \\
=& \frac{1}{\sum_{\tilde{\mathbf{x}}} \prod_e p_e(\tilde{\mathbf{x}}|\boldsymbol{\theta}_e)} \sum_{\tilde{\mathbf{x}}} \prod_{e \neq m} p_e(\tilde{\mathbf{x}}|\boldsymbol{\theta}_e) \frac{\partial p_m(\tilde{\mathbf{x}}|\boldsymbol{\theta}_m)}{\partial \boldsymbol{\theta}_m} \\
=& \frac{1}{\sum_{\tilde{\mathbf{x}}} \prod_e p_e(\tilde{\mathbf{x}}|\boldsymbol{\theta}_e)} \sum_{\tilde{\mathbf{x}}} \prod_{e} p_e(\tilde{\mathbf{x}}|\boldsymbol{\theta}_e) \frac{\partial \log p_m(\tilde{\mathbf{x}}|\boldsymbol{\theta}_m)}{\partial \boldsymbol{\theta}_m} \\
=& \sum_{\tilde{\mathbf{x}}} \frac{\prod_e p_e(\tilde{\mathbf{x}}|\boldsymbol{\theta}_e)}{\sum_{\tilde{\mathbf{x}}} \prod_e p_e(\tilde{\mathbf{x}}|\boldsymbol{\theta}_e)} \frac{\partial \log p_m(\tilde{\mathbf{x}}|\boldsymbol{\theta}_m)}{\partial \boldsymbol{\theta}_m} \\
=& \sum_{\tilde{\mathbf{x}}} p(\tilde{\mathbf{x}}|\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_M) \frac{\partial \log p_m(\tilde{\mathbf{x}}|\boldsymbol{\theta}_m)}{\partial \boldsymbol{\theta}_m}
\end{aligned} \tag{13}$$

From Eqns. (12) and (13), the gradient is given by

$$\frac{\partial}{\partial \boldsymbol{\theta}_m} \log p(\mathbf{X}|\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_M) = \sum_{\mathbf{x}} \frac{\partial \log p_m(\mathbf{x}|\boldsymbol{\theta}_m)}{\partial \boldsymbol{\theta}_m} - N \sum_{\tilde{\mathbf{x}}} p(\tilde{\mathbf{x}}|\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_M) \frac{\partial \log p_m(\tilde{\mathbf{x}}|\boldsymbol{\theta}_m)}{\partial \boldsymbol{\theta}_m} \tag{14}$$

With this gradient, one can obtain the optimal parameters for PoE model by a gradient descent method. However, when the dimension of data becomes increasingly high, the second term is computationally infeasible due to the exponentially increasing number of configurations. Therefore, it needs to be approximated by numerical sampling techniques such as Gibbs sampling or other algorithms from the famlily of Markov Chain Monte Carlo (MCMC) methods. However, running the sampling Markov Chain to convergence to the target distribution still takes very long time.

Contrastive Divergence

Due to the problem of evaluating the partition function, Contrastive Divergence proposed by Hinton [43] provides another way to estimate the gradient of the energy function without the need to reach the equilibrium distribution. The main idea of this technique is to minimize the difference between the data and equilibrium distributions by using the Kullback-Leibler divergence. Specifically, let p^0 and $p^\infty_{\boldsymbol{\theta}}$ be the data distribution and the equilibrium distribution obtained by running Gibbs sampling. Then an equivalent objective function to MLE is to minimize the Kullback-Leibler divergence (KL divergence) between p^0 and $p^\infty_{\boldsymbol{\theta}}$ which is defined as

$$\begin{aligned} p^0 || p^\infty_{\boldsymbol{\theta}} &= \sum_{\mathbf{x}} p^0(\mathbf{x}) \log \frac{p^0(\mathbf{x})}{p^\infty_{\boldsymbol{\theta}}(\mathbf{x})} &(15)\\ &= \sum_{x} p^0(\mathbf{x}) \log p^0(\mathbf{x}) - \sum_{x} p^0(\mathbf{x}) \log p^\infty_{\boldsymbol{\theta}}(\mathbf{x}) &(16)\\ &= H(p^0) - \left\langle \log p^\infty_{\boldsymbol{\theta}} \right\rangle_{p^0} &(17) \end{aligned}$$

where $H(p^0) = -\sum_x p^0(\mathbf{x}) \log p^0(\mathbf{x})$ denotes the entropy of the data distribution; $\langle f \rangle_p := \sum_x p(x) f(x)$ is the expectations of function $f(x)$ over the distribution $p(x)$; and $p^\infty_{\boldsymbol{\theta}} = p(\mathbf{x}|\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_M)$. Since $H(p^0)$ is independent of the model parameters, it can be ignored during optimization process.

Taking the derivative w.r.t $\boldsymbol{\theta}_m$, we have

$$\frac{\partial p^0 || p^\infty_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}_m} = -\left\langle \frac{\partial \log p^\infty_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}_m} \right\rangle_{p^0} \tag{18}$$

The eqn. (14) can be rewritten in term of expectations as follows.

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{\theta}_m} \log p(\mathbf{X}|\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_M) &= N \sum_{\mathbf{x}} \frac{1}{N} \frac{\partial \log p_m(\mathbf{x}|\boldsymbol{\theta}_m)}{\partial \boldsymbol{\theta}_m} - N \sum_{\tilde{\mathbf{x}}} p(\tilde{\mathbf{x}}|\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_M) \frac{\partial \log p_m(\tilde{\mathbf{x}}|\boldsymbol{\theta}_m)}{\partial \boldsymbol{\theta}_m} \\ &= N \sum_{\mathbf{x}} p^0(\mathbf{x}) \frac{\partial \log p_m(\mathbf{x}|\boldsymbol{\theta}_m)}{\partial \boldsymbol{\theta}_m} - N \sum_{\tilde{\boldsymbol{x}}} p(\tilde{\boldsymbol{x}}|\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_M) \frac{\partial \log p_m(\tilde{\mathbf{x}}|\boldsymbol{\theta}_m)}{\partial \boldsymbol{\theta}_m} \quad (19) \\ &= N \left\langle \frac{\partial \log p_m(\mathbf{x}|\boldsymbol{\theta}_m)}{\partial \boldsymbol{\theta}_m} \right\rangle_{p^0} - N \left\langle \frac{\partial \log p_m(\tilde{\mathbf{x}}|\boldsymbol{\theta}_m)}{\partial \boldsymbol{\theta}_m} \right\rangle_{p^\infty_{\boldsymbol{\theta}}} \end{aligned}$$

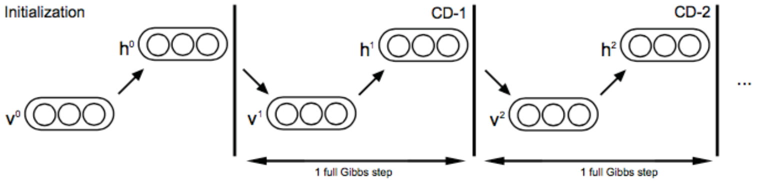


Figure 2.2: Gibbs chain in Contrastive Divergence.

Moreover, we also have

$$\begin{aligned}\frac{\partial}{\partial\boldsymbol{\theta}_m}\log p(\mathbf{X}|\boldsymbol{\theta}_1,...,\boldsymbol{\theta}_M) &= N\sum_{\mathbf{x}}\frac{1}{N}\frac{\partial\log p(\mathbf{x}|\boldsymbol{\theta}_1,...,\boldsymbol{\theta}_M)}{\partial\boldsymbol{\theta}_m}\\ &= N\sum_{\mathbf{x}}p^0(\mathbf{x})\frac{\partial\log p(\mathbf{x}|\boldsymbol{\theta}_1,...,\boldsymbol{\theta}_m)}{\partial\boldsymbol{\theta}_m}\\ &= \left\langle\frac{\partial\log p^\infty_{\boldsymbol{\theta}}}{\partial\boldsymbol{\theta}_m}\right\rangle_{p^0}\end{aligned}\tag{20}$$

From eqns. (20) and (19), we have

$$\left\langle\frac{\partial\log p^\infty_{\boldsymbol{\theta}}}{\partial\boldsymbol{\theta}_m}\right\rangle_{p^0} = \left\langle\frac{\partial\log p_m(\mathbf{x}|\boldsymbol{\theta}_m)}{\partial\boldsymbol{\theta}_m}\right\rangle_{p^0} - \left\langle\frac{\partial\log p_m(\tilde{\mathbf{x}}|\boldsymbol{\theta}_m)}{\partial\boldsymbol{\theta}_m}\right\rangle_{p^\infty_{\boldsymbol{\theta}}}\tag{21}$$

Substituting Eqn. (21) to Eqn. (18)

$$\frac{\partial p^0||p^\infty_{\boldsymbol{\theta}}}{\partial\boldsymbol{\theta}_m} = -\left\langle\frac{\partial\log p_m(\mathbf{x}|\boldsymbol{\theta}_m)}{\partial\boldsymbol{\theta}_m}\right\rangle_{p^0} + \left\langle\frac{\partial\log p_m(\tilde{\mathbf{x}}|\boldsymbol{\theta}_m)}{\partial\boldsymbol{\theta}_m}\right\rangle_{p^\infty_{\boldsymbol{\theta}}}\tag{22}$$

Notice that running the Gibbs chain for approximating $p^\infty_{\boldsymbol{\theta}}$ is very computationally expensive. Therefore, instead of computing the $\frac{\partial p^0||p^\infty_{\boldsymbol{\theta}}}{\partial\boldsymbol{\theta}_m}$ directly, Contrastive Divergence minimizes the difference between $p^0||p^\infty_{\boldsymbol{\theta}}$ and $p^1_{\boldsymbol{\theta}}||p^\infty_{\boldsymbol{\theta}}$ where $p^1_{\boldsymbol{\theta}}$ is the reconstructions of the data obtained by running the Gibbs chain for one step. By this way, the $p^\infty_{\boldsymbol{\theta}}$ can be canceled out. The new objective function is defined as follows:

$$CD = p^0||p^\infty_{\boldsymbol{\theta}} - p^1_{\boldsymbol{\theta}}||p^\infty_{\boldsymbol{\theta}}\tag{23}$$

Then the derivative w.r.t θ_m is given by

$$\begin{aligned} -\frac{\partial}{\partial \boldsymbol{\theta}_m}(p^0||p^\infty_{\boldsymbol{\theta}} - p^1_{\boldsymbol{\theta}}||p^\infty_{\boldsymbol{\theta}}) =& \left\langle \frac{\partial \log p_m(\mathbf{x}|\boldsymbol{\theta}_m)}{\partial \boldsymbol{\theta}_m} \right\rangle_{p^0} - \left\langle \frac{\partial \log p_m(\tilde{\mathbf{x}}|\boldsymbol{\theta}_m)}{\partial \boldsymbol{\theta}_m} \right\rangle_{p^\infty_{\boldsymbol{\theta}}} \\ & - \left\langle \frac{\partial \log p_m(\mathbf{x}|\boldsymbol{\theta}_m)}{\partial \boldsymbol{\theta}_m} \right\rangle_{p^1_{\boldsymbol{\theta}}} + \left\langle \frac{\partial \log p_m(\tilde{\mathbf{x}}|\boldsymbol{\theta}_m)}{\partial \boldsymbol{\theta}_m} \right\rangle_{p^\infty_{\boldsymbol{\theta}}} \\ & + \frac{\partial p^1_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}_m} \frac{\partial p^1_{\boldsymbol{\theta}}||p^\infty_{\boldsymbol{\theta}}}{\partial p^1_{\boldsymbol{\theta}}} \end{aligned} \tag{24}$$

$$-\frac{\partial}{\partial \boldsymbol{\theta}_m}(p^0||p^\infty_{\boldsymbol{\theta}} - p^1_{\boldsymbol{\theta}}||p^\infty_{\boldsymbol{\theta}}) \propto \left\langle \frac{\partial \log p_m(\mathbf{x}|\boldsymbol{\theta}_m)}{\partial \boldsymbol{\theta}_m} \right\rangle_{p^0} - \left\langle \frac{\partial \log p_m(\mathbf{x}|\boldsymbol{\theta}_m)}{\partial \boldsymbol{\theta}_m} \right\rangle_{p^1_{\boldsymbol{\theta}}} \tag{25}$$

The nice property of Contrastive Divergence is that the intractable expectation over $p^\infty_{\boldsymbol{\theta}}$ is now cancelled out. Therefore, if the experts are tractable, the exact value of $\frac{\partial \log p_m(\mathbf{x}|\boldsymbol{\theta}_m)}{\partial \boldsymbol{\theta}_m}$ can be computed and the whole process becomes computationally tractable.

For the number of steps in the Gibbs chain, if $p^1_{\boldsymbol{\theta}}$ is considered, we have CD-1 algorithm. On the other hand, in case the Gibbs chain is run for k steps and $p^k_{\boldsymbol{\theta}}$ is considered, we have CD-k. In summary, the CD-k algorithm consists of two main ideas:

(1) Run the Gibbs chain for *k* steps starting at the input data as illustrated in Figure 2.2.

(2) Compute the gradient for updating the parameters

$$\Delta \boldsymbol{\theta}_m \propto \left\langle \frac{\partial \log p_m(\mathbf{x}|\boldsymbol{\theta}_m)}{\partial \boldsymbol{\theta}_m} \right\rangle_{p^0} - \left\langle \frac{\partial \log p_m(\mathbf{x}|\boldsymbol{\theta}_m)}{\partial \boldsymbol{\theta}_m} \right\rangle_{p^k_{\boldsymbol{\theta}}} \tag{26}$$

2.2.4 Boltzmann Machines

Boltzmann Machines (BM) introduced by Hinton and Sejnowski [45] are probabilistic graphical models that can be interpreted as a stochastic recurrent neural network. In particular, this is an undirected graphical model consisting two layers of stochastic units, i.e. visible $v_i, i = 1..N_v$ and hidden units $h_j, j = 1..N_h$. The visible units represent the observed data while the hidden units are latent variables interpreting the conditional hidden representation of that data. The connections between units are undirected with the weights interpreting the pairwise constraints between them.

The structure of BM with three hidden units and four visible units is illustrated in Figure 2.3(a). With this structure, there are two notes about BM.

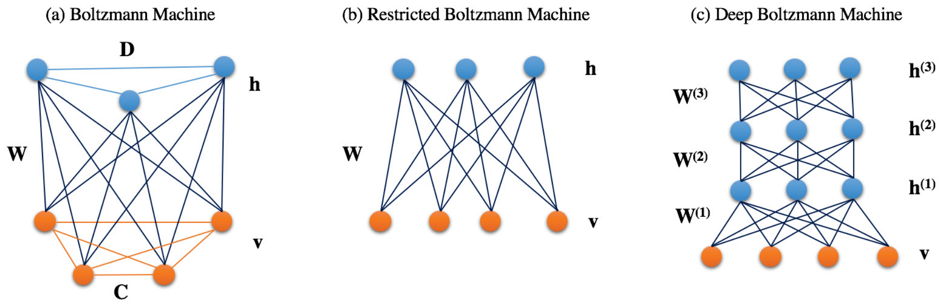


Figure 2.3: Examples of (a) a Boltzmann Machine; (b) Restricted Boltzmann Machines with three hidden units and four visible units; and(c) Deep Boltzmann Machines with three hidden layers.

(1) A fully connected BM with binary units is very similar to Hopfield network [46] except the units are stochastic rather than deterministic.

(2) Boltzmann Machines are a MRF with a particular energy function that reflects its structure of undirected graph.

Given a state of visible units $\mathbf{v}$ and hidden units $\mathbf{h}$, the energy is given as follows.

$$-E(\mathbf{v},\mathbf{h};\boldsymbol{\theta}) = \sum_{i\in vis} v_i b_i + \sum_{k\in hid} h_k a_k + \sum_{i<j} v_i v_j c_{ij} + \sum_{i,k} v_i h_k w_{ik} + \sum_{k<l} h_k h_l d_{kl} \tag{27}$$

where b_i stands for the bias term of i-th visible unit; a_k is the bias of k-th hidden unit; and $\boldsymbol{\theta} = \{\mathbf{W},\mathbf{C},\mathbf{D}\}$ are the weights of visible-to-hidden, visible-to-visible, and hidden-to-hidden interactions, respectively.

The probability of a visible vector $\mathbf{v}$ assigned by BM is given by

$$p(\mathbf{v};\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})}\sum_{\mathbf{h}} \exp(-E(\mathbf{v},\mathbf{h};\boldsymbol{\theta})) \tag{28}$$

where $Z(\boldsymbol{\theta}) = \sum_{\mathbf{v}}\sum_{\mathbf{h}} \exp(-E(\mathbf{v},\mathbf{h};\boldsymbol{\theta}))$ is the partition function. The conditional distributions over hidden and visible units are given by

$$p(h_k = 1|\mathbf{v},\mathbf{h}_{-k}) = \sigma(\sum_i w_{ik} v_i + \sum_{l\neq k} d_{kl} h_l) \tag{29}$$

$$p(v_i = 1|\mathbf{h},\mathbf{v}_{-\mathbf{i}}) = \sigma(\sum_k w_{ik} h_k + \sum_{j\neq i} c_{ij} v_j) \tag{30}$$

where $\sigma(x) = 1/(1 + \exp(-x))$ is the logistic function.

One of the interesting properties of BM is its generality. Given a training data, a BM can be trained such that it can assign a probability to every possible input data and generate new data according to the learn distribution.

Moreover, Boltzmann Machines are stackable. In other words, a set of BMs can be organized in several layers such that each BM is stacked on the top of another BM. As a result, a deeper network is produced. Figure 2.3(c) shows an example of deep network with three hidden layers. The new network, therefore, is more powerful to learn more complex probability densities and able to extract higher-level features of the data.

2.2.5 Restricted Boltzmann Machines (RBM)

Restricted Boltzmann Machines (RBM) [43, 113] is a simplified version of BM where there is no intra connections between units in the same layer. In other words, the visible-to-visible and hidden-to-hidden connections are removed and, therefore, resulted in a bipartite graph where visible and hidden units are pairwise conditionally independent. An example of RBM with three hidden units and four visible units is illustrated in Figure 2.3(b).

Thanks to this structure, the hidden units are conditionally independent given the states of visible units and, therefore, simplifying the training and inference processes. Given a binary state of $\mathbf{v}, \mathbf{h}$, the energy of RBM can be computed as

$$\begin{aligned} -E(\mathbf{v}, \mathbf{h}) &= \sum_i \sum_j v_i w_{ij} h_j + \sum_i b_i v_i + \sum_j a_j h_j \\ &= \mathbf{v}^T \mathbf{W} \mathbf{h} + \mathbf{b}^T \mathbf{v} + \mathbf{a}^T \mathbf{h} \end{aligned} \tag{31}$$

In contrast to general BM, the inference process in RBM is exact. Thanks to this important property, the Contrastive Divergence technique can perform well to obtain the model parameters for RBM.

The conditional propabilities

The conditional probability of hidden units given the visible units can be computed as follows.

$$\begin{aligned}
p(\mathbf{h}|\mathbf{v}) &= \frac{p(\mathbf{v},\mathbf{h})}{p(\mathbf{v})} = \frac{p(\mathbf{v},\mathbf{h})}{\sum_{\hat{\mathbf{h}}} p(\mathbf{v},\hat{\mathbf{h}})} && (32)\\
&= \frac{\frac{1}{Z}\exp(-E(\mathbf{v},\mathbf{h}))}{\sum_{\hat{\mathbf{h}}} \frac{1}{Z}\exp(-E(\mathbf{v},\hat{\mathbf{h}}))} && (33)\\
&= \frac{\exp(-E(\mathbf{v},\mathbf{h}))}{\sum_{\hat{\mathbf{h}}} \exp(-E(\mathbf{v},\hat{\mathbf{h}}))} && (34)\\
&= \frac{\exp(\mathbf{v}^T\mathbf{W}\mathbf{h} + \mathbf{b}^T\mathbf{v} + \mathbf{a}^T\mathbf{h})}{\sum_{\hat{\mathbf{h}}} \exp(\mathbf{v}^T\mathbf{W}\hat{\mathbf{h}} + \mathbf{b}^T\mathbf{v} + \mathbf{a}^T\hat{\mathbf{h}})} && (35)\\
&= \frac{\exp(\mathbf{b}^T\mathbf{v})\exp(\mathbf{v}^T\mathbf{W}\mathbf{h} + \mathbf{a}^T\mathbf{h})}{\exp(\mathbf{b}^T\mathbf{v})\sum_{\hat{\mathbf{h}}} \exp(\mathbf{v}^T\mathbf{W}\hat{\mathbf{h}} + \mathbf{a}^T\hat{\mathbf{h}})} && (36)\\
&= \frac{\exp(\sum_i\sum_j v_i w_{ij} h_j + \sum_j a_j h_j)}{\sum_{\hat{\mathbf{h}}} \exp(\sum_i\sum_j v_i w_{ij} \hat{h}_j + \sum_j a_j \hat{h}_j)} && (37)
\end{aligned}$$

Since hidden units $\hat{\mathbf{h}}$ are binary, we can decompose the denominator into two parts, i.e. the sum over all $\hat{\mathbf{h}}$ such that $\hat{h}_{N_h} = 0$ and the sum over all $\hat{\mathbf{h}}$ such that $\hat{h}_{N_h} = 1$. The denominator is then rewritten as

$$\begin{aligned}
&\sum_{\hat{\mathbf{h}}} \exp(\sum_{i=1}^{N_v}\sum_{j=1}^{N_h} v_i w_{ij}\hat{h}_j + \sum_{j=1}^{N_h} a_j \hat{h}_j)\\
=&(e^0 + e^{\sum_{i=1}^{N_v} v_i w_{iN_h}\hat{h}_1}) \sum_{\hat{\mathbf{h}}' \in \{0,1\}^{N_h-1}} \exp(\sum_{i=1}^{N_v}\sum_{j=1}^{N_h-1} v_i w_{ij}\hat{h}'_j + \sum_{j=1}^{N_h-1} a_j \hat{h}'_j)
\end{aligned} \qquad (38)$$

Repeating this process for all hidden units, we have

$$\sum_{\hat{\mathbf{h}}} \exp(\sum_{i=1}^{N_v}\sum_{j=1}^{N_h} v_i w_{ij}\hat{h}_j + \sum_{j=1}^{N_h} a_j \hat{h}_j) = \prod_{j=1}^{N_h}(1 + \exp(\sum_{i=1}^{N_v} v_i w_{ij} + a_j)) \qquad (39)$$

Combine Eqn. (39) and Eqn. (37), we have

$$
\begin{aligned}
p(\mathbf{h}|\mathbf{v}) &= \frac{\exp(\sum_i \sum_j v_i w_{ij} h_j + \sum_j a_j h_j)}{\prod_j (1 + \exp(\sum_i v_i w_{ij} + a_j))} && (40) \\
&= \prod_j \frac{\exp(\sum_i v_i w_{ij} h_j + a_j h_j)}{1 + \exp(\sum_i v_i w_{ij} + a_j)} && (41) \\
&= \prod_j p(h_j|\mathbf{v}) && (42)
\end{aligned}
$$

The individual activation probabilities are then given by

$$
p(h_j = 1|\mathbf{v}) = \sigma(\sum_i v_i w_{ij} + a_j) \tag{43}
$$

where $\sigma(\cdot)$ is the logistic function.

Since RBM is symmetric, the conditional probability of visible units given the hidden units can be derived in the same way. Therefore, we also have

$$
\begin{aligned}
p(\mathbf{v}|\mathbf{h}) &= \prod_i p(v_i|\mathbf{h}) && (44) \\
p(v_i = 1|\mathbf{h}) &= \sigma(\sum_j h_j w_{ij} + b_i) && (45)
\end{aligned}
$$

Training Restricted Boltzmann Machines

Given a set of training data, the RBM can be trained by performing CD-1 (or CD-k) learning. The partial derivative of the energy function w.r.t the model parameters $\theta = \{\mathbf{W}, \mathbf{a}, \mathbf{b}\}$ is given by

$$
\begin{aligned}
\frac{\partial E(\mathbf{v}, \mathbf{h})}{\partial \mathbf{W}} &= -\mathbf{v}\mathbf{h}^T && (46) \\
\frac{\partial E(\mathbf{v}, \mathbf{h})}{\partial \mathbf{b}} &= -\mathbf{v} && (47) \\
\frac{\partial E(\mathbf{v}, \mathbf{h})}{\partial \mathbf{a}} &= -\mathbf{h} && (48)
\end{aligned}
$$

Then the optimal parameter values can be obtained in a gradient ascent fashion given by

$$\begin{aligned}
\Delta \mathbf{W} &= \alpha \left(\langle \mathbf{v}\mathbf{h}^T \rangle_{p^0} - \langle \mathbf{v}\mathbf{h}^T \rangle_{p^1} \right) && (49) \\
\Delta \mathbf{b} &= \alpha \left(\langle \mathbf{v} \rangle_{p^0} - \langle \mathbf{v} \rangle_{p^1} \right) && (50) \\
\Delta \mathbf{a} &= \alpha \left(\langle \mathbf{h} \rangle_{p^0} - \langle \mathbf{h} \rangle_{p^1} \right) && (51)
\end{aligned}$$

where α stands for the learning rate.

Different types of Restricted Boltzmann Machines

In order to deal with different kinds of data, RBM has received several extensions in its structures and unit types (i.e. binary, linear). In this subsection, I will go through some main features of these extensions.

Gaussian Restricted Boltzmann Machine Instead of using the binary visible units as the original RBM, Gaussian RBM [61] assumes the visible units have values in $[-\infty, \infty]$ and normally distributed with mean b_i and variance σ_i^2. By this way, this extension of RBM can be used for modelling real-valued data, i.e. pixel intensities. The energy function is modified as follows.

$$E(\mathbf{v}, \mathbf{h}; \boldsymbol{\theta}) = \sum_i \frac{(v_i - b_i)^2}{2\sigma_i^2} - \sum_i \sum_j \frac{v_i}{\sigma_i} w_{ij} h_j - \sum_j a_j h_j \tag{52}$$

The conditional distributions over $\mathbf{v}$ and $\mathbf{h}$ are then given as in Eqn. (53).

$$\begin{aligned}
p(h_j | \mathbf{v}) &= \delta \left(\sum_i w_{ij} \frac{v_i}{\sigma_i} + a_j \right) \\
p(v_i | \mathbf{h}) &\sim \mathcal{N} \left(\sigma_i \sum_j w_{ij} h_j + b_i, \sigma_i^2 \right)
\end{aligned} \tag{53}$$

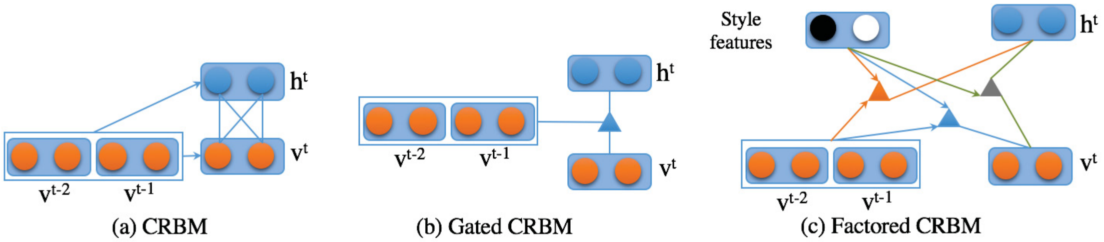


Figure 2.4: The structure of Conditional Restricted Boltzmann Machines and its variants.

The update rules for the model parameters are

$$\Delta w_{ij} = \alpha \left(\langle \frac{1}{\sigma_i} v_i h_j \rangle_{p^0} - \langle \frac{1}{\sigma_i} v_i h_j \rangle_{p^1} \right) \tag{54}$$

$$\Delta b_i = \alpha \left(\langle \frac{1}{\sigma_i^2} v_i \rangle_{p^0} - \langle \frac{1}{\sigma_i^2} v_i \rangle_{p^1} \right) \tag{55}$$

$$\Delta a_j = \alpha \left(\langle h_j \rangle_{p^0} - \langle h_j \rangle_{p^1} \right) \tag{56}$$

Conditional Restricted Boltzmann Machine With the need of modeling the temporal dependencies in time-series data, the standard form of RBM has been extended in many works. Although these adaptations are different in structures, their main motivation is very similar, i.e. treating the variables in previous time steps as additional input for the current time step. Inspired by this idea, in the Conditional RBM (CRBM) proposed by Taylor et al. [124], the feed forward connections from previous time steps between visible layers and from visible-to-hidden layers are incorporated as in Fig. 2.4(a). Therefore, the visible variables of time step t are further conditional on previous visible states. The main advantage of this model is that it can inherit most important properties of standard RBM, i.e. simple, exact inference and efficient approximate learning. Learning in CRBM is very similar to RBM, except the bias terms are redefined to take into account the new connections between layers.

Memisevic and Hinton [78] later introduced a Gated Conditional Restricted Boltzmann Machines (GCRBM) by implementing the multiplicative interactions in CRBM. In this model, there are three sets of units, i.e. input, output and hidden units. The main idea of this model is to let the input units directly influence the interactions between units instead of simply incorporating them via bias terms. By this way, the input units will be able to gate the basic function for reconstructing the output. This model was originally developed to learn the transformations between image pairs. However, it can be easily extended for sequential data by considering the output to be the current frame and the input to be the previous frames as in Figure 2.4(b). As a result, the new energy function is defined as

$$-E(\mathbf{v},\mathbf{h}|\mathbf{x},\boldsymbol{\theta}) = \sum_{ijk} w_{ijk} v_i h_j x_k + \sum_{ij} w'_{ij} v_i h_j + \sum_i b_i v_i + \sum_j a_j h_j \qquad (57)$$

By introducing a set of deterministic factors f, w_{ijk} can be factorized to three pairwise interactions: (1) $w^{\mathbf{v}}_{if}$ connects v_i to factor f; (2) $w^{\mathbf{h}}_{jf}$ connects h_j to factor f; (3) $w^{\mathbf{x}}_{kf}$ connects x_k to factor f.

A factored version of CRBM can be also found in the work of Taylor et al. [123] with the application to motion style modeling. This model is very similar to GCRBM with additional units for motion style. The structure of this factored CRBM (FCRBM) is illustrated in Figure 2.4(c). This model was then extended by Chiu et al. [21] with additional hierarchical structure for style interpolation. Taylor et al. [125] proposed another variant of CRBM that can learn from the data with several modes (e.g. walking and running in body modeling task). The main idea of this model is to introduce a new discrete variable with states. At each time step, only one element of has non-zero value and therefore, it can decide which particular CRBM is active. Since the variable is embedded directly to the energy function, the model is called implicit Mixtures of CRBM (imCRBM).

2.2.6 Deep Boltzmann Machines (DBM)

Deep Boltzmann Machines (DBM) [107] are a probabilistic generative model that consists of many hidden layers. Each higher layer plays a role of capturing the correlations between features of its lower layer. The structure of DBM contains several RBMs are organized in a layered manner. In DBM, the connections are between visible units and the hidden units in the first layer as well as between the hidden units in adjacent hidden layers. The structure of DBM with three hidden layers is illustrated in Figure 2.3(c).

Thanks to this structure, the hidden units in higher layer can learn more complicated correlations of features captured in lower layer. Another interesting point of DBM is that these higher representations can be built from the training data in an unsupervised fashion. Then the labeled training data, assumed to be very limited, can be used to fine tune the model for a particular application. Notice that unlike other models such as Deep Belief Network [44] or Deep Autoencoders [11], all connections between units in two consecutive layers are undirected. As a result, each unit receives both bottom-up and top-down information and, therefore, better propagate uncertainty during the inference process.

Let $\mathbf{v}$ be the set of visible units and $\{\mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \mathbf{h}^{(3)}\}$ be the set of units in three hidden layers, the energy of the state $\{\mathbf{v}, \mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \mathbf{h}^{(3)}\}$ is given as follows.

$$-E(\mathbf{v}, \mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \mathbf{h}^{(3)}; \boldsymbol{\theta}) = \mathbf{v}^{\top}\mathbf{W}^{(1)}\mathbf{h}^{(1)} + \mathbf{h}^{(1)\top}\mathbf{W}^{(2)}\mathbf{h}^{(2)} + \mathbf{h}^{(2)\top}\mathbf{W}^{(3)}\mathbf{h}^{(3)} \tag{58}$$

where $\boldsymbol{\theta} = \{\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{W}^{(3)}\}$ are the weights of visible-to-hidden and hidden-to-hidden connections. Notice that the bias terms for visible and hidden units are ignored in Eqn. (58) for simplifying the representation. Similar to RBM, the probability of a visible vector $\mathbf{v}$ assigned by the model is

$$p(\mathbf{v}; \theta) = \frac{1}{Z(\boldsymbol{\theta})} \sum_{\mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \mathbf{h}^{(3)}} \exp(-E(\mathbf{v}, \mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \mathbf{h}^{(3)}; \boldsymbol{\theta})) \tag{59}$$

and the conditional distributions over $\mathbf{v}$, $\mathbf{h}^{(1)}$, $\mathbf{h}^{(2)}$ and $\mathbf{h}^{(3)}$ are computed as follows.

$$p(h_j^{(1)}|\mathbf{v}, \mathbf{h}^{(2)}) = \sigma\left(\sum_i v_i w_{ij}^{(1)} + \sum_k h_k^{(2)} w_{jk}^{(2)}\right) \tag{60}$$

$$p(h_k^{(2)}|\mathbf{h}^{(1)}, \mathbf{h}^{(3)}) = \sigma\left(\sum_j h_j^{(1)} w_{jk}^{(2)} + \sum_l h_l^{(3)} w_{kl}^{(3)}\right) \tag{61}$$

$$p(h_l^{(3)}|\mathbf{h}^{(2)}) = \sigma\left(\sum_k h_k^{(2)} w_{kl}^{(3)}\right) \tag{62}$$

$$p(v_i = 1|\mathbf{h}^{(1)}) = \sigma\left(\sum_j w_{ij}^{(1)} h_j\right) \tag{63}$$

In order to train a DBM, the procedure for general Boltzmann Machines can still be applied. However, if one starts from the random initial weights, it will be slow, particularly when the hidden units are remote from the visible units. Therefore, a greedy layerwise pretraining for DBM is proposed in [106]. This algorithm is represented in Algorithm 1 and illustrated in Figure 2.5.

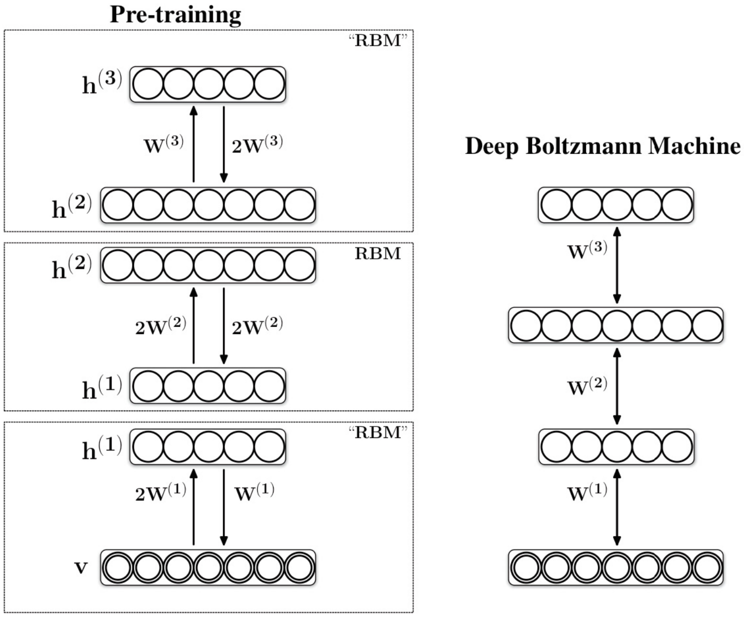


Figure 2.5: Pretraining a DBM [106]: learning a stack of RBMs. The weights of the first and the last RBMs are modified with a factor of two in one direction. For the intermediate RBMs, both directions are modified.

Inspiring from the advantages of RBM and DBM, in the next chapter, a novel approach called Deep Appearance Models (DAMs) will be introduced. With this proposed model, a face can be modeled effectively by employing two DBMs, i.e. one for facial shape and the other for its texture. The proposed DAMs are also shown their advantages of modeling large and non-linear facial variations.

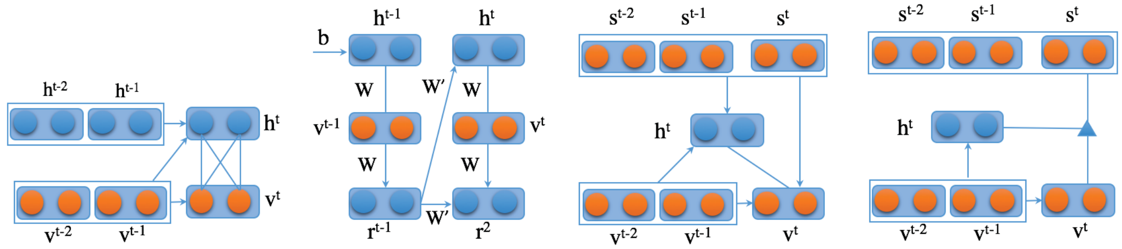
2.3 Temporal Restricted Boltzmann Machines (TRBM)

Temporal Restricted Boltzmann Machine (TRBM) [118] has gained significant attention as one of the probabilistic models that can accurately model complex time-series structure while keeping the inference tractable. It was shown to be successful in several tasks such as realistic human motion generating [124]; denoising low-resolution videos [118], and sequence-to-sequence mapping [144].

The major difference between the original RBM and TRBM is the directed connections from previous states of visible and hidden units as in Fig. 2.6(a). With these new connections, the short

Algorithm 1 Greedy Pretraining for DBM [106]

```
Input: Training data, number of layers $L$, $(L > 3)$ and number of hidden units in each layer.
Output: Initial weights for DBM $\{\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, ..., \mathbf{W}^{(L)}\}$.
Train the first layer: using CD-1 with mean-field reconstructions of the visible units. During the learning process, the bottom-up weights, $2\mathbf{W}^{(1)}$, are constrainted to be twice the top-down weights, $\mathbf{W}^{(1)}$.
Train the second layer: Freeze $2\mathbf{W}^{(1)}$ and use samples $\mathbf{h}^{(1)}$ from $P(\mathbf{h}^{(1)}|\mathbf{v}, 2\mathbf{W}^{(1)})$ as the training data. Then this is trained as original RBM with weights $2\mathbf{W}^{(2)}$ for both directions.
Train the third layer: Freeze $2\mathbf{W}^{(1)}, 2\mathbf{W}^{(2)}$ and use samples $\mathbf{h}^{(2)}$ from $P(\mathbf{h}^{(2)}|\mathbf{v}, 2\mathbf{W}^{(1)}, 2\mathbf{W}^{(2)})$ as the training data. This layer is trained in the same way as the previous one.
Train the other intermediate layers: Proceed recursively up to layer $L - 1$.
Train the top layer: the learning process is the same as training the first layer except the constraint is that the bottom-up weights, $\mathbf{W}^{(L)}$ is half of the top-down weights, $2\mathbf{W}^{(L)}$.
Use the weights $\{\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, ..., \mathbf{W}^{(L)}\}$ to compose a DBM.
```



(a) TRBM (b) Recurrent TRBM (c) IOTRBM (d) Factored Third-order TRBM

Figure 2.6: The structure of Temporal Restricted Boltzmann Machine and its variants.

history of their activations can act as "memory" and is able to contribute to the inference step of current states of visible units. The joint distribution over $(\mathbf{v}^t, \mathbf{h}^t)$ at time t is conditional on the past m states and given as

$$p(\mathbf{v}^t, \mathbf{h}^t|\mathbf{v}_{t-m}^{t-1}, \mathbf{h}_{t-m}^{t-1}) = \frac{1}{Z\left(\mathbf{v}_{t-m}^{t-1}, \mathbf{h}_{t-m}^{t-1}\right)} \exp\left(-E(\mathbf{v}^t, \mathbf{h}^t|\mathbf{v}_{t-m}^{t-1}, \mathbf{h}_{t-m}^{t-1})\right) \quad (64)$$

where $\left(\mathbf{v}_{t-m}^{t-1}, \mathbf{h}_{t-m}^{t-1}\right)$ are the sequence of visible and hidden units from time $t - m$ to $t - 1$. The energy is computed as in Eqn. (31) except the new bias terms are defined as

$$\hat{\mathbf{b}}^t = \mathbf{b} + \sum_{k=1}^{m} \mathbf{B}_k \mathbf{v}^{t-k} \quad (65)$$

$$\hat{\mathbf{a}}^t = \mathbf{a} + \sum_{k=1}^{m} \mathbf{A}_k \mathbf{h}^{t-k} + \sum_{k=1}^{m} \mathbf{C}_k \mathbf{v}^{t-k} \quad (66)$$

Sutskever et al. [119] later pointed out that the inference step in TRBM is non-trivial task due to the need of evaluating the exact ratio of two RBM partition functions. Therefore, they introduced the Recurrent TRBM (RTRBM) whose structure is very similar to TRBM but the exact inference is much easier. An illustration of the RTRBM structure is shown in Fig. 2.6(b). The main difference between TRBM and RTRBM is in the introduction of which is the expected value of the hidden units, i.e. $\mathbb{E}\left[\mathbf{h}^t|\mathbf{v}^t\right]$. Notice that the variable $\mathbf{r}^t$ are real valued while those of $\mathbf{h}^t$ are binary. The energy function of RTRBM for $t > 1$ is computed as

$$-E(\mathbf{v}^t, \mathbf{h}^t|\mathbf{r}^{t-1}) = \mathbf{h}^{t\top}\mathbf{W}\mathbf{v}^t + \hat{\mathbf{b}}^\top\mathbf{v}^t + \hat{\mathbf{a}}^\top\mathbf{h}^t + \mathbf{h}^{t\top}\mathbf{W}'\mathbf{r}^{t-1} \tag{67}$$

where $\mathbf{W}'$ denotes the weights of the connections from $\mathbf{r}^{t-1}$ to $\mathbf{h}^t$ and $\mathbf{r}^t$; $\mathbf{r}^t = \sigma(\mathbf{W}\mathbf{v}^t + \hat{\mathbf{a}} + \mathbf{W}'\mathbf{r}^{t-1})$ if $t > 1$ and $\mathbf{r}^t = \sigma(\mathbf{W}\mathbf{v}^t + \hat{\mathbf{a}}_{init})$ if $t = 1$.

Zeiler et al. [144] further extended the TRBM so that it can transfer the facial expression by mapping two facial landmark sequences. Two models were proposed in this work, i.e. Input-Output TRBM (IOTRBM) and Factor Third-order Input-Output TRBM (FIOTRBM). Experimental results in the facial expression transfer problem showed that these two models are very prominent for learning the mapping between sequences. The structures of IOTRBM and FIOTRBM are represented in Figs. 2.6(c) and 2.6(d). Let $\mathbf{s}_1^T$ be the input sequence containing the information to be transferred and $\mathbf{v}_1^T$ be the output sequence. Notice that the role of $\mathbf{v}_1^T$ is the same as visible units in the original form of TRBM. An assumption for this model is that the whole input sequence and the first N frames of the output are accessible. The extension of TRBM to IOTRBM when modeling $p(\mathbf{v}^t|\mathbf{v}_{t-N}^{t-1}, \mathbf{s}_{t-N}^t)$ is straightforward by incorporating the input $\mathbf{s}_{t-N}^t$ to its energy function via bias terms. The inference and learning stages are kept the same as TRBM.

The higher-order RBMs where the variables interact multiplicatively are also explored in the structure of FIOTRBM. Instead of using $\mathbf{W}, \mathbf{P}, \mathbf{Q}$, a three-way weight tensor is employed to connect the input, current output frame and hidden units. As a result, the energy function is redefined as Eqn. (68) where $\mathbf{s}^{<=t} = \mathbf{s}_{t-N}^t$.

$$-E(\mathbf{v}^t, \mathbf{h}^t|\mathbf{v}_{t-N}^{t-1}, \mathbf{s}_{t-N}^t) = \sum_i \frac{1}{2}(v_i^t - \hat{b}_i^t)^2 + \sum_j h_j^t \hat{a}_j^t + \sum_k \sum_{ijk} w_{ik}^{\mathbf{v}} w_{jk}^{\mathbf{h}} w_{lk}^{\mathbf{s}} v_i^t h_j^t s_l^{<=t} \tag{68}$$

H$\ddot{a}$usler et al. [41] improved the training process of TRBM by employing denoising Autoencoder to initialize the weights for the hidden-to-hidden connections. The main motivation is that the data frame at time $t - m$ can be considered as a corrupted version of the data frame at time t. Therefore, the pre-training step can be progressed in the fashion of denoising Autoencoder.

Recently, Mittelman et al. [81] introduced the structured RTRBM (SRTRBM). In this model, instead of employing the fully connected topology between visible and hidden units of the RTRBM, they constructed block masking $\mathbf{M}_{\mathbf{W}}$ and $\mathbf{M}_{\mathbf{W}'}$ for the weight matrices $\mathbf{W}$ and $\mathbf{W}'$ to model sparsely connectivity between groups of visible units and hidden units; and between groups of hidden units themselves. By this way, the proposed model is able to learn the dependency structure and patterns within the input data. The block masking matrices $\mathbf{M}_{\mathbf{W}}$ and $\mathbf{M}_{\mathbf{W}'}$ are adjacency matrices, whose entries can be either 0 or 1, representing this graph structure. Then the new energy function is similar to Eqn. (67) except the weight matrices are redefined as $\mathbf{W} = \mathbf{W} \odot \mathbf{M}_{\mathbf{W}}$ and $\mathbf{W}' = \mathbf{W}' \odot \mathbf{M}_{\mathbf{W}'}$. The authors also suggested to use spike and slab RBM instead of the Gaussian RBM for better conditional covariance modeling.

2.4 Convolutional Neural Networks (CNN/ ConvNet)

Unlike RBM whose neurons in one layer fully connect to all neurons in previous layer, Convolutional Neural Networks (CNN) is a biologically-inspired variant of feed-forward artificial neural network where each neuron only responds to a local region, i.e. receptive field, of the visual field. From studies on the visual cortex system [32, 48], neurons in visual cortex are more sensitive to local regions and, therefore, these local connectivities are well-suited to exploit spatially local correlation presented in input images. Moreover, when the parameters are shared among neurons, not only does the neural network have the translation invariance property but also the learning process is more efficient with smaller number of trainable parameters. Motivating from these studies, LeCun et al. [67, 68] proposed a CNN architecture with back-propagation training and successfully applied to several pattern recognition tasks such as zip code reading, hand written character recognition, etc. Then several CNN architectures have been proposed in literature such as AlexNet [62], ZF Net [143], GoogLeNet [120], VGGNet [112], and ResNet [42].

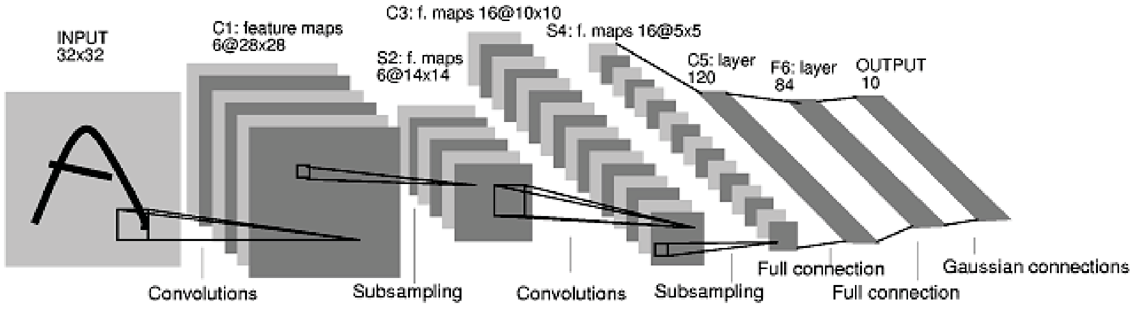


Figure 2.7: An example of CNN architecture: LeNet model [68].

Table 2.1: A comparison between RBM and CNN.

	RBM	**CNN**
Type	Stochastic	Deterministic
Learning	Joint Probability Distribution of hidden and input variables	A deterministic function
Structure	Bipartite graph	Feedforward
Topology	Fully Connected No specification about network's topology	Locally connected Neurons
Tractability	Intractable	Tractable
Weight update	Contrastive Divergence	Backpropagation

Comparing to traditional pattern recognition algorithms and other hand-engineered features, CNN has shown its advantages in (1) less preprocessing requirement, and (2) the independence of prior knowledge. The following subsections present the main structure of CNN architecture and its building blocks. A comparison between RBM and CNN is also provided in Table 2.1.

CNN structure: A simple CNN structure consists of a sequence of layers where each layer transforms an input 3D volumes of neurons to another via a differentiable function. There are three main types of fundamental CNN layers including Convolutional Layer, Pooling Layer, and Fully-Connected Layer. Together with these layers, two types of functions, i.e. Activation Function to increase the non-linearity and Loss Function defining the objective according to the task, are usually used. Figure 2.7 illustrates a simple CNN architecture with six layers. With this organization, the features extracted from CNN can be also divided into several levels. Features in the first level (i.e. extracted by some first convolutional layers) usually encode simple visual features such as

Algorithm 2 : Operation of Convolutional Layer [55]

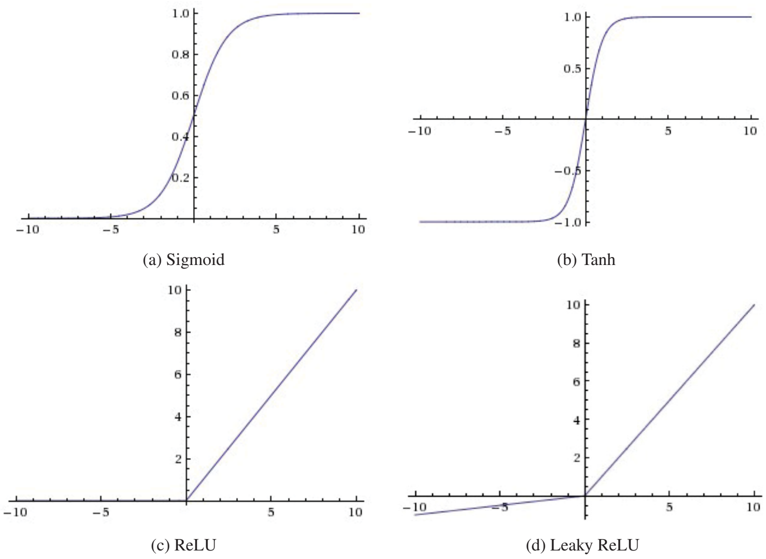
```
Input: 3D input volume $\mathbf{h}^k \in \mathbb{R}^{W_1 \times H_1 \times D_1}$; number of filter $K$; filter size $F$; strike $S$; the amount
    of zero padding $P$; weight $\mathbf{W}^k \in \mathbb{R}^{(F \times F \times D_1) \times K}$ ; and bias $\mathbf{b}^k \in \mathbb{R}^K$
Output: 3D input volume $\mathbf{h}^{k+1} \in \mathbb{R}^{W_2 \times H_2 \times D_2}$ where
    $W_2 = \frac{W_1 - F + 2P}{S} + 1$
    $W_2 = \frac{H_1 - F + 2P}{S} + 1$
    $D_2 = K$
    Perform convolutional operation
    $\mathbf{h}^{k+1} = \mathbf{W}^k \otimes \mathbf{h}^k + \mathbf{b}^k$
```

edge, color blobs, etc. In the next level, the extracted features will be the combinations of previous features, i.e. the combinations of edges, the corner. As a result, the more levels a CNN has, the higher-level features can be extracted.

Convolutional Layer: Considered as the building block of CNN that makes CNN different from other neural networks, this layer consists of multiple learnable filters that are small spatially and have the same depth as the input. The input to each layer is a 3D volume, i.e. a $W \times H \times C$ image with the width W, height H, and C channels. Each filter of this layer is defined by its weights and bias. The hyperparameters of each convolutional layer include the number of filters and their size; the stride defining the number of slided pixels before convolving the filters; and the padding amount to handle borders pixels. During the forward pass, all filters are convolved with all positions across the width and height of the input volume to produce a stack of feature maps which is the input for the next layer. Formally, let $\mathbf{h}^k$ be the input of k-th convolutional layer in the network and $\{\mathbf{W}^k, \mathbf{b}^k\}$ be the weights and bias of its filters. The output feature map of $\mathbf{h}^k$ can be computed as follows.

$$\mathbf{h}^{k+1} = \mathbf{W}^k \otimes \mathbf{h}^k + \mathbf{b}^k \tag{69}$$

where $\otimes$ is the convolutional operation. The Algorithm 2 illustrates the operation of a Convolutional Layer. By incorporating this type of layer to the structure, the CNN has more capabilities of capturing the local features (i.e. edges, corners) via the local connectivity and shift variance with the spatially weight sharing property. As one can see in the next sub-section, the pooling layers also help CNN to reduce its sensitivity to shifts and distortion.



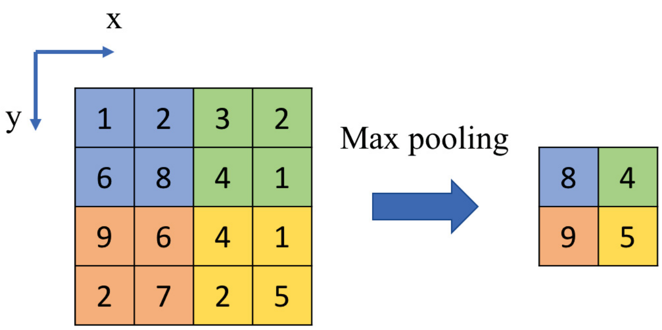
(a) Sigmoid

(b) Tanh

(c) ReLU

(d) Leaky ReLU

Figure 2.8: Four types of activation functions [55].



1	2	3	2
6	8	4	1
9	6	4	1
2	7	2	5

8	4
9	5

Figure 2.9: Max pooling operation.

Activation Function: Similar to other neural networks, the non-linearity plays an important role when the network becomes deeper with many layers. This non-linearity property is usually obtained via an activation function. There are many types of activation functions having been used in literature such as sigmoid, tanh, and Rectified Linear Unit (ReLU) functions. These functions are usually required to be differentiable to ensure the usage of back-propagation process. Thanks to these activation functions, the output feature maps are also constrained in a appropriate data ranges which help to improve the stability of the CNN network as well as the independence of neurons in consecutive layers. Figure 2.8 and Table 2.2 illustrate the figures, formulations, and properties of the four most common activation functions for CNN.

Table 2.2: Four commonly used activation functions in CNN.

Function	**Formulation**	**Properties**
Sigmoid	$sigm(x) = \frac{1}{1+e^{-x}}$	Convert the input value into $[0, 1]$. **Pros**: Has a nice interpretation as firing rate of a neuron. **Cons**: (1) easily saturated and has the vanishing problem that makes the network have no learning ability. (2) The output value is not zero-centered.
Tanh	$tanh(x) = \frac{1-e^{-2x}}{1+e^{-2x}}$	Convert the input value into $[-1, 1]$. **Pros**: (1) Zero-centered; (2) helping to avoid the zig-zagging dynamics during training. **Cons**: easily saturated
Rectified Linear Unit (ReLU)	$relu(x) = \max(0, x)$	**Pros**: (1) Simple formulation; (2) faster than $sigm$ and $tanh$ function; (3) does not have the saturating problem. **Cons**: some neurons can become totally inactive.
Leaky ReLU	$lrelu(x) = \begin{cases} x & x \geq 0 \\ \alpha x & x < 0 \end{cases}$	**Pros**: Similar to $relu$; fixing the problem of inactive neurons.

Algorithm 3 : The Pooling Operation [55]

```
Input: 3D input volume with the size of $W_1 \times H_1 \times D_1$; filter size $F$; strike $S$.
Output: 3D input volume with the size of $W_2 \times H_2 \times D_2$ where
    $W_2 = \frac{W_1-F}{S} + 1$
    $W_2 = \frac{H_1-F}{S} + 1$
    $D_2 = D_1$
1: Perform pooling operation (i.e. sub-sampling or max pooling).
```

Pooling Layer: In a CNN structure, pooling layer is usually incorporated between convolutional layers to reduce the resolution of the feature maps. This type of layer can help to reduce the number of network parameters and, therefore, reduce the computational cost as well as alleviate the overfitting. Moreover, reducing the resolution of the feature map could provide the spatial invariance for the whole network. Given a feature map from a convolutional layer, it is first divided into a set of non-overlaping $n \times n$ patches. Then a pooling operation is applied to each patch and produces a smaller sized feature map. Notice that this operation independently operated on each depth slice of the input feature map. The Algorithm 3 and Figure 2.9 show the pooling operation and an example of max pooling, respectively.

There are two common pooling layers for CNN including *sub-sampling pooling* and *max pooling*. Let $\mathbf{a}_{ij}^{n\times n}$ be a $n \times n$ patch centered in the position (i, j) of the input feature map.

- **Sub-sampling pooling:** The output value $\hat{a}$ for this patch can be computed as follows.

$$\hat{a} = \beta \sum_{hk} \left(\mathbf{a}_{ij}^{n\times n}\right)_{hk} + b \tag{70}$$

  In a specific case of $\beta = \frac{1}{n\times n}$ and $b = 0$, this is similar to the average pooling.

- **Max pooling:** The output value is obtained by extracting the maximum value of $\mathbf{a}_{ij}^{n\times n}$:

$$\hat{a} = \max\left(\left(\mathbf{a}_{ij}^{n\times n}\right)_{hk}\right) \tag{71}$$

  In practice, compared to sub-sampling pooling, max pooling operation provides a better performance in terms of less training parameters and calculations as well as faster convergence rate.

Fully Connected Layer: The fully connected layer in CNN is similar to that of regular neural networks. In addition, one can view this layer type as a specific case of convolutional layer where the filter size is $W_1 \times H_1$.

Loss Function: Similar to regular neural networks, according to different tasks, various loss functions, i.e. objective function, are designed and added to the end of the CNN as a measurement to enforce the whole network learning useful information. For example, euclidean loss is usually used for real-valued regression tasks while Softmax loss is for single-class classification task.

Chapter 3

Literature Review

This chapter presents recent advances of face modeling including Active Appearance Models, Restricted Boltzmann Machines, and Generative Adversarial Networks Approaches. Then, in the second part of the chapter, different techniques for longitudinal face modeling will be presented.

3.1 Face Modeling

This section briefly reviews recent advances of AAM-based approaches for constructing and fitting deformable models.

3.1.1 AAM Approaches

AAM Modeling: One of the major drawbacks of AAM is that the models only capture small amounts of appearance variations which can be only expressed as a linear combination of the training samples. AAM perform poorly when unknown appearance variations are encountered due to changes in the real-world environment, e.g. facial poses, lighting conditions, camera change, etc. This leads to another drawback of AAM that the *person specific* AAM substantially outperform a *generic* one, i.e. models trained across numerous subjects.

Addressing the first drawback, some improvements have been made by applying the ideas of mixture models [130] and Probabilistic PCAs [53] to represent as much variations as possible especially for the appearance model. Maaten et al. [130] presented a mixture of K probabilistic PCA to model the texture variations and employed the Expectation-Maximization (EM) algorithm to train the appearance model. Joan et al. [53] also used the probabilistic PCA to model the appearance. In the fitting steps, a test image is linearized and projected to a latent texture space before the shape parameters are optimized using the gradient descent algorithm. Their method is prominent to detect the facial features. However, the assumption of multivariate Gaussian distribution is a prerequisite condition in these methods.

Descriptive feature-based approaches were employed instead of intensities-based AAM to deal with the second drawback of AAM. Ge et al. [33] proposed three Gabor-based texture representations for AAM capturing the characteristics of both Gabor magnitude and Gabor phase over scales (CGMPS), directions (CGMPD), and combination of scales and directions (CGMPSD). These Gabor-based texture representations are more compact, i.e. much smaller texture dimension, and more robust to various conditions, e.g. expression, illumination and pose changes. Antonakos et al. [7] proposed to use dense Histogram of Oriented Gradients (HOG) features with AAM. Their AAM fitting method achieves efficiently with Inverse Compositional (IC) optimization technique. The authors [7] showed that HOG features enhance the robustness and performance of AAM that generalize well to unseen faces with illumination, identity, pose and occlusion variations.

Following the aim of improving the generalization ability of AAM, Hasse et al. [39] proposed a completely different approach by incorporating related knowledge obtained from another training set. For example, in the case of illumination changes across face images, knowledge about unseen illumination conditions can be transfered to the existing AAM. Hasse et al. [39] used a transfer learning technique from machine learning to learn from related training data. The basic idea of their instance-weight transfer learning method is to estimate sample-specific weights to integrate similar and informative examples from the additional source training data.

AAM Fitting: Fitting steps in AAM are an iterative optimization process. It measures the cost between a new testing image and a model texture in the coordinate of a reference frame. Generally, previous fitting techniques can be divided into two categories, i.e. discriminative and generative approaches. In the first category, the optimizing process is updated using a trained parameter-updating model. There are several ways to train a model in this approach, e.g. perturbing the parameters and recording the residuals [23], directly using texture information to predict the shape [47], linear regression technique [27] and non-linear regression method [109], etc. These techniques usually require low computational costs. However, since the mapping function is fixed and independent of current model parameters, their performance in term of fitting quality is still limited.

In the second category, the fitting steps are formulated as an image alignment problem and iteratively solved using the Gaussian-Newton optimization technique. Matthews et al. [77] presented a project out inverse algorithm to work on the orthogonal complement of the texture subspace. Although the algorithm runs very fast since most of the terms can be precomputed, it can not perform well in generic AAM when testing faces are from untrained subjects. Thus, improving the ability of the AAM to generalize to unseen conditions has been a well investigated topic in the AAM fitting literature, Navarathna et al. [84] investigated the use of multiple filter response (e.g. Gabor) representation of the input image and proposed a computationally efficient AAM fitting algorithm based on a variant of the Lucas-Kanade (LK) algorithm, called Fourier LK (FLK). This fitting technique works in the Fourier domain that provides invariance to both expression and illumination, so their method is known as Fourier AAM (FAAM) [84]. Other methods find the shape and texture increments either simultaneously [37] or alternatively [91]. Amberg et al. [4] presented the compositional framework. Recently, Tzimiropoulos et al. [128] presented a fitting algorithm that works effectively in both forward and inverse cases. However, their method is also limited due to the assumption of the PCA-based model. Mollahosseini et al. [82] proposed bidirectional warping method based on image alignment for AAM fitting. The authors suggested to warp both the input image and the appearance template using incremental update by an affine transformation and an inverse compositional approach, respectively.

3.1.2 RBM Approaches

In addition to different types of RBM structrures as presented in Section 2.2.5, this section focuses on the RBM structures that have been used for face modeling. Vinod Nair et al. [83] proposed to generalize the formulation of binary hidden units by viewing each of them as a combination of a set of binary units with shared weights and different fixed bias offsets. By this way, the hidden units have more capability of encoding more information. The sum of their probabilities can be

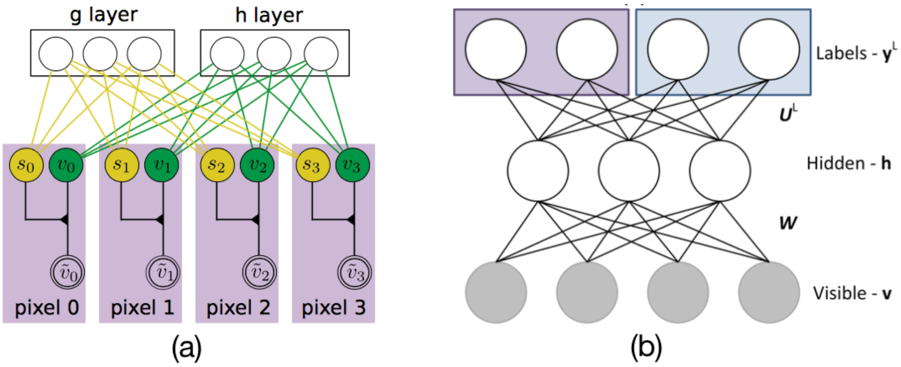


Figure 3.1: (a) Robust Restricted Boltzmann Machines (RoBM) [122] and (b) Multi-task Restricted Boltzmann Machines [29].

formulated by

$$\sum_{1}^{N} \sigma(\mathbf{v}\mathbf{W}^T + b - i + 0.5) \approx \log(1 + e^{\mathbf{v}\mathbf{W}^T + b}) \tag{72}$$

This formulation makes the hidden unit behaved as a noisy version of a smoothed rectified linear unit. With this new form, the authors have shown improvements compared to original RBM in both object recognition and face verification tasks.

Tang et al. [122] later have further developed Robust Boltzmann Machines (RoBM) that robustly deal with corruptions, i.e. occlusions, presented in the input data. In particular, the authors introduced a gating mechanism to distinguish the "good" and corrupted pixels by a scaled mixture of two Gaussians. Figure 3.1(a) presents the graphical model of RoBM. The energy function of a RoBM is written as following.

$$\begin{aligned} E_{RoBM}(\mathbf{v}, \tilde{\mathbf{v}}, \mathbf{s}, \mathbf{h}, \mathbf{g}) =& \frac{1}{2} \sum_i \frac{\gamma_i^2}{\sigma_i^2} s_i (v_i - \tilde{v}_i)^2 \\ & - \sum_i d_i s_i - \sum_k e_k g_k - \sum_{ik} U_{ik} s_i g_k \\ & + \frac{1}{2} \sum_i \frac{(v_i - b_i)^2}{\sigma_i^2} - \sum_j c_j h_j - \sum_{ij} W_{ij} v_i h_j \\ & + \frac{1}{2} \sum_i \frac{\left(\tilde{v}_i - \tilde{b}_i\right)^2}{\tilde{\sigma}_i^2} \end{aligned} \tag{73}$$

where s_i (a binary indicator), v_i (i-th visible unit of "clean" data), and $\tilde{v}_i$ (i-th visible unit of original data) are interacted via the first term of Eqn. (73). When $s_i = 0$, v_i and $\tilde{v}_i$ are allowed to be very different. When $s_i = 1$, they are controlled by the regulation variable γ_i^2. The next three lines of Eqn. (73) present energy functions to model the structure of s_i, GRBM for "clean" data $\mathbf{v}$ and the noise distribution for the original input data $\tilde{\mathbf{v}}$, respectively. Experimental results on several face databases such as Yale, Toronto Face and AR have shown the potential of this extension for various face modeling tasks.

Max Ehrlich et al. [29] introduced a Multi-Task Restricted Boltzmann Machines (MT-RBM) for facial attribute classification. Figure 3.1(b) illustrates the structure of MT-RBM. In this structure, given features of different attributes and their labels, an RBM is applied to learn the shared representation for all of them.

3.1.3 Generative Adversarial Networks (GAN) Approaches

In order to avoid the intractable Markov chain sampling, Goodfellow et al. [36] borrowed the idea from adversarial system to design their Generative Adversarial Networks (GAN). The intuition behind this approach is to set up a game between two players, i.e. *generator* and *discriminator*. On one hand, the discriminator learns to determine whether given data are from the generator or real samples. On the other hand, the generator learns how to fool the discriminator by its generated samples. This game continues as the learning process takes place. The learning process will stop at a point that the discriminator can't distinguish between real data and the ones produced by the generator. Moreover, this is also an indication that the generator has already learned the distribution of input data. This section reviews GAN structure and its extensions for face modeling. Figure 3.2 shows the graph structure of a GAN.

Formally, let $\mathbf{x}$ be the input data, p_g be the distribution learned from generator, and $p_z(\mathbf{z})$ be the prior distribution of noise variable $\mathbf{z}$. The two neural networks representing two differentiable functions for the generator G and discriminator D can be defined as follows.

$$\begin{aligned} G(\mathbf{z}, \theta_g) &: \mathbf{z} \mapsto \mathbf{x} \\ D(\mathbf{x}, \theta_d) &: \mathbf{x} \mapsto y \end{aligned} \tag{74}$$

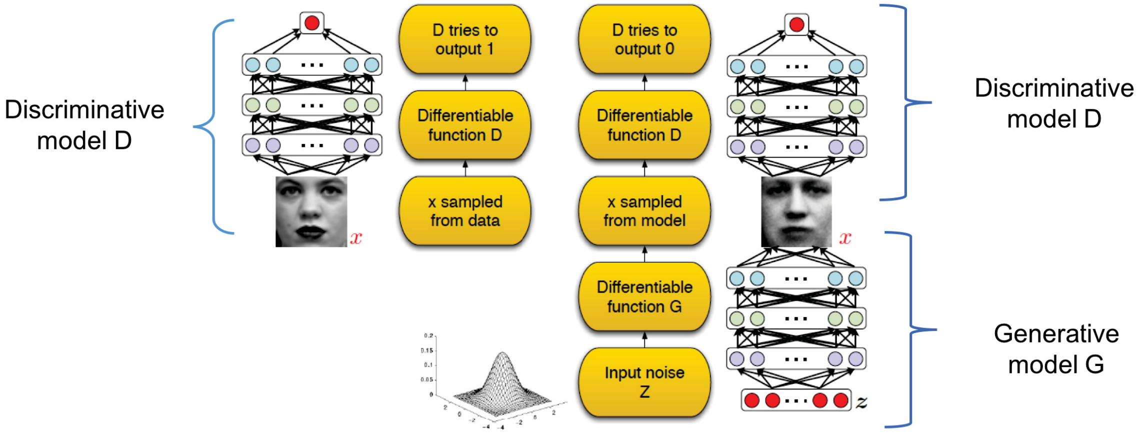


Figure 3.2: GAN framework [36].

where θ_g and θ_d are the parameters of the CNNs representing G and D, respectively. y denotes the probability that $\mathbf{x}$ comes from the data distribution rather than p_g. The training process is then formulated as maximizing the probability $D(\mathbf{x})$ while minimizing $\log(1 - D(G(\mathbf{z})))$:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))] \tag{75}$$

Algorithm 4 illustrates the steps to train a GAN. In original GAN, the use of fully connected neural network for its generator makes it very hard to generate high-resolution face images.

Numerous extensions of GAN focusing on different aspects of this structure have been proposed in literature. Denton et al. [25] scaled up the original GAN to produce high quality image in their proposed Laplacian pyramid Generative Adversarial Networks (LAPGAN). In LAPGAN, a conditional form of GAN is integrated into a Laplacian pyramid and generate images in coarse-to-fine manner. By this way, the generator at each level of the Laplaccian pyramid can capture the distribution of input images at the corresponding resolution. Experimental results have shown the potential of this model with compelling high-resolution images. In addition to generate images, the class labels can be also incorporated to the generating process for controllable generation.

Radford et al. [99] later introduced Deep Convolutional Generative Adversarial Networks (DCGAN) by adopting CNN architecture in place of the multilayer perceptron for higher-resolution image generation. In this approach, four main modifications are proposed for a stable DCGAN:

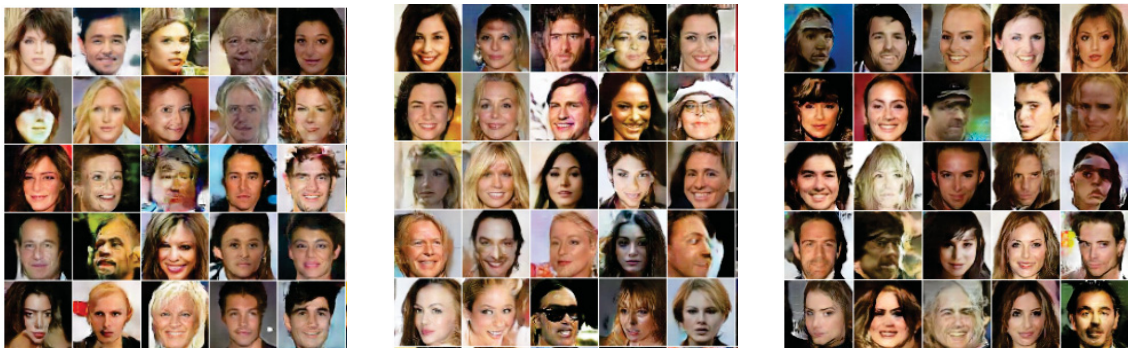
Algorithm 4 Training steps of Generative Adversarial Networks [36]

```
Input: Training data $\{x^{(1)}, \dots, x^{(m)}\}$; number of steps to apply to the discriminator $k$.
Output: The generator $G$ and discriminator $D$.
for number of training iterations do
    for $k$ steps do
        Sample minibatch of $m$ noise samples $\{z^{(1)}, \dots, z^{(m)}\}$ from noise prior $p_g(\mathbf{z})$.
        Sample minibatch of $m$ examples $\{x^{(1)}, \dots, x^{(m)}\}$ from data generating distribution $p_{data}(\mathbf{x})$.
        Update the discriminator by ascending its stochastic gradient:
            $\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\mathbf{x}^{(\mathbf{i})} + \log\left(1 - D\left(G\left(\mathbf{z}^i\right)\right)\right)\right)\right]$
    end for
    Sample minibatch of $m$ noise samples $\{z^{(1)}, \dots, z^{(m)}\}$ from noise prior $p_g(\mathbf{z})$.
    Update the generator by descending its stochastic gradient:
        $\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \left[\log\left(1 - D\left(G\left(\mathbf{z}^i\right)\right)\right)\right)$
end for
The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.
```

(1) utilizing strided convolutions for all pooling layers of discriminator and fractional-strided convolutions for pooling layers of genenerator; (2) removing fully connected hidden layers; (3) incorporating batch normalization; (4) and utilizing ReLU activation for generator and LeakyReLU for discriminator. Info-GAN [20] embeds the latent code for information loss to increase the meaning and interpretability of the generator's input $\mathbf{z}$.

Focusing on the convergence of GAN, in [8], a comprehensive theoretical analysis on distribution learning using Earth Mover (EM) distance was provided. Then, a new variant with EM based loss function, named Wasserstein GAN, was introduced. Although the training process of this model is slow, WGAN enjoys the benefit of stability and better convergence. Following these derivations from Wasserstein distance, two improved versions of WGAN, named WGAN-GP and BEGAN, is also proposed in [38] and [13], respectively.

Taken into account the advantages of energy models for stabilizing GAN training, Energy based GAN (EBGAN) [145] incorporated an Auto-Encoder architecture to GAN. In this approach, the



(a) DCGAN

(b) WGAN-GP

(c) α-GAN

Figure 3.3: Face generations from DCGAN [99], WGAN-GP [38], and α-GAN [103].

reconstruction error is considered as an energy measurement and the CNN structure of discriminator is replaced by an Auto-Encoder. This variant is easy to train and robust to the choices of hyper-parameters. In [65], the variational auto-encoder (VAE) is also combined with GAN by adding adversarial loss to the variational loss. Recently, Rosca et al. [103] also incorporated the auto-encoder with variational inference to the generator in their α-GAN for better synthesized results. Figure 3.3 shows synthesis results of different GAN variants including DCGAN [99], WGAN-GP [38], and α-GAN [103] for face modeling task.

3.2 Longitudinal Face Modeling

This section reviews various age progression approaches which can be divided into five groups: *anthropology*, *prototyping*, *modeling*, *reconstructing*, and *deep learning-based approaches*.

Anthropology approaches simulate the biological structure and aging process of facial features such as muscles and facial skins based on theories from anthropometric studies [9, 12, 15]. Inspiring from the 'revise' cardioidal strain transformation, Ramanathan et al. [100] proposed a physiological craniofacial growth model for age progression. Ramanathan et al. [101] later introduced an aging model that incorporates both shape and texture variation models. To simulate the geometry changes, the shape transformation models are designed to capture the aging variations of three facial muscles, i.e. linear muscles, sheet muscles and sphincter muscles. For the texture model, an image gradient based transformation function is adopted to characterize the facial wrinkles and skin artifacts.

Prototyping approaches use the age prototypes to synthesize new face images. The average faces of people in the same age group are used as the prototypes [104]. The input image can be transformed into the age-progressed face by adding the differences between the prototypes of two age groups [17]. Recently, Kemelmacher-Shlizerman et al. [57] proposed to construct sharper average prototype faces from a large-scale set of images in combining with subspace alignment and illumination normalization. In particular, sharper average faces are obtained via the collection flow method introduced in [69] to align and normalize all the images in one age group. Then illumination normalization and subspace alignment technique are proposed to better handle images with various lighting conditions. Although the implementation of *prototyping approaches* is usually straightforward, this type of approaches requires good alignments between faces in order to produce plausible results.

Modeling-based approaches represent facial shape and appearance via a set of parameters and model facial aging process via aging functions. Lanitis et al. [63] proposed to use AAM parameters and introduced several aging functions to model both generic and specific aging processes. Four variations of aging functions were introduced in this work: Global Aging Function, Appearance Specific Aging Function (ASA), Weighted Appearance Aging Function (WAA), and Weighted Person Specific Aging Function (WSA). Pattersons et al. [92] also used AAM and aging function in their system. However, they put more efforts on simulating the adult aging stage. The genetic facial features of siblings and parents were also incorporated to age progression in [73]. Geng et al. [34] proposed an AGing pattErn Subspace (AGES) approach for both age estimation and age synthesis. The key idea of this approach is to construct a representative subspace for *aging patterns* where each aging pattern is a chronological sequence of face images of the same subject. Then given an image, the proper aging pattern is determined by the projection in this subspace that produces smallest reconstruction error. Finally, the age of that face is indicated by its position in the aging pattern while the synthesized results in other ages are the reconstructed faces corresponding to other positions. Tsai et al. [127] then extended the AGES with the guidance faces corresponding to the subject's characteristics for more stable results. Instead of representing faces in a global fashion, Suo et al. [116] proposed to decompose a face into smaller components (i.e. eyes, mouth, etc.) and learning the aging process for each component. A three-layer And-Or graph is adopted for face representation. Then the changes in face aging are modeled by a Markov chain on parse graphs. Similarly, in [117], Suo et al. further employed this decomposition strategy in temporal aspects where long-term evolution of the graphical representation is learned by connecting sequences of short-term patterns.

Reconstructing-based methods reconstruct the aging face from the combination of an aging basis in each group. Shu et al. [111] proposed to build aging coupled dictionaries (CDL) to represent personalized aging pattern by preserving personalized facial features. The dictionaries are learned using face pairs from neighboring age groups via a “personality-aware coupled reconstruction loss”. Yang et al. [138] proposed to model person-specific and age-specific factors separately via sparse representation hidden factor analysis (HFA). Since only age-specific gradually changes over time, the age factor is transformed to the target age group via sparse reconstruction and then combined with the identity factor to achieve the aged face.

Recently, *deep learning-based approaches* are being developed to exploit the power of deep learning methods, i.e. Recurrent Neural Network (RNN), DBM, and Generative Adversarial Networks (GANs). Duong et al. [89] employed Temporal Restricted Boltzmann Machines (TRBM) to model the non-linear aging process with geometry constraints and spatial DBMs to model a sequence of reference faces and wrinkles of adult faces. Similarly, Wang et al. [132] modeled aging sequences using a recurrent neural network with a two-layer gated recurrent unit (GRU). Conditional Generative Adversarial Networks (cGAN) is also applied to synthesize aged images in [6]. The authors focus on optimizing identity preserving GAN’s latent vectors while facial aging property is controlled by conditional GANs.

Chapter 4

Deep Appearance Models for Face Modeling

In this chapter, we discuss the proposed Deep Appearance Models (DAM) that take advantage of deep model for face representation and reconstruction under large variations. In contrast to previous models where Deep Boltzmann Machine is used as shape prior model [30, 125, 135] or higer-level representation [107], the novelty of DAM approach is threefold. First, both face shape and texture are model by two DBMs to deal with large variations. Second, the higher-level representation of both shape and texture can be extracted from the top-level hidden layer of DAM. Finally, a fitting step is proposed to synthesize new input image. Some materials of this chapter have been published in [88]. This chapter consists of two main sections: (1) the structure of DAM; and (2) the modeling fitting step.

4.1 DAM architecture

The structure of DAM consists of three main parts, i.e. two prior models for shape and texture and an additional higher-level hidden layer for appearance modeling. Figure 4.1 demonstrates the architecture of DAM. The shape model is used to learn the facial shape structure while texture model is used for texture variations. Both of them are mathematically modeled using the Deep Boltzmann

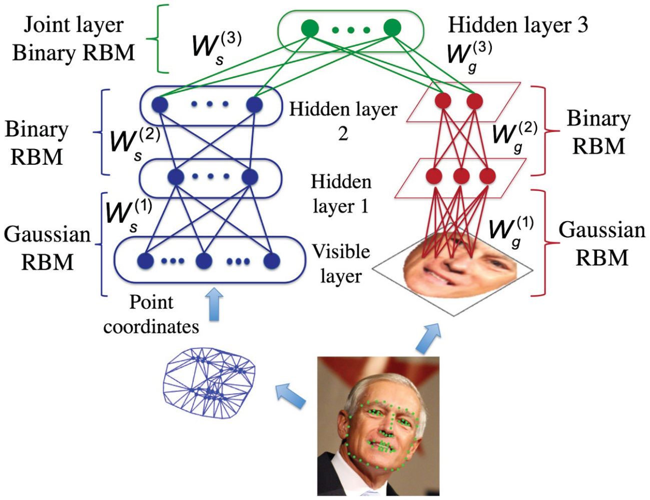


Figure 4.1: Deep Appearance Models that consists of shape model (left), texture model (right) and the joint representation of shape and texture.

Machines that are capable to model high-order correlations among input data. Its undirected connections provide both bottom-up and top-down passes to efficiently send updates between the texture model and the shape model. These modeling shape and texture parameters are then embedded in a higher-level layer that can be learned by clamping both shapes and textures as observations for the model. In this section, three main steps of constructing the model, i.e. shape, texture and appearance modeling, are introduced. Then in next section, a fitting algorithm will be presented in order to synthesize any given new face image.

4.2 Shape modeling

In order to generalize possible patterns of facial shapes, a two-layer DBM is employed to learn the distributions of their landmark points. As illustrated in Figure 4.1, the shape model, i.e. left part of DAM, consists of a set of visible units encoding the coordinates of landmark points and two sets of hidden units that are latent variables. The connections are symmetric and only those connecting units in adjacent layers are employed.

Let a shape $\mathbf{s} = [x_1, y_1, ..., x_N, y_N]^T$ with N landmark points $\{x_i, y_i\}, x_i \in \mathbb{R}, y_i \in \mathbb{R}$ be the visible units; and $\mathbf{h}_s^{(1)} \in \{0,1\}^{F_s^1}, \mathbf{h}_s^{(2)} \in \{0,1\}^{F_s^2}$ be the binary variables of the first and second hidden layers respectively. F_s^1 and F_s^2 stand for the number of units in these hidden layers. Since

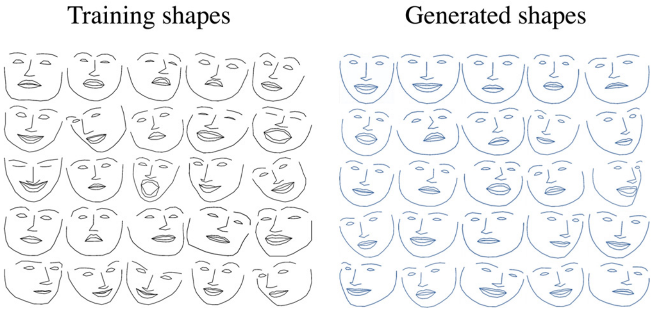


Figure 4.2: A subset of training shapes and generated shapes from shape model with 10-step Gibbs sampling.

$\{x_i, y_i\}$ are real values while $\mathbf{h}_s^{(1)}$ and $\mathbf{h}_s^{(2)}$ are binary, the Gaussian-Bernoulli Restricted Boltzmann Machine (GRBM) is employed for the first layer and a binary-binary RBM is for the subsequent one. The energy of the joint configuration $\{\mathbf{s}, \mathbf{h}_s^{(1)}, \mathbf{h}_s^{(2)}\}$ in facial shape modeling is formulated as follows:

$$\begin{aligned} E(\mathbf{s}, \mathbf{h}_s^{(1)}, \mathbf{h}_s^{(2)}; \theta_s) = &\sum_i \frac{(s_i - b_{s_i})^2}{2\sigma_{s_i}^2} - \sum_{i,j} \frac{s_i}{\sigma_{s_i}} W_{sij}^{(1)} h_{sj}^{(1)} \\ &- \sum_{j,l} h_{sj}^{(1)} W_{sjl}^{(2)} h_{sl}^{(2)} \end{aligned} \tag{76}$$

where $\theta_s = \{\mathbf{W}_s^{(1)}, \mathbf{W}_s^{(2)}, \boldsymbol{\sigma}_s^2, \mathbf{b}_s\}$ are the model parameters representing the connecting weights of visible-to-hidden and hidden-to-hidden interactions, the variance, and the bias of visible units.

Notice that in Eqn. (76), the bias terms of hidden units are ignored to simplify the equation. Its corresponding probability is then given by the Boltzmann distribution:

$$\begin{aligned} P(\mathbf{s}; \theta_s) &= \sum_{\mathbf{h}_s^{(1)}, \mathbf{h}_s^{(2)}} P(\mathbf{s}, \mathbf{h}_s^{(1)}, \mathbf{h}_s^{(2)}; \theta_s) \\ &= \frac{1}{Z(\theta_s)} \sum_{\mathbf{h}_s^{(1)}, \mathbf{h}_s^{(2)}} e^{-E(\mathbf{s}, \mathbf{h}_s^{(1)}, \mathbf{h}_s^{(2)}; \theta_s)} \end{aligned} \tag{77}$$

where $Z(\theta_s)$ is the partition function.

The conditional distributions over $\mathbf{s}$, $\mathbf{h}_s^{(1)}$, and $\mathbf{h}_s^{(2)}$ are then given as in Eqn. (78).

$$\begin{aligned} p(h_{sj}^{(1)}|\mathbf{s},\mathbf{h}_s^{(2)}) &= \delta\left(\sum_i W_{sij}^{(1)}\frac{s_i}{\sigma_{s_i}} + \sum_l W_{sjl}^{(2)} h_{sl}^{(2)}\right) \\ p(h_{sl}^{(2)}|\mathbf{h}_s^{(1)}) &= \delta\left(\sum_j W_{sjl}^{(2)} h_{sj}^{(1)}\right) \\ s_i|\mathbf{h}_s^{(1)} &\sim \mathcal{N}\left(\sigma_{s_i}\sum_j W_{sij}^{(1)} h_{sj}^{(1)} + b_{s_i}, \sigma_{s_i}^2\right) \end{aligned} \tag{78}$$

where $\delta(x) = 1/(1+\exp(-x))$ is the logistic function.

Figure 4.2 illustrates a subset of training shapes together with samples generated from shape model after 10-step Gibbs sampling. From this, one can see that the shape model is able to capture the overall shape structure as well as a wide range of head poses and expressions.

4.3 Texture modeling

As opposed to facial shapes, the appearance of human face usually varies drastically due to numerous factors such as identities, lighting conditions, facial occlusions, expressions, image resolutions, etc. These factors can significantly change pixel values presented in these textures and result in much higher non-linear variations. Therefore, the process of texture modeling is more complicated and requires the texture model to be sophisticated enough to represent these variations.

The structure of texture model is represented in the right part of DAM in Figure 4.1. Different from the shape model which directly works with landmark coordinates in image domain $\mathcal{I}$, the given facial image is first warped from $\mathcal{I}$ to texture domain $\mathcal{D}$ using a reference candidate obtained from the training data. Then the obtained shape-free image is vectorized and used as the visible units for texture model. The purpose of warping step is to remove the effect of shape factors from the texture model and, therefore, making it more robust to shape changes during modeling process. Specificially, given an image I, the texture $\mathbf{g}$ is computed as

$$\mathbf{g} = \mathrm{vec}\left(I(W(r_{\mathcal{D}},\mathbf{s}))\right) \tag{79}$$

where $\mathrm{vec}(\cdot)$ denotes the vectorization operator and $W(r_{\mathcal{D}}, \mathbf{s})$ is the warping operator defined as in Eqn. (5).

A two-layer DBM is then employed to model the distributions of texture feature represented in $\mathbf{g}$. Similar to shape model, the GRBM is used in the bottom layer while interations between hidden units in higher layers are formulated by a binary-binary RBM. The energy of the state $\{\mathbf{g}, \mathbf{h}_g^{(1)}, \mathbf{h}_g^{(2)}\}$ in facial texture modeling is given as in Eqn. (80) where $\{\mathbf{h}_g^{(1)}, \mathbf{h}_g^{(2)}\}$ denote the set of hidden units and $\theta_g = \{\mathbf{W}_g^{(1)}, \mathbf{W}_g^{(2)}, \boldsymbol{\sigma}_g^2, \mathbf{b}_g\}$ are the model parameters.

$$\begin{aligned} E(\mathbf{g}, \mathbf{h}_g^{(1)}, \mathbf{h}_g^{(2)}; \theta_g) =& \sum_k \frac{(g_k - b_{g_k})^2}{2\sigma_{g_k}^2} - \sum_{k,t} \frac{g_k}{\sigma_{g_k}} W_{gkt}^{(1)} h_{gt}^{(1)} \\ & - \sum_{t,v} h_{gt}^{(1)} W_{gtv}^{(2)} h_{gv}^{(2)} \end{aligned} \tag{80}$$

The probability of $\mathbf{g}$ assigned by the model is then computed as follows:

$$\begin{aligned} P(\mathbf{g}; \theta_g) &= \sum_{\mathbf{h}_g^{(1)}, \mathbf{h}_g^{(2)}} P(\mathbf{g}, \mathbf{h}_g^{(1)}, \mathbf{h}_g^{(2)}; \theta_g) \\ &= \frac{1}{Z(\theta_g)} \sum_{\mathbf{h}_g^{(1)}, \mathbf{h}_g^{(2)}} e^{-E(\mathbf{g}, \mathbf{h}_g^{(1)}, \mathbf{h}_g^{(2)}; \theta_g)} \end{aligned} \tag{81}$$

The conditional distributions over $\mathbf{g}$, $\mathbf{h}_g^{(1)}$, and $\mathbf{h}_g^{(2)}$ are derived similar to those of shape model as in Eqn. (82). Figure 4.3 illustrates a subset of training texture as well as the learned feature obtained using the first layer of the presented texture model.

$$\begin{aligned} p(h_{gt}^{(1)} | \mathbf{g}, \mathbf{h}_g^{(2)}) &= \delta\left(\sum_k W_{gkt}^{(1)} \frac{g_k}{\sigma_{g_k}} + \sum_v W_{gtv}^{(2)} h_{gv}^{(2)}\right) \\ p(h_{gv}^{(2)} | \mathbf{h}_g^{(1)}) &= \delta\left(\sum_t W_{gtv}^{(2)} h_{gt}^{(1)}\right) \\ g_k | \mathbf{h}_g^{(1)} &\sim \mathcal{N}\left(\sigma_{g_k} \sum_t W_{gkt}^{(1)} h_{gt}^{(1)} + b_{g_k}, \sigma_{g_k}^2\right) \end{aligned} \tag{82}$$

4.4 Appearance modeling

A straightforward way to extract model parameters for both shape and texture is to employ a weighted concatenation and apply a dimensional reduction method such as PCA. However, this

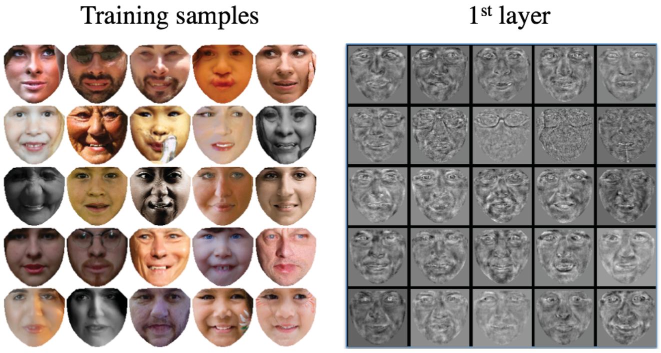


Figure 4.3: A subset of training faces and learned features of the first layer texture model.

is not an optimal solution since these parameters are presented in different domains, i.e. shape parameters $\boldsymbol{\alpha}_s$ determine the coordinates of landmark points while texture parameters $\boldsymbol{\alpha}_g$ present facial appearance in the texture domain $\mathcal{D}$. Therefore, the gaps between them still exist in the final model parameters although weight values are employed to balance the combined features.

Meanwhile, our Deep Appearance Models also aim to produce a robust facial shape and texture representation. It, however, can be considered as the problem of data learning from multiple sources. In this problem, the information learned from multiple input channels can complement each other and boost the overall performance of the whole model. Particularly, captions and tags can be used to improve the classification accuracy [49, 86, 114].

In order to generate a robust feature in DAM, one should notice that the hidden units are powerful in term of increasing the flexibility of deep model. Beside the ability of capturing different factors from the observations, the higher layer these hidden units are in, the more independent of the specific correlations of an input source [114]. Therefore, we can use them as a source-free representation. From that reason, we construct one more high-level layer to interpret the connections between face shape and its texture. Since $\mathbf{h}_s^{(2)}$ and $\mathbf{h}_g^{(2)}$ are independent of the spaces where the coordinates and appearance are in, the new layer can encode the shape and texture information more naturally and effectively.

Let $\mathbf{h}^{(3)}$ be the connection layer and $\theta = \{\theta_s, \theta_g\}$. Then the energy of the joint configuration $\{\mathbf{s}, \mathbf{g}, \mathbf{h}_s^{(1)}, \mathbf{h}_s^{(2)}, \mathbf{h}_g^{(1)}, \mathbf{h}_g^{(2)}, \mathbf{h}^{(3)}\}$ in DAM is defined as the summation of three energy functions of shape model, texture model and the joint layer.

$$
\begin{aligned}
E(\mathbf{s}, \mathbf{g}, \mathbf{h}_s, \mathbf{h}_g; \theta) = &\sum_i \frac{(s_i - b_{s_i})^2}{2\sigma_{s_i}^2} - \sum_{i,j} \frac{s_i}{\sigma_{s_i}} W_{sij}^{(1)} h_{sj}^{(1)} - \sum_{j,l} h_{sj}^{(1)} W_{sjl}^{(2)} h_{sl}^{(2)} \\
&+ \sum_k \frac{(g_k - b_{g_k})^2}{2\sigma_{g_k}^2} - \sum_{k,t} \frac{g_k}{\sigma_{g_k}} W_{gkt}^{(1)} h_{gt}^{(1)} - \sum_{t,v} h_{gt}^{(1)} W_{gtv}^{(2)} h_{gv}^{(2)} \\
&- \sum_{l,n} h_{sl}^{(2)} W_{sln}^{(3)} h_n^{(3)} - \sum_{v,n} h_{gv}^{(2)} W_{gvn}^{(3)} h_n^{(3)}
\end{aligned}
\tag{83}
$$

where $\mathbf{h}_s = \{\mathbf{h}_s^{(1)}, \mathbf{h}_s^{(2)}\}$ and $\mathbf{h}_g = \{\mathbf{h}_g^{(1)}, \mathbf{h}_g^{(2)}\}$. The joint distribution over the multimodal input can be written as:

$$
P(\mathbf{s}, \mathbf{g}; \theta) = \sum_{\mathbf{h}_s^{(2)}, \mathbf{h}_g^{(2)}, \mathbf{h}^{(3)}} P(\mathbf{h}_s^{(2)}, \mathbf{h}_g^{(2)}, \mathbf{h}^{(3)}) \left(\sum_{\mathbf{h}_s^{(1)}} P(\mathbf{s}, \mathbf{h}_s^{(1)}, \mathbf{h}_s^{(2)}) \right) \left(\sum_{\mathbf{h}_g^{(1)}} P(\mathbf{g}, \mathbf{h}_g^{(1)}, \mathbf{h}_g^{(2)}) \right) \tag{84}
$$

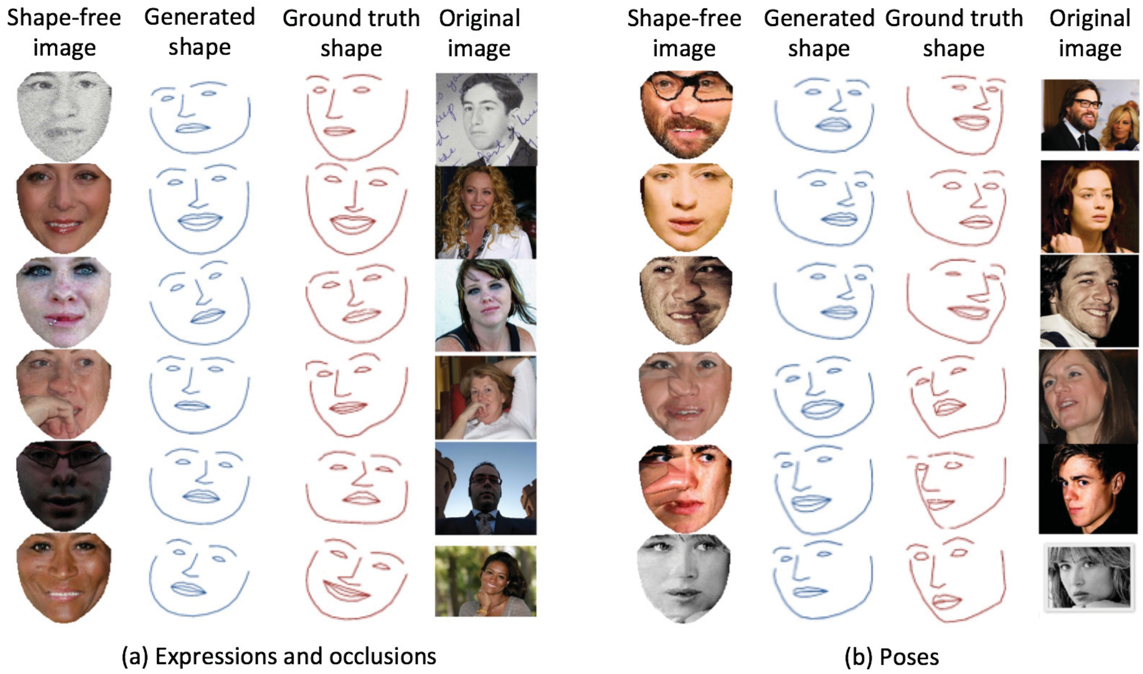
and the conditional distributions over $\mathbf{h}_s^{(2)}$, $\mathbf{h}_g^{(2)}$, and $\mathbf{h}^{(3)}$ are derived as

$$
\begin{aligned}
p(h_{sl}^{(2)} | \mathbf{h}_s^{(1)}, \mathbf{h}^{(3)}) &= \delta \left(\sum_j W_{sjl}^{(2)} h_{sj}^{(1)} + \sum_n W_{sln}^{(3)} h_n^{(3)} \right) \\
p(h_{gv}^{(2)} | \mathbf{h}_g^{(1)}, \mathbf{h}^{(3)}) &= \delta \left(\sum_t W_{gtv}^{(2)} h_{gt}^{(1)} + \sum_n W_{gvn}^{(3)} h_n^{(3)} \right) \\
p(h_n^{(3)} | \mathbf{h}_s^{(2)}, \mathbf{h}_g^{(2)}) &= \delta \left(\sum_l W_{sln}^{(3)} h_{sl}^{(2)} + \sum_v W_{gvn}^{(3)} h_{gv}^{(2)} \right)
\end{aligned}
\tag{85}
$$

Other conditional distributions over $\mathbf{s}$, $\mathbf{g}$, $\mathbf{h}_s^{(1)}$ and $\mathbf{h}_g^{(1)}$ are the same as in Eqns. (78) and (82).

4.5 Properties of Deep Appearance Models

Deep Appearance Models provide the capability of generating facial shapes using texture information and vice versa. For example, one can predict a facial shape from the appearance using DAM as follows: (1) clamping the texture information $\mathbf{g}$ as observations for the texture model and initializing hidden units with random states; (2) performing standard Gibbs sampling as a posterior inference step; and (3) obtaining the reconstructed shape from $P(\mathbf{s}|\mathbf{g}; \theta)$. To generate the appearance from a given shape, one can apply the same way with reversed pathways after clamping the shape information to the shape model. Figure 4.4 represents the generated shapes given textures



(a) Expressions and occlusions

(b) Poses

Figure 4.4: Facial shape generation using texture information with (a) expressions and occlusions; and (b) poses. In both cases, given the shape-free image (first column), DAM are able to generated the facial shape (second column) by sampling from $P(\mathbf{s}|\mathbf{g},\theta)$. The ground truth shapes and original images are also given in the third and fourth columns, respectively.

in three cases of expressions, occlusions and poses. In all these cases, the DAM model is able to predicted the shape correctly.

In addition, it is more natural to interpret both shapes and textures using higher hidden layers. In order to obtain this representation, one can clamp both observed shape $\mathbf{s}$ and texture $\mathbf{g}$ together before applying the Gibbs sampling procedure to estimate $P(\mathbf{h}^{(3)}|\mathbf{s},\mathbf{g};\theta)$. Eventually, probabilities of these hidden layers can be used as features. Notice that, beside the advantage of better features for discriminative tasks, one can easily see that even when one of two inputs is missing (i.e. shape), $P(\mathbf{h}^{(3)}|\mathbf{g};\theta)$ is still able to approximate. Hence, DAM can be considered as a more generative model compared to other appearance models.

The proposed method can also deal with facial reconstruction in various challenging conditions, such as: facial occlusions, facial expressions, facial off-angles, etc.

4.6 Model Learning

The parameters in the model are optimized in order to maximize the log likelihood

$$\theta^* = \arg\max_{\theta} \log P(\mathbf{s}, \mathbf{g}; \theta) \tag{86}$$

Then the optimal parameter values can be obtained in a gradient descent fashion given by

$$\frac{\partial}{\partial \theta} \mathbb{E} \left[\log P(\mathbf{s}, \mathbf{g}; \theta) \right] = \mathbb{E}_{\text{data}} \left[\frac{\partial E}{\partial \theta} \right] - \mathbb{E}_{\text{model}} \left[\frac{\partial E}{\partial \theta} \right] \tag{87}$$

where $\mathbb{E}_{\text{data}} [\cdot]$ and $\mathbb{E}_{\text{model}} [\cdot]$ are the expectations with respect to data distribution, i.e. *data-dependent expectation*, and distribution estimated by Deep Appearance Models, i.e. *model's expectation*. The former term can be approximated by mean-field inference while the latter term can be estimated using Markov-chain Monte-Carlo (MCMC) based stochastic approximation.

Computing Data-dependent Expectation: Mean-field approximation can be used to compute the first term of Eqn. (87) [107]. The main idea of this technique comes from the variational approach where the lower bound of the log-likelihood is maximized with respect to the variational parameters $\boldsymbol{\mu}$. In the mean-field approximation, for each training face with its shape and texture $\mathbf{s}, \mathbf{g}$, all visible units corresponding to $\mathbf{s}$ and $\mathbf{g}$ are fixed and the states of hidden units in the models are set to $\boldsymbol{\mu}$ which are iteratively updated through layers using mean-field fixed-point equations:

$$\begin{aligned}
\mu_{sj}^{(1)} &\leftarrow \delta \left(\sum_i W_{sij}^{(1)} \frac{s_i}{\sigma_{s_i}} + \sum_l W_{sjl}^{(2)} \mu_{sl}^{(2)} \right) \\
\mu_{sl}^{(2)} &\leftarrow \delta \left(\sum_j W_{sjl}^{(2)} \mu_{sj}^{(1)} + \sum_n W_{sln}^{(3)} \mu_{n}^{(3)} \right) \\
\mu_{gt}^{(1)} &\leftarrow \delta \left(\sum_k W_{gkt}^{(1)} \frac{g_k}{\sigma_{g_k}} + \sum_v W_{gtv}^{(2)} \mu_{gv}^{(2)} \right) \\
\mu_{gv}^{(2)} &\leftarrow \delta \left(\sum_t W_{gtv}^{(2)} \mu_{gt}^{(1)} + \sum_n W_{gvn}^{(3)} \mu_{n}^{(3)} \right) \\
\mu_{n}^{(3)} &\leftarrow \delta \left(\sum_l W_{sln}^{(3)} \mu_{sl}^{(2)} + \sum_v W_{gvn}^{(3)} \mu_{gv}^{(2)} \right)
\end{aligned} \tag{88}$$

Using these variational parameters, the data-dependent statistics are then computed by averaging over training cases.

Computing Expectation of the Model: For the second term of Eqn. (87), the MCMC sampling can be applied [108]. Specifically, given the current state of visible and hidden units, their new states are obtained by employing a few steps of persistent Gibbs sampling using Eqns. (78), (82) and (85). Then $\mathbb{E}_{\text{model}}[\cdot]$ is approximated by the expectations with respect to new states of the model units.

4.7 Fitting in Deep Appearance Models

In this section, two fitting methods are proposed to synthesize DAM to new given face images: the forward composition based fitting and the dictionary learning based fitting.

4.7.1 Forward Composition Based Fitting

Given a testing face I, the fitting process in DAM can be formulated as finding an optimal shape $\mathbf{s}$ that maximizes the probability of the shape-free image as in Eqn. (89).

$$\mathbf{s}^* = \arg\max_{\mathbf{s}} P(I(W(r_{\mathcal{D}}, \mathbf{s}))|\mathbf{s}; \theta) \tag{89}$$

Since the connections between textures and hidden units $\mathbf{h}_g^{(1)}$ are modeled by a Gaussian Restricted Boltzmann Machines, the probability of texture $\mathbf{g}$ given hidden units $\mathbf{h}_g^{(1)}$ is computed as follows:

$$P(\mathbf{g}|\mathbf{h}_g^{(1)}; \mathbf{s}, \theta) = \mathcal{N}(\sigma_g \mathbf{W}_g^{(1)} \mathbf{h}_g^{(1)} + \mathbf{b}_g, \sigma_g^2 \mathbf{A}) \tag{90}$$

where $\mathbf{A}$ is the identity matrix; $\{\sigma_g, \mathbf{b}_g\}$ are the standard-deviation and bias of visible units in the texture model; and $\mathbf{W}_g^{(1)}$ are learned weights of the visible-hidden texture.

During the fitting steps, the states of hidden units $\mathbf{h}_g^{(1)}$ are estimated by clamping both the current shape $\mathbf{s}$ and the texture $\mathbf{g}$ to the model. The Gibbs sampling method is then applied to find the optimal estimated texture of the testing face given a current shape $\mathbf{s}$. By this way, the hidden units in DAM can take into account both shape and texture information in order to reconstruct a better texture for further refinement. Let $\mathbf{m} = \sigma_g \mathbf{W}_g^{(1)} \mathbf{h}_g^{(1)} + \mathbf{b}_g$ be the mean of the Gaussian distribution, we have the following approximation:

$$P(I(W(r_{\mathcal{D}}, \mathbf{s}))|\mathbf{h}_g^{(1)}; \theta) = \mathcal{N}(\mathbf{m}, \sigma_g^2 \mathbf{A}) \tag{91}$$

The maximum likelihood can be then estimated as follows:

$$\begin{aligned} \mathbf{s}^* &= \arg\max_{\mathbf{s}}(P(I(W(r_{\mathcal{D}}, \mathbf{s}))|\mathbf{s}; \theta)) \\ &= \arg\max_{\mathbf{s}} \mathcal{N}(I(W(r_{\mathcal{D}}, \mathbf{s}))|\mathbf{m}, \sigma_g^2 \mathbf{A})) \\ &= \arg\min_{\mathbf{s}} \frac{1}{\sigma_g^2} \sum (I(W(r_{\mathcal{D}}, \mathbf{s})) - \mathbf{m})^2 \end{aligned} \tag{92}$$

Then the forward compositional algorithm can be used to solve the Eqn. (92) by finding the updating parameter $\Delta\mathbf{s}$ that increases the likelihood:

$$\Delta\mathbf{s} = \arg\min_{\Delta\mathbf{s}} \|I(W(W(r_{\mathcal{D}}, \Delta\mathbf{s}), \mathbf{s})) - \mathbf{m}\|^2 \tag{93}$$

The linearization is taken place of the test image coordinate using first order Taylor expansion $I(W(W(r_{\mathcal{D}}, \Delta\mathbf{s}), \mathbf{s})) = I(W(r_{\mathcal{D}}, \mathbf{s})) + \mathbf{J}_I \Delta\mathbf{s}$ and the update parameter is given as:

$$\Delta\mathbf{s} = -(\mathbf{J}_I^T \mathbf{J}_I)^{-1} \mathbf{J}_I^T \left[I(W(r_{\mathcal{D}}, \mathbf{s})) - \mathbf{m}) \right] \tag{94}$$

where $\mathbf{J}_I = \nabla I \frac{\partial W}{\partial \mathbf{s}}$ is the Jacobian.

4.7.2 Dictionary Learning based Fitting

In this section, we further improve the fitting process so that it can deal with occlusions and other variations. From Eqn. (93), we can see that the shape update $\Delta\mathbf{s}$ mostly relies on the difference between the shape-free image and its DAM reconstruction. However, this metric is easily affected by the presence of occlusions. When part of the face is occluded, the occlusion is removed in DAM reconstruction but still remained in the shape-free image as the result of warping operator. Therefore, even when the current shape is the ground truth one, the gap between the two images is still large. As a result, the ℓ_2-norm of their difference is still not robust enough to guide the fitting process to the true shape when occlusions occur.

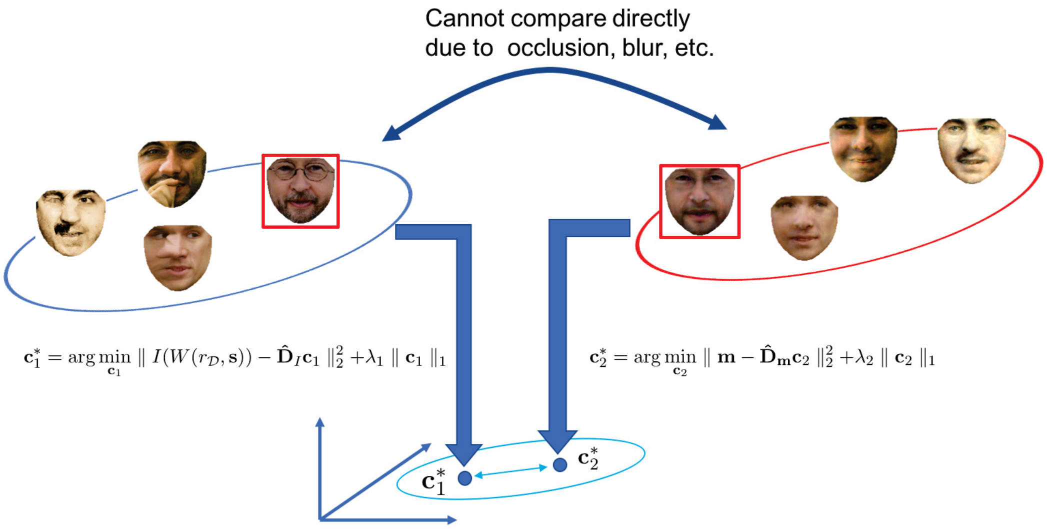

Cannot compare directly
due to occlusion, blur, etc.
$\mathbf{c}_1^* = \arg\min_{\mathbf{c}_1} \| I(W(r_{\mathcal{D}}, \mathbf{s})) - \hat{\mathbf{D}}_I \mathbf{c}_1 \|_2^2 + \lambda_1 \| \mathbf{c}_1 \|_1$
$\mathbf{c}_2^* = \arg\min_{\mathbf{c}_2} \| \mathbf{m} - \hat{\mathbf{D}}_{\mathbf{m}} \mathbf{c}_2 \|_2^2 + \lambda_2 \| \mathbf{c}_2 \|_1$
$\mathbf{c}_1^*$
$\mathbf{c}_2^*$

Figure 4.5: An illustration of Dictionary Learning Fitting for DAM.

To address this problem more effectively, instead of working directly in texture space, we define a function f such that the relationship between $f(I(W(r_{\mathcal{D}}, \mathbf{s})))$ and $f(\mathbf{m})$ is more robust to occlusions. Then this relationship can be used for fitting process.

The function f can be defined as

$$\begin{aligned} f(I(W(r_{\mathcal{D}}, \mathbf{s}))) &= \mathbf{c}_1^* \\ f(\mathbf{m}) &= \mathbf{c}_2^* \end{aligned} \tag{95}$$

where

$$\begin{aligned} \mathbf{c}_1^* &= \arg\min_{\mathbf{c}_1} \| I(W(r_{\mathcal{D}}, \mathbf{s})) - \hat{\mathbf{D}}_I \mathbf{c}_1 \|_2^2 + \lambda_1 \| \mathbf{c}_1 \|_1 \\ \mathbf{c}_2^* &= \arg\min_{\mathbf{c}_2} \| \mathbf{m} - \hat{\mathbf{D}}_{\mathbf{m}} \mathbf{c}_2 \|_2^2 + \lambda_2 \| \mathbf{c}_2 \|_1 \end{aligned} \tag{96}$$

and $\{\hat{\mathbf{D}}_I, \mathbf{c}_1\}$ and $\{\hat{\mathbf{D}}_{\mathbf{m}}, \mathbf{c}_2\}$ are the dictionaries and representation coefficients of the shape free image and its DAM reconstruction, respectively. Notice that when $\mathbf{s}$ is the ground truth shape, there should be a close connection between $f(I(W(r_{\mathcal{D}}, \mathbf{s})))$ and $f(\mathbf{m})$ even when the face is occluded. For that reason, we set $\mathbf{c} = \mathbf{c}_1 = \mathbf{c}_2$ and use ground truth shape to learn the dictionary $\{\hat{\mathbf{D}}_I, \hat{\mathbf{D}}_{\mathbf{m}}\}$. Then the DAM fitting can be decomposed into two steps, i.e. training and testing. Figure 4.5

illustrates the idea of dictionary learning for DAM fitting.

Training step: Given a training dataset with N images and their shapes $(\{I^i, \mathbf{s}^i\})_{i=1}^N$, the dictionaries are learned by minimizing the loss function

$$\begin{aligned}\{\hat{\mathbf{D}}_I, \hat{\mathbf{D}}_\mathbf{m}\} = \arg \min_{\mathbf{D}_I, \mathbf{D}_\mathbf{m} \in \mathbb{R}^{k \times l}} \frac{1}{N} \sum_{i=1}^{N} \{ \min_{\mathbf{c}^i \in \mathbb{R}^l} & \parallel \mathbf{I}_W^i - \mathbf{D}_I \mathbf{c}^i \parallel_2^2 \\ & + \parallel \mathbf{m}^i - \mathbf{D}_\mathbf{m} \mathbf{c}^i \parallel_2^2 + \\ & + \lambda \parallel \mathbf{c}^i \parallel_1 \}\end{aligned} \tag{97}$$

where $\mathbf{I}_W^i = I^i(W(r_\mathcal{D}, \mathbf{s}))$, k is the length of texture vector and l is the size of dictionaries. To solve this problem, we apply the four-step iterative procedure as in [136]. The main steps of this procedure are summarized in Algorithm 5. There are two main advantages of learning the dictionaries as in Eqn. (97). Firstly, since both shape-free image and its DAM reconstruction are forced to share the same representation $\mathbf{c}^i$, their underlying relationships are naturally embedded in these coefficients. Secondly, when the coefficients vector $\mathbf{c}^i$ is sparse, the optimization will result in the most related features between the shape-free image and its reconstruction. Therefore, it will be more robust to occlusions and other variations.

Algorithm 5 Dictionary Learning for fitting

```
Input: Training data $\{(I^i, \mathbf{s}^i)\}_{i=1}^N$, regularization parameter $\lambda$
Output: Learned dictionaries $\{\hat{\mathbf{D}}_I, \hat{\mathbf{D}}_\mathbf{m}\}$
Construct the matrix $\mathbf{Y} \in \mathbb{R}^{k \times N}$ whose $i$-th column is shape-free image $\mathbf{I}_W^i$.
Construct the matrix $\mathbf{M} \in \mathbb{R}^{k \times N}$ whose $i$-th column is $\mathbf{m}^i$.
Initialize $\mathbf{D}_I \in \mathbb{R}^{k \times l}$ and $\mathbf{D}_\mathbf{m} \in \mathbb{R}^{k \times l}$ with random samples from a normal distribution with
    zero mean and unit variance.
while not converged do
    (1) Fix $\mathbf{D}_\mathbf{m}$, learn $\mathbf{D}_I$ and coefficient matrix $\mathbf{C} \in \mathbb{R}^{l \times N}$
            $\{\mathbf{D}_I, \mathbf{C}\} = \arg \min_{\mathbf{D}_I, \mathbf{C}} \parallel \mathbf{Y} - \mathbf{D}_I \mathbf{C} \parallel_2^2 + \lambda \parallel \mathbf{C} \parallel_1$
    (2) Update $\mathbf{D}_\mathbf{m}$ as $\mathbf{D}_\mathbf{m} = \mathbf{M}/\mathbf{C}$. Notice that this result is used as initial $\mathbf{D}_\mathbf{m}$ for step (3).
    (3) Fix $\mathbf{D}_I$, learn $\mathbf{D}_\mathbf{m}$ and new coefficient matrix $\mathbf{C}$
            $\{\mathbf{D}_\mathbf{m}, \mathbf{C}\} = \arg \min_{\mathbf{D}_\mathbf{m}, \mathbf{C}} \parallel \mathbf{M} - \mathbf{D}_\mathbf{m} \mathbf{C} \parallel_2^2 + \lambda \parallel \mathbf{C} \parallel_1$
    (4) Update $\mathbf{D}_I$ as $\mathbf{D}_I = \mathbf{Y}/\mathbf{C}$.
end while
Set $\hat{\mathbf{D}}_I = \mathbf{D}_I$ and $\hat{\mathbf{D}}_\mathbf{m} = \mathbf{D}_\mathbf{m}$.
```

After obtaining the dictionaries, instead of following the Gaussian-Newton optimization as in Eqn. (93), we learn a linear regressor to directly infer the shape update $\Delta\mathbf{s}$ from the difference between $f(I_W)$ and $f(\mathbf{m})$. Specificially, given training images with their initial estimate $\{\bar{\mathbf{s}}^i\}_{i=1}^N$ of their ground truth shape, the linear regressor is learned by minimizing

$$\arg\min_{\mathbf{H},\mathbf{b}} \sum_{i=1}^{N} \parallel \Delta\mathbf{s}^i - \mathbf{H}\left[f(I_W^i) - f(\mathbf{m}^i)\right] - \mathbf{b} \parallel^2 \tag{98}$$

where $\Delta\mathbf{s}^i = \mathbf{s}^i - \bar{\mathbf{s}}^i$; $\{(\mathbf{H}, \mathbf{b})\}$ are the regressor's parameters.

Testing step: Given an input face with its initial shape, the shape-free image I_W and DAM reconstruction $\mathbf{m}$ are first computed. Their representation coefficients $\mathbf{c}_1^*$ and $\mathbf{c}_2^*$ are also estimated using Eqn. (96). Then the shape is updated using the difference $\mathbf{c}_1^* - \mathbf{c}_2^*$ together with the learned regressor. After that the image pair (I_W, m) is recomputed for the next iteration.

4.8 Discussion

In order to learn the dictionary $\mathbf{D}_I$ and $\mathbf{D}_\mathbf{m}$ (step 5 and 6 in Algorithm 5), beside K-SVD [2] which is one of the most popular techniques, one can use some other discriminative dictionary learning methods such as [52, 131, 141]. These methods have achieved good performance particularly for image classification. This is related to my study of sparse representation and dictionary learning for handwritten character recognition published in [87]. An extension of these two learning dictionary steps is to use the ℓ_p-norm instead of ℓ_1-norm. By this way, the obtained coefficients could be more robust when dealing with occlusions (or missing values), poses (noise), the work of using ℓ_p-norm in Robust PCA has been published in [98].

In summary, this chapter has presented the structure of DAM as well as the fitting step to synthesize any given new face image. Its performance in several face modeling tasks will be shown in chapter 8. In the next chapter, a robust version of DAM is proposed to extend its capability of handling face occlusion. A new texture modeling approach is developed on top of DAM structure and makes DAM not only be able to distinguish between "good" and "bad" face regions but also enjoyed improvements on both reconstruction and fitting processes.

Chapter 5

Robust Deep Appearance Models For Texture Modeling

Dealing with face occlusions, numerous approaches have been proposed in literature [50, 51, 121, 140]. Among them, Robust Principal Component Analysis (RPCA) [18] can be considered as one of the most common approaches. The main idea of RPCA is to decompose an observed matrix (i.e. represents a set of faces of a subject) into two components: low-rank component (i.e. common information of the subject's face) and sparse component (i.e. corrupted or contiguous occlusion and other variations including illumination and expression). Inspired by the idea of RPCA, in [96], we introduced a novel face recognition system based on learning low-rank matrix and sparse variation representation to improve the system performance under various affecting conditions. Figure 5.1 illustrates the main architecture of this system. This system has shown its advantages and achieved a performance boost compared to other *classical* face recognition methods. However, having similar issues as PCA, RPCA is still limited in its generalization capabilities. More importantly, it requires a set of considerable number of input images to be able to separate the non-occluded faces and occlusions.

This chapter introduced the Robust Deep Appearance Models (RDAM), a robust generative deep model, that can separate unwanted factors while preserving identity information given an input face image. Comparing to DAM, RDAM can produce remarkable reconstruction results even when faces

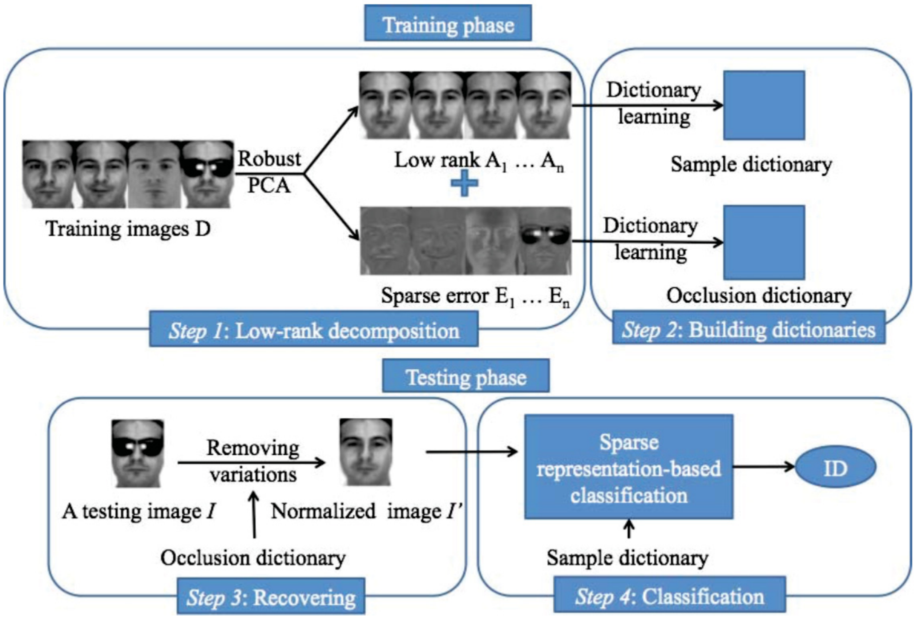


Figure 5.1: The proposed Face Recognition system [96] by decomposing the input faces into non-occluded faces and occlusions.

are occluded or having extreme poses. Moreover, the proposed fitting algorithms fit well with the new texture model such that it can make use of the occlusion mask generated by the proposed model. Some materials of this chapters have been published in [96, 97].

5.1 RDAM structure

Similar to DAM, as illustrated in Figure 5.2, the structure of RDAM also consists of three main components: two prior models for shape and texture and a high-level hidden layer for appearance modeling. Unlike the texture model of DAM, this model consists of a visible layer with three gating components: $\mathbf{g}$, $\tilde{\mathbf{g}}$, and $\bar{\mathbf{m}}$, a binary RBM for the mask variable $\bar{\mathbf{m}}$ and a Gaussian DBM with the real-valued input variable $\mathbf{g}$. The motivation for using this gating term is to improve modeling and fitting of the DAM by eliminating the effects of missing, occluded or corrupted pixels. In next section, the details of constructing the texture model with gating component and fitting algorithms are presented.

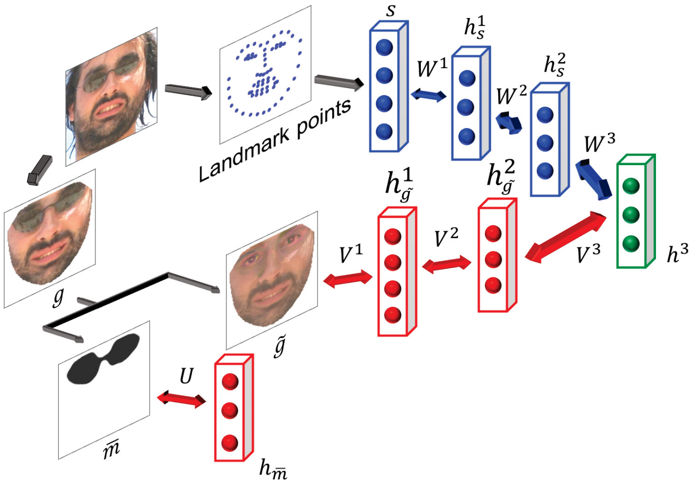


Figure 5.2: Robust Deep Appearance Models that contains shape model (top), texture model with "clean" texture and an occlusion mask (bottom), and a joint representation (i.e. appearance) of shape and "clean" texture.

5.2 Texture Modeling

Given a shape-free image $\mathbf{g}$, the energy function of the configuration $\{\mathbf{g}, \tilde{\mathbf{g}}, \bar{\mathbf{m}}, \mathbf{h}_{\bar{m}}, \mathbf{h}_{\tilde{g}}^{(1)}, \mathbf{h}_{\tilde{g}}^{(2)}\}$ in facial texture modeling is optimized as follows:

$$\begin{aligned} E_{RDBM_g}(\mathbf{g}, \tilde{\mathbf{g}}, \bar{\mathbf{m}}, \mathbf{h}_{\bar{m}}, \mathbf{h}_{\tilde{g}}^{(1)}, \mathbf{h}_{\tilde{g}}^{(2)}; \theta_g) = & \sum_i \frac{\gamma_i^2 \bar{m}_i (g_i - \tilde{g}_i)^2}{2\sigma_{\tilde{g}_i}^2} \\ & - \sum_{i,k} U_{ik} \bar{m}_i h_{\bar{m}k} + \sum_i \frac{(g_i - b_{g_i})^2}{2\sigma_{g_i}^2} \\ & + \sum_i \frac{(\tilde{g}_i - b_{\tilde{g}_i})^2}{2\sigma_{\tilde{g}_i}^2} - \sum_{i,j} W_{\tilde{g}ij}^{(1)} \tilde{g}_i h_{\tilde{g}j}^{(1)} - \sum_{j,l} W_{\tilde{g}jl}^{(2)} h_{\tilde{g}j}^{(1)} h_{\tilde{g}l}^{(2)} \end{aligned} \tag{99}$$

where $\theta_g = \{\mathbf{W}_{\tilde{g}}^{(1)}, \mathbf{W}_{\tilde{g}}^{(2)}, \mathbf{U}, \sigma_g, \mathbf{b}_g, \sigma_{\tilde{g}}, \mathbf{b}_{\tilde{g}}\}$ are the texture model parameters. It is noted that all the bias terms in Eqn. (99) are ignored for simplicity. The probability distribution of the configuration $\{\mathbf{g}, \tilde{\mathbf{g}}, \bar{\mathbf{m}}, \mathbf{h}_{\bar{m}}, \mathbf{h}_{\tilde{g}}^{(1)}, \mathbf{h}_{\tilde{g}}^{(2)}\}$ is computed as follow:

$$P(\mathbf{g}; \theta_g) = \sum_{\mathbf{h}_{\tilde{g}}^{(1)}, \mathbf{h}_{\tilde{g}}^{(2)}} \frac{\exp\left(-E_{RDBM_g}\left(\mathbf{g}, \tilde{\mathbf{g}}, \bar{\mathbf{m}}, \mathbf{h}_{\bar{m}}, \mathbf{h}_{\tilde{g}}^{(1)}, \mathbf{h}_{\tilde{g}}^{(2)}; \theta_g\right)\right)}{Z(\theta_a)} \tag{100}$$

Given an input $\mathbf{g}$, the states of all layers can be inferred by computing the posterior probability of the latent variables, i.e. $p(\tilde{\mathbf{g}}, \bar{\mathbf{m}}, \mathbf{h}_{\bar{m}}, \mathbf{h}_{\tilde{g}}^{(1)}, \mathbf{h}_{\tilde{g}}^{(2)} | \mathbf{g})$. Therefore, the sampling can be divided into two folds, i.e. one for the visible units and one for the hidden units. For the visible variables $\tilde{\mathbf{g}}$ and $\bar{\mathbf{m}}$,

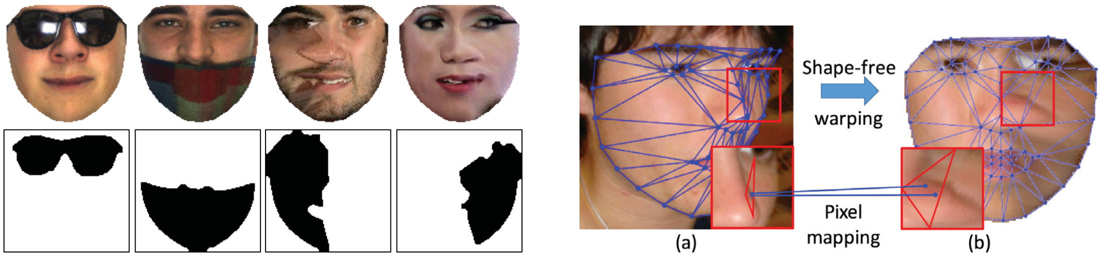


Figure 5.3: **LEFT**: Examples of automatically detected masks from the shape-free images. Top row: shape-free images. Bottom row: detected binary masks using the technique in section 5.3, **RIGHT**: An illustration in pose stretching detection: (a) Source image (b) Target warped shape-free image

the conditional distributions can be sampled as,

$$p(\tilde{\mathbf{g}}, \bar{\mathbf{m}}|\mathbf{h}_{\bar{m}}, \mathbf{h}_{\tilde{g}}^{(1)}, \mathbf{g}) = p(\tilde{\mathbf{g}}|\bar{\mathbf{m}}, \mathbf{h}_{\tilde{g}}^{(1)}, \mathbf{g})p(\bar{\mathbf{m}}|\mathbf{h}_{\bar{m}}, \mathbf{h}_{\tilde{g}}^{(1)}, \mathbf{g}) \tag{101}$$

For the hidden variables $\mathbf{h}_{\bar{m}}, \mathbf{h}_{\tilde{g}}^{(1)}, \mathbf{h}_{\tilde{g}}^{(2)}$, the conditional distributions can be sampled as follows,

$$p(\mathbf{h}_{\bar{m}}, \mathbf{h}_{\tilde{g}}^{(1)}, \mathbf{h}_{\tilde{g}}^{(2)}|\tilde{\mathbf{g}}, \bar{\mathbf{m}}, \mathbf{g}) = p(\mathbf{h}_{\bar{m}}|\bar{\mathbf{m}})p(\mathbf{h}_{\tilde{g}}^{(1)}|\tilde{\mathbf{g}}, \mathbf{h}_{\tilde{g}}^{(2)})p(\mathbf{h}_{\tilde{g}}^{(2)}|\mathbf{h}_{\tilde{g}}^{(1)}) \tag{102}$$

The sampling process can be applied on each unit separately since the distribution is factorial. Section 5.4 will discuss the learning procedure of this texture model.

5.3 Learning Binary Mask RBM

This section aims to generate masks from the training images having poses and occlusions, e.g. sunglasses and scarves. We consider learning three types of binary mask, i.e. sunglasses, scarves and pose stretching. A binary RBM is learned to represent each type of mask. We will focus on the last type, i.e. pose stretching since it is the hardest.

In 2D texture model, warping faces with a large pose (e.g. larger than ±45°) will likely cause stretching effects on half of the faces since the same pixel values are copied over a large region (see Fig. 5.3-RIGHT). Therefore, we propose a technique that can detect such stretching regions during warping process. The main idea is to count the number of unique pixels in the source triangle that are mapped to the pixels in the target triangle. As we know, a source pixel can be mapped to multiple target pixels due to interpolation. The degree of a target triangle being stretched is equivalent to $p = (\frac{n_0}{N})$, where $p = 1$ means there is no stretching, n_0 and N are the number of unique pixels and the total number of pixels in the corresponding source triangle, respectively. Finally, we can use the detected regions as a mask to pre-train the above robust texture model.

5.4 Model Learning

To train our presented texture model, a DBM is first trained with only "clean" images, i.e. without occlusions, and then the parameters in this texture model are optimized to maximize the log likelihood as follows,

$$\theta_g^* = \arg\max_{\theta_g} \log P(\mathbf{g}; \theta_g) \tag{103}$$

The optimal parameter values can then be obtained using a gradient descent procedure given by,

$$\frac{\partial}{\partial\theta_g}\mathbb{E}\left[\log P(\mathbf{g};\theta_g)\right] = \mathbb{E}_{P_{\text{data}}}\left[\frac{\partial E_{\text{RDBM}_g}}{\partial\theta_g}\right] - \mathbb{E}_{P_{\text{model}}}\left[\frac{\partial E_{\text{RDBM}_g}}{\partial\theta_g}\right] \tag{104}$$

where $\mathbb{E}_{P_{\text{data}}}[\cdot]$ and $\mathbb{E}_{P_{\text{model}}}[\cdot]$ are the expectations respecting to data distribution and distribution estimated by the RDBM. The two terms can be approximated using mean-field inference and Markov Chain Monte Carlo (MCMC) based stochastic approximation, respectively.

In our method, pre-training the parameters of the DBM on "clean" data first will make the process of learning the texture model faster and much easier. Similarly, we also propose to first learn the parameters of the binary RBM (to represent the mask $\bar{\mathbf{m}}$) on pre-defined and extracted masks (as shown in Fig.5.3-LEFT) instead of randomizing the parameters. Then, the next question is how to generate the training masks from the training set. An automatic technique is presented to extract such training masks for the binary RBM in the section 5.3.

5.5 Fitting in Robust Deep Appearance Models

Similar to DAM, the fitting in RDAM is formulated as finding the optimal shape $\mathbf{s}$ that maximizes the probability of the "clean" shape-free images. Then the optimization process is to solve the Eqn. (92). Since our proposed model can generate a mask of corrupted pixels, we propose to incorporate the mask $\bar{\mathbf{m}}$ into the original objective function in Eqn. (92) as:

$$\mathbf{s}^* = \arg\min_{\mathbf{s}} \|\bar{\mathbf{m}} \odot (I(\mathbf{W}(r_{\mathcal{D}}, \mathbf{s})) - \tilde{\mathbf{g}})\|^2 \tag{105}$$

where $\odot$ is the component-wise multiplication. The modified forward additive, forward compositional and inverse compositional algorithms are introduced in the next three sub-sections.

5.5.1 Forward Additive Algorithm

Forward Additive algorithm, also known as Lucas-Kanade algorithm, was first proposed for image alignment by Lucas and Kanade [70]. The idea of the algorithm is to find the best warp parameters that minimize the sum of squares error between a fixed template image and an input image I when warped. The warp parameters are iteratively updated by adding $\Delta\mathbf{s}$ each time, thus, the algorithm is considered as an *additive* approach. Using this idea, we solve the problem in Eqn. (105) by linearizing it and then solve it iteratively with respect to an increment of the parameters $\Delta\mathbf{s}$. Then we minimize the following:

$$\Delta\mathbf{s} = \arg\min_{\Delta\mathbf{s}} \|\bar{\mathbf{m}} \odot (I(\mathbf{W}(r_{\mathcal{D}}, \mathbf{s})) + \mathbf{J}_I \Delta\mathbf{s} - \tilde{\mathbf{g}})\|^2 \tag{106}$$

where $\mathbf{J}_I = \nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{s}}$ is the Jacobian matrix of the image I.

The first step is to optimize (106) with respect to $\Delta\mathbf{s}$ and then update $\mathbf{s} \rightarrow \mathbf{s} + \Delta\mathbf{s}$. This gives us the following:

$$\Delta\mathbf{s} = \mathbf{H}^{-1}\mathbf{J}_I^T(\bar{\mathbf{m}} \odot (I(\mathbf{W}(r_{\mathcal{D}}, \mathbf{s})) - \tilde{\mathbf{g}})) \tag{107}$$

where the Hessian matrices **H** are given by

$$\mathbf{H} = (\bar{\mathbf{m}} \odot \mathbf{J}_I)^T(\bar{\mathbf{m}} \odot \mathbf{J}_I) \tag{108}$$

In general, the computations of Hessian and Jacobian matrices are the costliest steps and they need to be re-computed at each iteration. Thus, the Lucas-Kanade algorithm is slow. The modified Forward Additive algorithm with the use of a mask $\bar{\mathbf{m}}$ is summarized in Algorithm 6.

Algorithm 6 − Forward Additive

```
1. Pre-compute: the gradient, the Jacobian and the Hessian matrix need to be recomputed at each iteration.

2. At each iteration:
    (I) Perform warping operator W to obtain warped texture $I(\mathbf{W}(r_{\mathcal{D}}, \mathbf{s}))$
    (II) Compute the texture reconstruction error $(\bar{\mathbf{m}} \odot I(\mathbf{W}(r_{\mathcal{D}}, \mathbf{s})) - \tilde{\mathbf{g}})$
    (III) Compute $\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{s}}(\bar{\mathbf{m}} \odot I(\mathbf{W}(r_{\mathcal{D}}, \mathbf{s})) - \tilde{\mathbf{g}})$
    (IV) Compute the Hessian matrix using Eqn. (108)
    (IV) Compute $\Delta s$ using Eqn. (107)
    (IV) Update new shape as $\mathbf{s} \rightarrow \mathbf{s} + \Delta\mathbf{s}$
```

```
Algorithm 7 − Forward Compositional
1. Pre-compute: The Jacobian $\frac{\partial \mathbf{W}}{\partial \mathbf{s}}$ at $(r_{\mathcal{D}}; 0)$

2. At each iteration:
    (I) Perform warping operator $\mathbf{W}$ to obtain warped texture $I(\mathbf{W}(r_{\mathcal{D}}, \mathbf{s}))$
    (II) Compute the texture reconstruction error $(\bar{\mathbf{m}} \odot I(\mathbf{W}(r_{\mathcal{D}}, \mathbf{s})) - \tilde{\mathbf{g}})$
    (III) Compute $\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{s}} (\bar{\mathbf{m}} \odot I(\mathbf{W}(r_{\mathcal{D}}, \mathbf{s})) - \tilde{\mathbf{g}})$
    (IV) Compute $\Delta s$ using equation (107)
    (V) Update the shape parameters by composing the warp operator $\mathbf{s} \rightarrow \mathbf{s} \circ \Delta\mathbf{s}^{-1}$
```

5.5.2 Forward Compositional Algorithm

For computing the warp parameters, the forward additive or Lucas-Kanade algorithm estimates a small offset from the current warp parameters. In the compositional algorithms, the composition of an incremental warp and the current warp is computed instead. Applying to our problem in (105), we have the following minimization problem:

$$\Delta\mathbf{s} = \arg\min_{\Delta\mathbf{s}} \|\bar{\mathbf{m}} \odot (I(\mathbf{W}(\mathbf{W}(r_{\mathcal{D}}, \Delta\mathbf{s}), \mathbf{s})) - \tilde{\mathbf{g}})\|^2 \tag{109}$$

The forward compositional algorithm can be used to solve the above problem (109) by first linearizing the image I around $\mathbf{s}$. An update $\Delta\mathbf{s}$ is found using least-squares, and $\mathbf{s}$ is updated from $\mathbf{s} \leftarrow \mathbf{s} \circ \Delta\mathbf{s}$, where $\circ$ denotes the composition of two warps. Noting that the algorithm is processed with occluded/missing data being ignored while computing the residual error. The linearization applied to the test image side via first order Taylor expansion gives us:

$$\Delta\mathbf{s} = \arg\min_{\Delta\mathbf{s}} \|\bar{\mathbf{m}} \odot (I(\mathbf{W}(\mathbf{W}(r_{\mathcal{D}}, 0), \mathbf{s})) + \mathbf{J}_I \Delta\mathbf{s} - \tilde{\mathbf{g}})\|^2 \tag{110}$$

When $\mathbf{s} = 0$, we have an identity warp, i.e. $\mathbf{W}(r_{\mathcal{D}}, 0) = r_{\mathcal{D}}$. The key difference between forward additive and forward compositional is that the Jacobian $\partial\frac{\mathbf{W}}{\partial\mathbf{s}}$ is computed at $(r_{\mathcal{D}}, 0)$. Thus, it is a constant and can be pre-computed. Not having to compute the Jacobian $\partial\frac{\mathbf{W}}{\partial\mathbf{s}}$ in each iteration reduces the computational cost despite that the compositional update step is costlier.

5.5.3 Inverse Compositional Algorithm

The inverse compositional algorithm is a modification of the forward compositional algorithm where the roles of the model image and testing image are reversed. The incremental warp is computed with respect to the model image $\tilde{\mathbf{g}}$ instead of with respect to $I(\mathbf{W}(r_{\mathcal{D}}, \mathbf{s}))$. Thus, changing

Algorithm 8 $-$ Inverse Compositional

```
1. Pre-compute: The gradient $\nabla\tilde{\mathbf{g}}$,the Jacobian $\frac{\partial \mathbf{W}}{\partial \mathbf{s}}$ at $(r_{\mathcal{D}};0)$, the steepest descent $SD = \nabla\tilde{\mathbf{g}}\frac{\partial \mathbf{W}}{\partial \mathbf{s}}$, the Hessian matrix $H = SD^T SD$

2. At each iteration:
    (I) Perform warping operator $\mathbf{W}$ to obtain warped texture $I(\mathbf{W}(r_{\mathcal{D}}, \mathbf{s}))$
    (II) Compute the texture reconstruction error $(\bar{\mathbf{m}} \odot I(\mathbf{W}(r_{\mathcal{D}}, \mathbf{s})) - \tilde{\mathbf{g}})$
    (III) Compute $\nabla\tilde{\mathbf{g}}\frac{\partial \mathbf{W}}{\partial \mathbf{s}}(\bar{\mathbf{m}} \odot I(\mathbf{W}(r_{\mathcal{D}}, \mathbf{s})) - \tilde{\mathbf{g}})$
    (IV) Compute $\Delta s$ using equation (112)
    (V) Update the shape parameters by composing the warp operator $\mathbf{s} \rightarrow \mathbf{s} \circ \Delta\mathbf{s}^{-1}$
```

the roles of $I(\mathbf{W}(r_{\mathcal{D}}, \mathbf{s}))$ and $\tilde{\mathbf{g}}$ in Eqn. (110) gives us the inverse compositional algorithm by minimizing:

$$\Delta\mathbf{s} = \arg\min_{\Delta\mathbf{s}} \|\bar{\mathbf{m}} \odot (I(\mathbf{W}(r_{\mathcal{D}}, \mathbf{s})) - \tilde{\mathbf{g}}(\mathbf{W}(r_{\mathcal{D}}, \Delta\mathbf{s})))\,\|^2 \tag{111}$$

with respect to $\Delta\mathbf{s}$ and then updating the parameters as $\mathbf{s} \leftarrow \mathbf{s} \circ \Delta\mathbf{s}^{-1}$. The solution of the least squares problem in Eqn. (111) is:

$$\Delta\mathbf{s} = \mathbf{H}^{-1}\mathbf{J}_{\tilde{\mathbf{g}}}^T(\bar{\mathbf{m}} \odot (I(\mathbf{W}(r_{\mathcal{D}}, \mathbf{s})) - \tilde{\mathbf{g}})) \tag{112}$$

where $\mathbf{J}_{\tilde{\mathbf{g}}} = \nabla\tilde{\mathbf{g}}\frac{\partial \mathbf{W}}{\partial \mathbf{s}}$ is the Jacobian matrix of the model image $\tilde{\mathbf{g}}$. The Hessian matrices $\mathbf{H}$ are then given by $\mathbf{H} = (\bar{\mathbf{m}} \odot \mathbf{J}_{\tilde{\mathbf{g}}})^T(\bar{\mathbf{m}} \odot \mathbf{J}_{\tilde{\mathbf{g}}})$.

5.6 Discussion

In summary, this chapter has introduced a texture model that enable the ability of separating the “clean” and occluded pixels for DAM. By incorporating these information, the robustness of the model in both modeling and fitting processes can be improved significantly. Since DAM and RDAM mainly focus on the pixel distribution and their relationships within single image, in the next chapter, we further extend our exploration to temporal relationship, i.e. the third dimension, between face images in a sequence. We then apply this proposed architecture to model the longitudinal face sequence. The new model is also able to predict the future of a sequence from its past.

Chapter 6

Temporal Restricted Boltzmann Machines for Longitudinal Face Modeling

In this chapter, we firstly present a brief overview about the longitudinal face modeling task with a specific application to face age progression. Then the Temporal Restricted Boltzmann Machines based age progression model is proposed to efficiently capture the non-linear aging process and automatically synthesize a series of age-progressed faces in various age ranges. Some materials of this chapters have been published in [89].

6.1 Introduction

Face age progression presents the capability to predict future faces of an individual in input photos. In most cases, there is only one photo of that individual and we have to predict the future faces, i.e. age progression, or construct the former faces, i.e. age regression or deaging, of that subject [3]. Face aging can find its origins from missing children when police require age progressed pictures. This problem is also applicable in cases of wanted fugitives where face age progression is also required. The predominant approach to aging pictures involves the use of forensic artists [126]. Although forensic artists are trained in the anatomy and geometry of faces, they still can suffer from psycho-cognitive bias that may affect their interpretation of the source face data. In addition, an age-progressed image can differ significantly from one forensic artist to the next. Manual age progression usually takes lots of time and requires the work of numerous professional forensic artists. Therefore, automatic and computerized age-progression systems are important. Their applications range from very sensitive national security problems to tobacco or alcohol stores/bars to control the patron's age and cosmetic studies against aging.

Synthesizing plausible faces of individuals at different stages in their life is an extremely challenging task, even for human, due to several reasons. Firstly, human face aging is a complicated process since people usually age in different ways. It is non-deterministic and greatly depends on intrinsic factors, i.e. gender, ethnicity and heredity. Moreover, extrinsic factors, i.e. environment, living styles and smoking, have also created various effects to the facial changes and resulted in large aging variations even between people in the same age group. Secondly, facial shapes and textures dramatically change over the long periods. Thirdly, it is very hard to collect a longitudinal face age database that is generative enough to learn an aging model. Currently existing aging databases in the research community are small or unbalanced among genders, ethnicities and age groups. In addition, they are usually mixed with other variations, e.g. expressions and illuminations. Figure 6.1 illustrates some examples of the wide-range input images of six subjects with their ages and the their real faces at the target ages. We also show the results obtained by our proposed approach.

Automatic face age progression has attracted huge interest from the computer vision community in recent years. There are numerous efforts to model the longitudinal aging process presented in computer vision literature [34, 57, 63, 92, 117]. In most conventional methods, linear models, e.g. Active Appearance Models (AAM) and 3D Morphable Model, are usually adopted to interpret the geometry and appearance of the faces before the aging rules are learned. However, the face aging variations are not only large but also non-linear. It apparently violates the assumption of linear models. Therefore, these age-progression methods meet a lot of difficulties and limitations to interpret these non-linear aging variations.

Recently, Temporal Restricted Boltzmann Machines (TRBM) [118, 124, 144] have gained attention significantly as one of the probabilistic models to accurately model complex time-series structure while keeping the inference tractable. As an extension of Restricted Boltzmann Machines

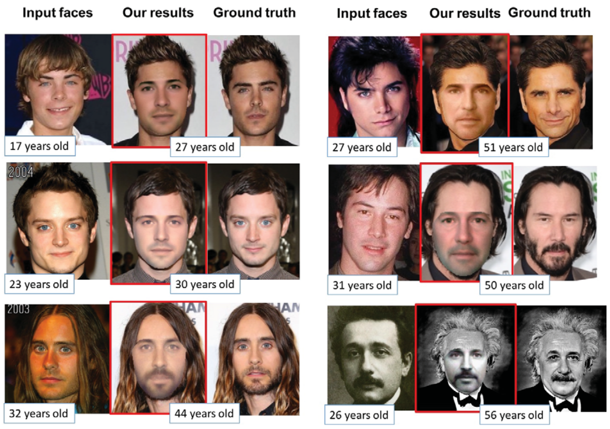


Figure 6.1: Examples of age progression using our proposed approach. Each subject has three images: the input image (left), the synthesized age-progressed face (middle), and the ground truth (right). Our system also can predict the ages of input faces in case these ground-truths are not available.

(RBM), the structure of TRBM consists of further directed connections from previous states of visible and hidden units. By this way, the short history of their activations can act as "memory" and is able to contribute to the inference step of visible units. In this structure, multiple factors are learned and interacted to efficiently explain the temporal data. Therefore, TRBM provides the ability to extract more complicated and nonlinear structures in time series data.

This work presents a novel deep model based approach to face age progression. Instead of synthesizing faces directly from long periods, the long-term aging process is considered as a set of short-term changes and presented using a sequence of faces. The TRBM based model is then constructed to capture the aging transformation between consecutive faces in the sequence. In addition, to enforce the model on the capabilities of aging variations, a set of reference faces that are mainly different in age conditions is generated and incorporated into the model. Then, a set of RBMs based wrinkle models is developed to enhance the wrinkle details in these aging faces. Finally, the facial geometric information of each age group is extracted and adopted to adjust the face shapes. Figure 6.2 illustrates the main processing steps of our proposed system.

The novelties of our approach are :

- The face structure and specific aging features presented in each age group are modeled using

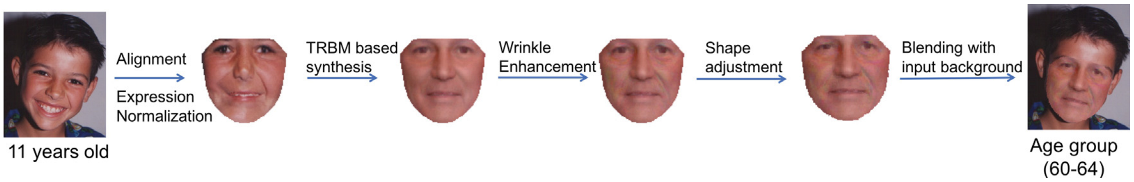


Figure 6.2: Processing steps of our proposed method to synthesize the face at ages of 60s given a face at age of 10-14

RBM. Compared to other linear models, the use of RBMs can help to better interpret the non-linear variations and produce faces with more aging details. In addition, the high-level features extracted from hidden layer can be transferred between RBMs of different age groups for reconstructing a reference face sequence that can benefit the learning process.

- Together with the reference sequence, the proposed TRBM based model provides an efficient way to capture the aging transformation between faces in different age groups. Similar to RBM, TRBM is more advanced in interpreting the complex and non-linear aging process.
- Far apart from previous approaches where wrinkles are cloned from an average face or the closest faces of each age group, we propose a machine learning based approach to learn these aging rules, i.e. construct a set of RBMs based wrinkle models for every age group. In this way, the method is able to learn their distributions and generate synthetic wrinkles by sampling from these distributions. As a result, our model is more flexible in producing more wrinkle types.
- The geometric differences between face shapes in every age group are also taken into account in our system.
- A large-scale dataset named AginG Faces in the Wild (AGFW) is collected for analysing the aging effects.

Our proposed age progression system (as shown in Figure 6.3(B)) consists of five main steps: (1) Preprocessing, (2) Reference sequence generation; (3) Texture age progression; (4) Wrinkles enhancement; and (5) Shape adjustment.

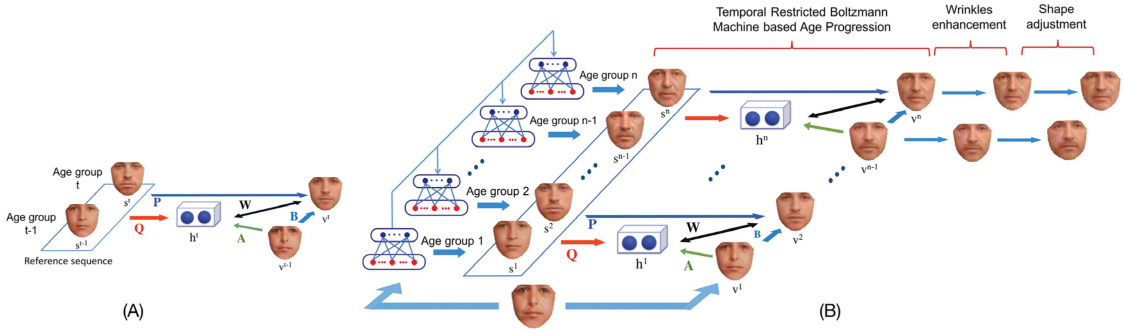


Figure 6.3: The proposed age progression approach: (A) Temporal Restricted Boltzmann Machines for learning aging transformation in a single node; (B) The proposed system using multiple nodes; wrinkle enhancement and shape adjustment.

6.2 Preprocessing

Face Alignment: In order to align all face images in the dataset, a reference shape is extracted from a selected subset of 2,000 face images in the passport style photos, i.e. frontal faces without expressions. All face images in the AGFW dataset are then warped to the texture domain corresponding to this reference shape. The warping step aims to remove the effects of shape variations during the texture modeling step. Finally, we obtain the dense correspondence between all faces in the training data. The DLIB tool [59] is employed to extract 68 landmarks for each face and the Procrustes Analysis is used to align these face images.

Expression Normalization: The expressions in the images of each age group are further normalized using the Collection Flow technique [56].

6.3 Reference Sequence Generation

This section presents how to generate the set of reference faces that are mainly different in age conditions.

6.3.1 Baseline

A straightforward approach to construct the reference sequence is to order the mean faces of all age groups chronologically. The advantage of using mean faces is that several variations such as

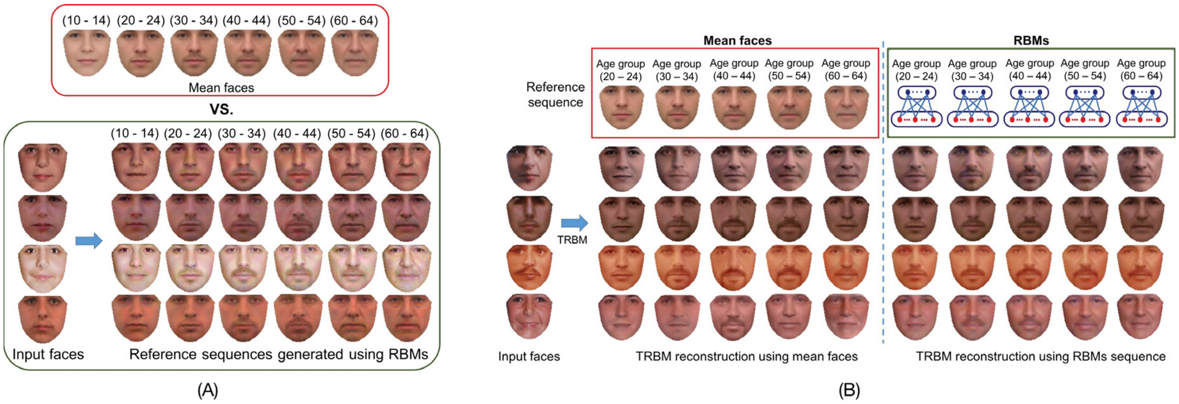


Figure 6.4: A comparison between (A) two approaches to generate reference sequences and (B) synthesized aging faces using these two reference sequences. Faces in the red box: the sequence of mean faces in several age groups. Faces in the green box: reference faces generated by transferring features among RBMs of these age groups. Given input images in the age range of 10-14, our system automatically synthesizes a sequence of age-progressed images in various age ranges respectively.

identity, occlusion can be removed. However, due to the averaging property, the aging variation is also smoothed out in the mean faces. Therefore, mean faces usually look younger than those from their own age groups. Moreover, it is noted that the lighting presented in the mean faces could be remarkably different from that of the input face. Figure 6.4(A) shows the unmatched tones between the sequence of mean faces and the input faces.

6.3.2 Our Improvement using RBM

Given an input face I at a particular age, instead of using the set of mean faces in all age groups as the reference sequence, a set of RBMs is constructed to model faces in different age groups. The high-level features are then transferred among RBMs to generate the reference faces for I.

In particular, for each age group k, all images collected at that age group are used to construct an RBM to model the distributions of texture features presented in this age group. Since the texture data is real-valued, the Gaussian-Bernoulli RBM (GRBM) is employed. Once RBMs of all age groups are constructed, given an input face image, its high-level features are first extracted using the RBM of the corresponding age group. These features are then transferred to the hidden layers of other RBMs to reconstruct the faces of other age groups. Gibbs sampling technique is used for this reconstruction stage.

There are several advantages of using RBMs in this step. Firstly, RBMs can help to model faces in more details comparing to mean faces. Secondly, since each RBM is built for a particular age group, it has the ability to generalize the faces with specific aging features. Therefore, transferring the high-level features between RBMs can generate new faces that consist of both original subject and new aging features. Thirdly, the lighting has implicitly corrected during the reconstruction process. Figure 6.4(A) illustrates the sequence of mean faces and the RBMs reconstructions by transferring features in six age groups.

6.4 Modeling the Aging Transformation via TRBM

In order to learn the aging transformation between faces in the sequence, we employ a TRBM with Gaussian visible units. As illustrated in Figure 6.3(A), the model consists of two sets of visible units (i.e. $\mathbf{v}^t, \mathbf{v}^{t-1}$) encoding the texture of current face at age group t and previous face at age group $t-1$; and a set of binary hidden units $\mathbf{h}^t$ that are latent variables. In addition, the faces in reference sequence, $\mathbf{s}^{<=t} = \{\mathbf{s}^t, \mathbf{s}^{t-1}\}$, at age group t and $t-1$ are also incorporated by the connections to both hidden and visible units.

The energy of the joint configuration $\{\mathbf{v}^t, \mathbf{h}^t\}$ is formulated as follows.

$$\begin{aligned} E(\mathbf{v}^t, \mathbf{h}^t | \mathbf{v}^{t-1}, \mathbf{s}^{<=t}; \boldsymbol{\theta}) = & \sum_i \frac{(v_i^t - b_i^t)^2}{2\sigma_i^2} - \sum_j h_j^t a_j^t \\ & - \sum_{i,j} \frac{v_i^t}{\sigma_i} W_{ij} h_j^t \end{aligned} \tag{113}$$

where $\boldsymbol{\theta} = \{\mathbf{W}, \mathbf{A}, \mathbf{B}, \mathbf{P}, \mathbf{Q}, \boldsymbol{\sigma}^2, \mathbf{b}^t, \mathbf{a}^t\}$ are the model parameters. In particular, $\{\mathbf{W}, \mathbf{A}, \mathbf{B}, \mathbf{P}, \mathbf{Q}\}$ are the weights of connections as illustrated in Figure 6.3(A); $\{\boldsymbol{\sigma}^2, \mathbf{b}^t, \mathbf{a}^t\}$ are the variance, bias of the visible units and bias of the hidden units, respectively. Notice that the form of this energy function is very similar to the original form of an RBM. However, the bias terms are redefined as:

$$b_i^t = b_i + B_i \mathbf{v}^{t-1} + \sum_l P_{li} \mathbf{s}_l^{<=t} \tag{114}$$

$$a_j^t = a_j + A_j \mathbf{v}^{t-1} + \sum_l Q_{lj} \mathbf{s}_l^{<=t} \tag{115}$$

where l is the index of reference faces in sequence $\mathbf{s}^{<=t}$.

The probability of $\mathbf{v}^t$ assigned by the model is given by

$$\begin{aligned} p(\mathbf{v}^t|\mathbf{v}^{t-1}, \mathbf{s}^{<=t}; \boldsymbol{\theta}) &= \sum_{\mathbf{h}^t} p(\mathbf{v}^t, \mathbf{h}^t|\mathbf{v}^{t-1}, \mathbf{s}^{<=t}; \boldsymbol{\theta}) \\ &= \frac{1}{Z} \sum_{\mathbf{h}^t} e^{-E(\mathbf{v}^t, \mathbf{h}^t|\mathbf{v}^{t-1}, \mathbf{s}^{<=t}; \boldsymbol{\theta})} \end{aligned} \tag{116}$$

where Z is the partition function. The probability of a sequence with T faces given the first face and the reference sequence $\mathbf{s}^{1:T}$ is defined as Eqn. (117).

$$p(v^{2:T}|\mathbf{v}^1, \mathbf{s}^{1:T}; \boldsymbol{\theta}) = \prod_{t=2}^{T} p(\mathbf{v}_t|\mathbf{v}^{t-1}, \mathbf{s}^{<=t}; \boldsymbol{\theta}) \tag{117}$$

The conditional distributions over $\mathbf{v}^t$ and $\mathbf{h}^t$ are given as

$$\begin{aligned} p(h_j^t = 1|\mathbf{v}^t, \mathbf{v}^{t-1}, \mathbf{s}^{<=t}) &= \sigma \left(\sum_i W_{ij} \frac{v_i^t}{\sigma_i} + a_j^t \right) \\ v_i^t|\mathbf{h}^t, \mathbf{v}^{t-1}, \mathbf{s}^{<=t} &\sim \mathcal{N} \left(\sigma_i \sum_j W_{ij} h_j^t + b_i^t, \sigma_i^2 \right) \end{aligned} \tag{118}$$

6.4.1 Model Properties

With this structure, two types of information can be learned from the model:

(1) The temporal information presented in the relationship between previous face $\mathbf{v}^{t-1}$ and the current face $\mathbf{v}^t$.

(2) The aging information provided by the reference sequence. This type of information acts as guidance information enforcing the model to learn the aging differences rather than other variations.

Moreover, in order to transfer the information between faces, both linear and nonlinear interactions are employed in this model. In particular, $\mathbf{v}^{t-1}$ and $\mathbf{v}^t$ are connected via two pathways: (1) the linear and direct connections using weight matrix $\mathbf{B}$; and (2) the nonlinear connections through the latent variables $\mathbf{h}^t$ with the weight matrices $\mathbf{A}$ and $\mathbf{W}$. Similar to the relationship between $\mathbf{v}^t$ and $\mathbf{s}^{<=t}$, the direct (with weight matrix $\mathbf{P}$) and indirect (with weights $\mathbf{Q}$ and $\mathbf{W}$) connections allow both linear and nonlinear interactions. Notice that except the undirected connections between

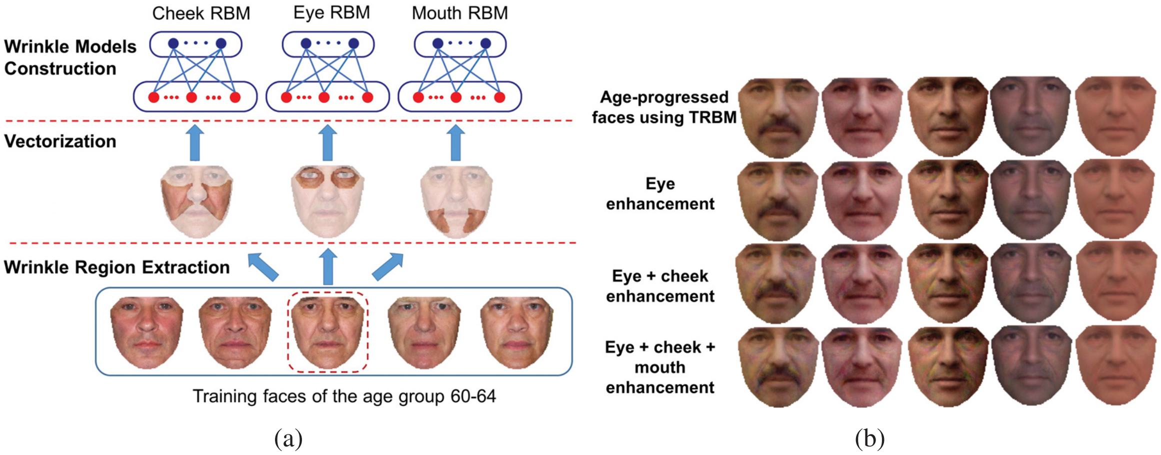

Cheek RBM
Eye RBM
Mouth RBM
Wrinkle Models Construction
Vectorization
Wrinkle Region Extraction
Training faces of the age group 60-64
(a)
Age-progressed faces using TRBM
Eye enhancement
Eye + cheek enhancement
Eye + cheek + mouth enhancement
(b)

Figure 6.5: (a) **Wrinkle Model Construction Steps**. (b) **Wrinkle Enhancement**. From top to bottom: the synthesized images from the previous step, the results after enhancing eye; eye and cheek; eye, cheek and mouth regions.

hidden units $\mathbf{h}^t$ and visible units $\mathbf{v}^t$, all connections are directed.

6.4.2 Model Learning

The learning process is to find the model parameters that maximize the log-likelihood:

$$\theta^* = \arg\max_{\theta} \sum_{t=2}^{T} \log p(\mathbf{v}_t | \mathbf{v}^{t-1}, \mathbf{s}^{<=t}; \boldsymbol{\theta}) \tag{119}$$

The optimal parameter values can then be obtained via a gradient descent procedure given by

$$\frac{\partial}{\partial \theta} \mathbb{E}\left[\log p(\mathbf{v}_t | \mathbf{v}^{t-1}, \mathbf{s}^{<=t}; \boldsymbol{\theta})\right] = \sum_{t=2}^{T} \mathbb{E}_{\text{data}}\left[\frac{\partial E}{\partial \theta}\right] - \mathbb{E}_{\text{model}}\left[\frac{\partial E}{\partial \theta}\right] \tag{120}$$

where $\mathbb{E}_{\text{data}}[\cdot]$ and $\mathbb{E}_{\text{model}}[\cdot]$ are the expectations with respect to data distribution and distribution estimated by the TRBM model. The Contrastive Divergence [43] is used for the learning process.

6.5 RBM based Wrinkle Modeling

Since facial muscles play an important role on the changes of wrinkle appearance during aging process, we make use of the anatomical evidence for wrinkles enhancement. In particular, inspiring from the analysis on the behaviors of facial muscles [101], we select the muscles that are more relevant to wrinkle appearance and use their physical positions to extract the wrinkle subregions from the face image. Three chosen subregions are shown in Figure 6.5a. A set of RBMs is then employed to learn the distributions of wrinkle appearance for every age group.

Once RBMs for all subregions and age groups are learned, the wrinkles are enhanced via a two-step process: (1) Generating the wrinkles through a Gibbs sampling process with the learned distributions; and (2) Wrinkle rendering by blending the generated wrinkles with the synthesized faces obtained from the TRBM based texture progression step. The Poisson blending technique [94] is used for seamless fusion results. Figure 6.5b shows the wrinkles enhancement results in three wrinkle regions.

6.6 Shape Adjustment

To further take into account the changes of shape during aging process, for each age group, we compute the average face shape using the same pipeline as in Section 6.2 with the AGFW dataset. Then the synthesized faces obtained from the previous step are warped to the corresponding face shapes for the final age-progressed result.

6.7 Discussion

This chapter has developed a novel deep model based approach for face age progression that can operate in the wild. With the deep structured models for both face representation and aging transformation modeling, the proposed model can efficiently capture the non-linear aging changes as well as robustly handle other variations such as pose, expressions, and illuminations. The aging rules in terms of wrinkle appearance and geometric constraints are also taken into account for more consistent progression results. Similar to other RBM based approach, the training process of the proposed model still needs some approximation due to the issue when evaluating the intractable partition function. In the next chapter, we further propose a novel deep generative probabilistic model for age progression. This new modeling approach enjoys the strengths of both probabilistic graphical models to produce better image synthesis quality and deep residual networks (ResNet) [42] to improve the highly non-linear feature generation.

Chapter 7

Temporal Non-Volume Preserving Approach for Facial Age-Progression and Age Invariant Face Recognition

With the advantages of probabilistic graphical models, the Temporal Restricted Boltzmann Machines (TRBM) based model (presented in previous chapter) has shown its potential in the age progression task [89]. However, its partition function is intractable and needs some approximations during training process. Other Recurrent Neural Networks (RNN) based approach is also introduced to model the intermediate states between two consecutive age groups for better aging transition [132]. However, it still has the limitations of producing blurry results by the use of a fixed reconstruction loss function, i.e. ℓ_2-norm. In this chapter, we design a novel generative probabilistic model, named Temporal Non-Volume Preserving (TNVP) transformation, for age progression. This modeling approach enjoys the strengths of both probabilistic graphical models to produce better image synthesis quality by avoiding the regular reconstruction loss function, and deep residual networks (ResNet) [42] to improve the highly non-linear feature generation. The proposed TNVP guarantees a *tractable* log-likelihood density estimation, *exact* inference and evaluation for embedding the feature transformations between faces in consecutive age groups. As illustrated in Figure

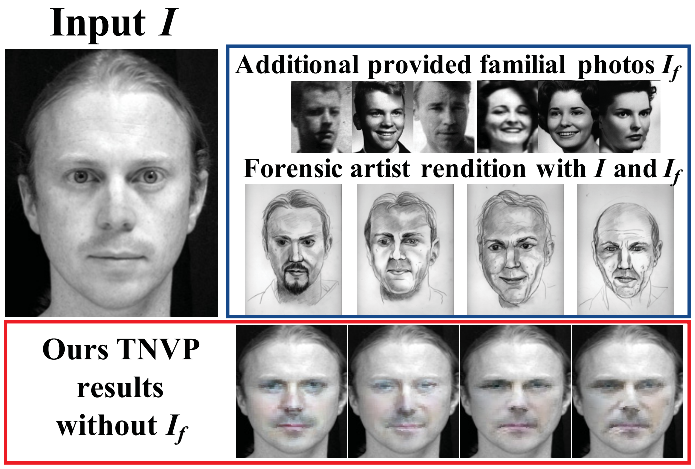

Input I
Additional provided familial photos I_f
Forensic artist rendition with I and I_f
Ours TNVP results without I_f

Figure 7.1: An illustration of age progression from forensic artist and our TNVP model. Given an input I of a subject at 34 years old [93], a forensic artist rendered his age-progressed faces at 40s, 50s, 60s and 70s by reference to his familial photos I_f. Without using I_f, our TNVP can aesthetically produce his age-progressed faces.

7.1, given a face of a subject at the age of 34 [93], a set of closely related family faces has to be provided to a forensic artist as references to generate multiple outputs of his faces at 40s, 50s, 60s, and 70s. The bottom row shows our corresponding synthesized-faces achieved by the TVNP method. Materials of this chapters have been published in [90].

7.1 TNVP architecture

In our framework, the long-term face aging is first considered as a composition of short-term stages. Then our TNVP models are constructed to capture the facial aging features transforming between two successive age groups. By incorporating the design of ResNet [42] based Convolutional Neural Network (CNN) layers in the structure, our TNVP is able to efficiently capture the non-linear facial aging feature related variance. In addition, it can be robustly employed on face images in the wild without strict alignments or any complicated preprocessing steps. Finally, the connections between latent variables of our TNVP can act as “memory” and contribute to produce a smooth age progression between faces while preserving the identity throughout the transitions.

In summary, the novelties of our approach are three-fold. **(1)** We propose a novel generative probabilistic models with tractable density function to capture the non-linear age variances. **(2)**

Table 7.1: Comparing the properties between our TNVP approach and other age progression methods, where ✗ represents *unknown* or *not directly applicable* properties. Deep learning (DL), Dictionary (DICT), Prototype (PROTO), AGing pattErn Subspace (AGES), Composition (COMP), Probabilistic Graphical Models (PGM), Log-likelihood (LL), Adversarial (ADV)

	Our TNVP	TRBM [89]	RNN [132]	acGAN [6]	HFA [138]	CDL [111]	IAAP [57]	HAGES [127]	AOG [117]
Model Type	DL	DL	DL	DL	DICT	DICT	PROTO	AGES	COMP
Architecture	PGM + CNN	PGM	CNN	CNN	Bases	Bases	✗	✗	Graph
Loss Function	LL	LL	ℓ_2	ADV+ℓ_2	LL+ℓ_0	$\ell_2 + \ell_1$	✗	ℓ_2	✗
Tractable	✓	✗	✓	✓	✓	✓	✗	✓	✗
Non-Linearity	✓	✓	✓	✓	✗	✗	✗	✗	✗

The aging transformation can be effectively modeled using our TNVP. Similar to other probabilistic models, our TNVP is more advanced in term of embedding the complex aging process. **(3)** Unlike previous aging approaches that suffer from a burdensome preprocessing to produce the dense correspondence between faces, our model is able to synthesize realistic faces given any input face in the wild. Table 7.1 compares the properties between our TNVP approach and other age progression methods.

The proposed TNVP age-progression architecture consists of three main steps. (1) Preprocessing; (2) Face variation modeling via mapping functions; and (3) Aging transformation embedding. With the structure of the mapping function, our TNVP model is tractable and highly non-linear. It is optimized using a log-likelihood objective function that produces sharper age-progressed faces compared to the regular ℓ_2-norm based reconstruction models. Figure 7.2 illustrates our TNVP-based age progression architecture.

7.2 Preprocessing

Figure 7.3 compares our preprocessing step with other recent age progression approaches, including Illumination Aware Age Progression (IAAP) [57], RNN based [132], and TRBM based Age Progression [89] models. In those approaches, burdensome face normalization steps are applied to obtain the dense correspondence between faces. The use of a large number of landmark points

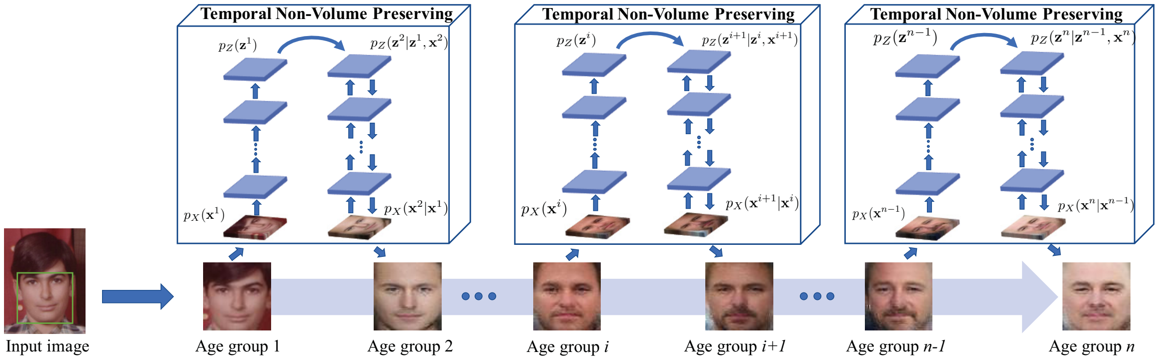

Temporal Non-Volume Preserving
$p_Z(\mathbf{z}^1)$
$p_Z(\mathbf{z}^2|\mathbf{z}^1, \mathbf{x}^2)$
$p_X(\mathbf{x}^1)$
$p_X(\mathbf{x}^2|\mathbf{x}^1)$
Temporal Non-Volume Preserving
$p_Z(\mathbf{z}^i)$
$p_Z(\mathbf{z}^{i+1}|\mathbf{z}^i, \mathbf{x}^{i+1})$
$p_X(\mathbf{x}^i)$
$p_X(\mathbf{x}^{i+1}|\mathbf{x}^i)$
Temporal Non-Volume Preserving
$p_Z(\mathbf{z}^{n-1})$
$p_Z(\mathbf{z}^n|\mathbf{z}^{n-1}, \mathbf{x}^n)$
$p_X(\mathbf{x}^{n-1})$
$p_X(\mathbf{x}^n|\mathbf{x}^{n-1})$
Input image
Age group 1
Age group 2
Age group *i*
Age group *i+1*
Age group *n-1*
Age group *n*

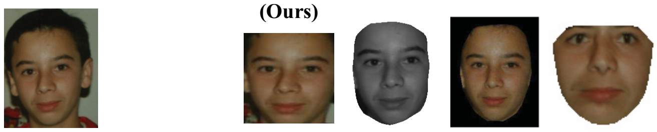
Figure 7.2: The proposed TNVP based age progression framework. The long-term face aging is decomposed into multiple short-term stages. Then given a face in age group i, our TNVP model is applied to synthesize face in the next age group. Each side of our TNVP is designed as a deep ResNet network to efficiently capture the non-linear facial aging features.

makes them highly dependent on the stability of landmarking methods that are challenged in the wild conditions. Moreover, masking the faces with a predefined template requires a separate shape adjustment for each age group in later steps.

In our method, given an image, the facial region is simply detected and aligned according to fixed positions of four landmark points, i.e. two eyes and two mouth corners. By avoiding complicated preprocessing steps, our proposed architecture has two advantages. Firstly, a small number of landmark points, i.e. only four points, leverages the dependency to the quality of any landmarking method. Therefore, it helps to increase the robustness of the system. Secondly, parts of the image background are still included, and thus it implicitly embeds the shape information during the modeling process. From the experimental results, one can easily notice the change of the face shape when moving from one age group to the next.

7.3 Face Aging Modeling

Let $\mathcal{I} \subset \mathbb{R}^D$ be the image domain and $\{\mathbf{x}^t, \mathbf{x}^{t-1}\} \in \mathcal{I}$ be observed variables encoding the texture of face images at age group t and $t-1$, respectively. In order to embed the aging transformation between these faces, we first define a bijection mapping function from the image space $\mathcal{I}$ to a latent space $\mathcal{Z}$ and then model the relationship between these latent variables. Formally, let $\mathcal{F} : \mathcal{I} \to \mathcal{Z}$ define a bijection from an observed variable $\mathbf{x}$ to its corresponding latent variable $\mathbf{z}$ and $\mathcal{G} : \mathcal{Z} \to \mathcal{Z}$



Input	TNVP (Ours)	IAAP	RNN	TRBM
Using Landmarks	**4 points**	10 points	66 points	68 points
Pose estimation	✗	✓	✓	✗
Dense correspondence	✗	✓	✓	✓
Masking Image	✗	✓	✓	✓
Expression Normalization	✗	✓	✓	✓

Figure 7.3: Comparisons between the preprocessing processes of our approach and other aging approaches: IAAP [57], RNN based [132], and TRBM based [89] models. Our preprocessing is easy to run, less dependent on the landmarking tools, and efficiently deals with in-the-wild faces. ✓represents "included in the preprocessing steps".

be an aging transformation function modeling the relationships between variables in latent space. As illustrated in Figure 7.4, the relationships between variables are defined as in Eqn. (121).

$$\begin{aligned}\mathbf{z}^{t-1} &= \mathcal{F}_1(\mathbf{x}^{t-1};\theta_1)\\ \mathbf{z}^{t} &= \mathcal{H}(\mathbf{z}^{t-1},\mathbf{x}^{t};\theta_2,\theta_3)\\ &= \mathcal{G}(\mathbf{z}^{t-1};\theta_3)+\mathcal{F}_2(\mathbf{x}^{t};\theta_2)\end{aligned} \tag{121}$$

where $\mathcal{F}_1, \mathcal{F}_2$ define the bijections of $\mathbf{x}^{t-1}$ and $\mathbf{x}^t$ to their latent variables, respectively. $\mathcal{H}$ denotes the summation of $\mathcal{G}(\mathbf{z}^{t-1};\theta_3)$ and $\mathcal{F}_2(\mathbf{x}^t;\theta_2)$. $\theta = \{\theta_1,\theta_2,\theta_3\}$ present the parameters of functions $\mathcal{F}_1, \mathcal{F}_2$ and $\mathcal{G}$, respectively. Indeed, given a face image in age group $t-1$, the probability density function can be formulated as in Eqn. (122).

$$\begin{aligned}p_{X^t}(\mathbf{x}^t|\mathbf{x}^{t-1};\theta) &= p_{X^t}(\mathbf{x}^t|\mathbf{z}^{t-1};\theta)\\ &= p_{Z^t}(\mathbf{z}^t|\mathbf{z}^{t-1};\theta)\left|\frac{\partial\mathcal{H}(\mathbf{z}^{t-1},\mathbf{x}^t;\theta_2,\theta_3)}{\partial\mathbf{x}^t}\right|\\ &= p_{Z^t}(\mathbf{z}^t|\mathbf{z}^{t-1};\theta)\left|\frac{\partial\mathcal{F}_2(\mathbf{x}^t;\theta_2)}{\partial\mathbf{x}^t}\right|\end{aligned} \tag{122}$$

where $p_{X^t}(\mathbf{x}^t|\mathbf{x}^{t-1};\theta)$ and $p_{Z^t}(\mathbf{z}^t|\mathbf{z}^{t-1};\theta)$ are the distribution of $\mathbf{x}^t$ conditional on $\mathbf{x}^{t-1}$ and the distribution of $\mathbf{z}^t$ conditional on $\mathbf{z}^{t-1}$, respectively. In Eqn. (122), the second equality is obtained using

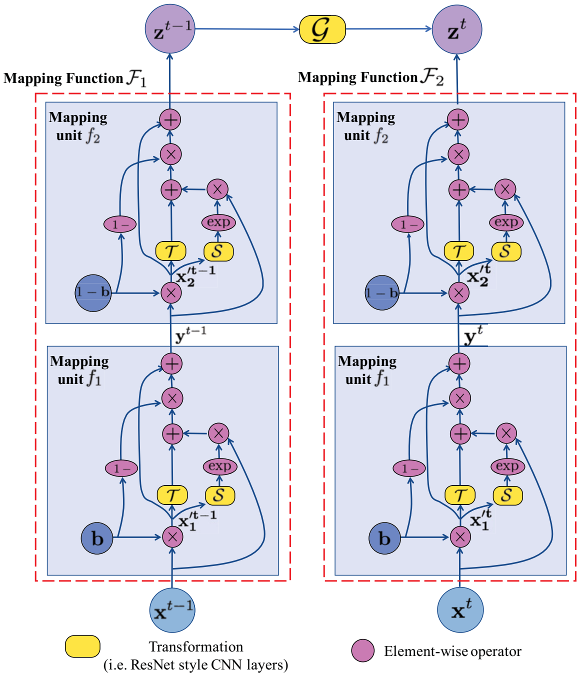


Figure 7.4: Our proposed TNVP structure with two mapping units. Both transformations $\mathcal{S}$ and $\mathcal{T}$ can be easily formulated as compositions of CNN layers.

the change of variable formula. $\frac{\partial \mathcal{F}_2(\mathbf{x}^t;\theta_2)}{\partial \mathbf{x}^t}$ is the Jacobian. Using this formulation, instead of estimating the density of a sample $\mathbf{x}^t$ conditional on $\mathbf{x}^{t-1}$ directly in the complicated high-dimensional space $\mathcal{I}$, the assigned task can be accomplished by computing the density of its corresponding latent point $\mathbf{z}^t$ given $\mathbf{z}^{t-1}$ associated with the Jacobian determinant $\left|\frac{\partial \mathcal{F}_2(\mathbf{x}^t;\theta_2)}{\partial \mathbf{x}^t}\right|$. There are some recent efforts to achieve the tractable inference process via approximations [60] or specific functional forms [26, 35, 64]. Section 7.4 introduces a non-linear bijection function that enables the exact and tractable mapping from the image space $\mathcal{I}$ to a latent space $\mathcal{Z}$ where the density of its latent variables can be computed exactly and efficiently. As a result, the density evaluation of the whole model becomes exact and tractable.

7.4 Mapping function as CNN layers

In general, a bijection function between two high-dimensional domains, i.e. image and latent spaces, usually produces a large Jacobian matrix and is expensive for its determinant computation.

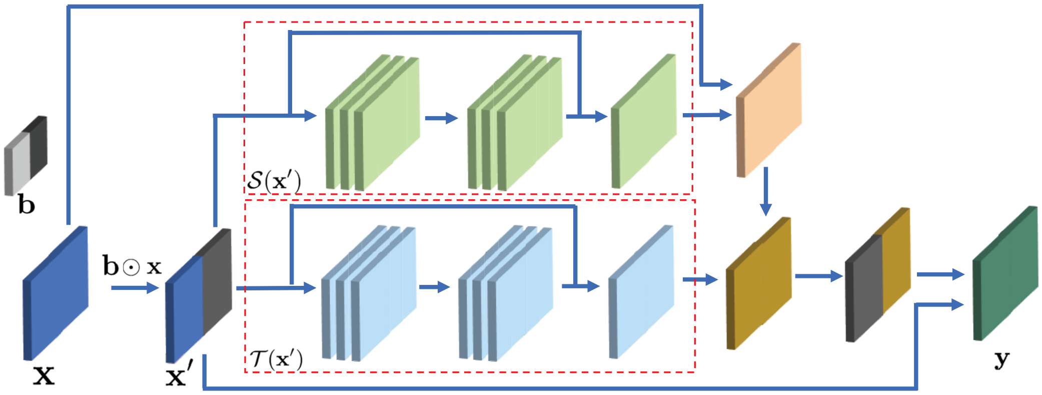

$\mathbf{b}$
$\mathbf{b} \odot \mathbf{x}$
$\mathbf{x}$
$\mathbf{x}'$
$\mathcal{S}(\mathbf{x}')$
$\mathcal{T}(\mathbf{x}')$
$\mathbf{y}$

Figure 7.5: An illustration of mapping unit f whose transformations $\mathcal{S}$ and $\mathcal{T}$ are represented with 1-residual-block CNN network.

In order to enable the tractable property for $\mathcal{F}$ with lower computational cost, this section introduces a non-linear mapping unit structure that maps variables from image space to intermediate latent spaces where the density can be computed exactly and efficiently. Then the bijection mapping function $\mathcal{F}$ is formulated as a composition of mapping units. With this structure, $\mathcal{F}$ can be efficiently set up as a deep convolutional network and enjoys the strengths of both deep networks and probabilistic models with tractable log-likelihood density estimation.

7.4.1 Mapping unit

Given an input $\mathbf{x}$, a unit $f : \mathbf{x} \to \mathbf{y}$ defines a mapping between $\mathbf{x}$ to an intermediate latent state $\mathbf{y}$ as in Eqn. (123).

$$\mathbf{y} = \mathbf{x}' + (1 - \mathbf{b}) \odot [\mathbf{x} \odot \exp(\mathcal{S}(\mathbf{x}')) + \mathcal{T}(\mathbf{x}')] \tag{123}$$

where $\mathbf{x}' = \mathbf{b} \odot \mathbf{x}$; $\odot$ denotes the Hadamard product; $\mathbf{b} = [1, \cdots, 1, 0, \cdots, 0]$ is a binary mask where the first d elements of $\mathbf{b}$ is set to one and the rest is zero; $\mathcal{S}$ and $\mathcal{T}$ represent the scale and the translation functions, respectively. The Jacobian of this transformation unit is given by

$$\begin{aligned} \frac{\partial f}{\partial \mathbf{x}} &= \begin{bmatrix} \frac{\partial \mathbf{y}_{1:d}}{\partial \mathbf{x}_{1:d}} & \frac{\partial \mathbf{y}_{1:d}}{\partial \mathbf{x}_{d+1:D}} \\ \frac{\partial \mathbf{y}_{d+1:D}}{\partial \mathbf{x}_{1:d}} & \frac{\partial \mathbf{y}_{d+1:D}}{\partial \mathbf{x}_{d+1:D}} \end{bmatrix} \\ &= \begin{bmatrix} \mathbb{I}_d & 0 \\ \frac{\partial \mathbf{y}_{d+1:D}}{\partial \mathbf{x}_{1:d}} & \operatorname{diag}\left(\exp(\mathcal{S}(\mathbf{x}_{1:d}))\right) \end{bmatrix} \end{aligned} \tag{124}$$

where diag $(\exp(\mathcal{S}(\mathbf{x}_{1:d})))$ is the diagonal matrix such that $\exp(\mathcal{S}(\mathbf{x}_{1:d}))$ is their diagonal elements. This form of $\frac{\partial f}{\partial \mathbf{x}}$ provides two nice properties for the mapping unit f. Firstly, since the Jacobian matrix $\frac{\partial f}{\partial \mathbf{x}}$ is triangular, its determinant can be efficiently computed as,

$$\left|\frac{\partial f}{\partial \mathbf{x}}\right| = \prod_j \exp(s_j) = \exp\left(\sum_j s_j\right) \tag{125}$$

where $\mathbf{s} = \mathcal{S}(\mathbf{x}_{1:d})$. This property also introduces the tractable feature for f. Secondly, the Jacobian of the two functions $\mathcal{S}$ and $\mathcal{T}$ are not required in the computation of $\left|\frac{\partial f}{\partial \mathbf{x}}\right|$. Therefore, any non-linear function can be chosen for $\mathcal{S}$ and $\mathcal{T}$. From this property, the functions $\mathcal{S}$ and $\mathcal{T}$ are set up as a composition of CNN layers in ResNet (i.e. residual networks) [42] style with skip connections. This way, high level features can be extracted during the mapping process and improve the generative capability of the proposed model. Figure 7.5 illustrates the structure of a mapping unit f. The inverse function $f^{-1} : \mathbf{y} \rightarrow \mathbf{x}$ is also derived as

$$\mathbf{x} = \mathbf{y}' + (1 - \mathbf{b}) \odot \left[(\mathbf{y} - \mathcal{T}(\mathbf{y}')) \odot \exp(-\mathcal{S}(\mathbf{y}'))\right] \tag{126}$$

where $\mathbf{y}' = \mathbf{b} \odot \mathbf{y}$.

7.4.2 Mapping function

The bijection mapping function $\mathcal{F}$ is formulated by composing a sequence of mapping units $\{f_1, f_2, \cdots, f_n\}$.

$$\mathcal{F} = f_1 \circ f_2 \circ \cdots \circ f_n \tag{127}$$

The Jacobian of $\mathcal{F}$ is no more difficult than its units and still remains tractable.

$$\frac{\partial \mathcal{F}}{\partial \mathbf{x}} = \frac{\partial f_1}{\partial \mathbf{x}} \cdot \frac{\partial f_2}{\partial f_1} \cdots \frac{\partial f_n}{\partial f_{n-1}} \tag{128}$$

Similarly, the derivations of its determinant and inverse are

$$\begin{aligned}\left|\frac{\partial \mathcal{F}}{\partial \mathbf{x}}\right| &= \left|\frac{\partial f_1}{\partial \mathbf{x}}\right| \cdot \left|\frac{\partial f_2}{\partial f_1}\right| \cdots \left|\frac{\partial f_n}{\partial f_{n-1}}\right| \\ \mathcal{F}^{-1} &= (f_1 \circ f_2 \circ \cdots \circ f_n)^{-1} = f_1^{-1} \circ f_2^{-1} \circ \cdots \circ f_n^{-1}\end{aligned} \tag{129}$$

Since each mapping unit leaves part of its input unchanged (i.e. due to the zero-part of the mask $\mathbf{b}$), we alternatively change the binary mask $\mathbf{b}$ to $1 - \mathbf{b}$ in the sequence so that every component of $\mathbf{x}$ can be jointed through the mapping process. As mentioned in the previous section, since each mapping unit is set up as a composition of CNN layers, the bijection $\mathcal{F}$ with the form of Eqn. (127) becomes a deep convolutional networks that maps its observed variable $\mathbf{x}$ in $\mathcal{I}$ to a latent variable $\mathbf{z}$ in $\mathcal{Z}$.

7.5 The aging transform embedding

In the previous section, we present the invertible mapping function $\mathcal{F}$ between a data distribution p_X and a latent distribution p_Z. In general, p_Z can be chosen as a prior probability distribution such that it is simple to compute and its latent variable z is easily sampled. In our system, a Gaussian distribution is chosen for p_Z, but notice that our proposed model can still work well with any other prior distributions. Since the connections between $\mathbf{z}^{t-1}$ and $\mathbf{z}^t$ embed the relationship between variables of different Gaussian distributions, we further assume that their joint distribution is a Gaussian. From Eqn. (121) and Figure 7.4, the latent variable $\mathbf{z}^t$ is computed from two sources: (1) the mapping from observed variable $\mathbf{x}^t$ defined by $\mathcal{F}_2(\mathbf{x}^t; \theta_2)$ and (2) the aging transformation from $\mathbf{z}^{t-1}$ defined by $\mathcal{G}(\mathbf{z}^{t-1}; \theta_3)$. The transformation $\mathcal{G}$ between $\mathbf{z}^{t-1}$ and $\mathbf{z}^t$ is formulated as,

$$\mathcal{G}(\mathbf{z}^{t-1}; \theta_3) = \mathbf{W}\mathbf{z}^{t-1} + \mathbf{b}_{\mathcal{G}} \tag{130}$$

where $\theta_3 = \{\mathbf{W}, \mathbf{b}_{\mathcal{G}}\}$ represents the connecting weights and bias of latent-to-latent interactions. Then the joint distribution $p_{Z^t,Z^{t-1}}(\mathbf{z}^t, \mathbf{z}^{t-1})$ can be computed as follows.

$$\begin{aligned} \mathbf{z}^{t-1} &\sim \mathcal{N}(0, \mathbb{I}) \\ \mathcal{F}_2(\mathbf{x}^t, \theta_2) = \bar{\mathbf{z}}^t &\sim \mathcal{N}(0, \mathbb{I}) \\ p_{Z^t,Z^{t-1}}(\mathbf{z}^t, \mathbf{z}^{t-1}; \theta) &\sim \mathcal{N}\left(\begin{bmatrix} \mathbf{b}_{\mathcal{G}} \\ 0 \end{bmatrix}, \begin{bmatrix} \mathbf{W}^T\mathbf{W} + \mathbb{I} & \mathbf{W} \\ \mathbf{W} & \mathbb{I} \end{bmatrix} \right) \end{aligned} \tag{131}$$

7.6 Model Properties

Tractability and Invertibility: With the specific structure of the bijection $\mathcal{F}$, our proposed graphical model has the capability of modeling arbitrary complex data distributions while keeping the inference process tractable. Furthermore, from Eqns. (126) and (129), the mapping function is invertible. Therefore, both inference (i.e. mapping from image to latent space) and generation (i.e. from latent to image space) are exact and efficient.

Flexibility: as presented in Section 7.4.1, our proposed model introduces the freedom of choosing the functions $\mathcal{S}$ and $\mathcal{T}$ for their structures. Therefore, different types of deep learning models can be easily exploited to further improve the generative capability of the proposed TNVP. In addition, from Eqn. (123), the binary mask $\mathbf{b}$ also provides the flexibility for our model if we consider this as a template during the mapping process. Several masks can be used in different levels of mapping units to fully exploit the structure of the data distribution of the image domain $\mathcal{I}$.

Although our TNVP shares some similar features with RBM and its family such as TRBM, the log-likelihood estimation of TNVP is tractable while that in RBM is intractable and requires some approximations during training process. Compared to other methods, our TNVP also shows its advantages in high-quality synthesized faces (by avoiding the ℓ_2 reconstruction error as in *Variational Autoencoder*) and efficient training process (i.e. avoid the step of maintaining a good balance between generator and discriminator as in case of GANs which is difficult to achieve).

7.7 Model Learning

The parameters $\theta = \{\theta_1, \theta_2, \theta_3\}$ of the model are optimized to maximize the log-likelihood:

$$\theta_1^*, \theta_2^*, \theta_3^* = \arg \max_{\theta_1, \theta_2, \theta_3} \log p_{X^t}(\mathbf{x}^t | \mathbf{x}^{t-1}; \theta_1, \theta_2, \theta_3) \tag{132}$$

From Eqn. (122), the log-likelihood can be computed as

$$\begin{aligned}\log p_{X^t}(\mathbf{x}^t | \mathbf{x}^{t-1}; \theta) &= \log p_{Z^t}(\mathbf{z}^t | \mathbf{z}^{t-1}, \theta) + \log \left| \frac{\partial \mathcal{F}_2(\mathbf{x}^t; \theta_2)}{\partial \mathbf{x}^t} \right| \\ &= \log p_{Z^t, Z^{t-1}}(\mathbf{z}^t, \mathbf{z}^{t-1}; \theta) \\ &- \log p_{Z^{t-1}}(\mathbf{z}^{t-1}; \theta_1) + \log \left| \frac{\partial \mathcal{F}_2(\mathbf{x}^t; \theta_2)}{\partial \mathbf{x}^t} \right|\end{aligned}$$

where the first two terms are the two density functions and can be computed using Eqn. (131) while the third term (i.e. the determinant) is obtained using Eqns. (129) and (125). Then the Stochastic Gradient Descent (SGD) algorithm is applied to optimize parameter values.

7.8 Discussion

This chapter has presented a novel generative probabilistic model with a tractable density function for age progression. The model inherits the strengths of both probabilistic graphical model and recent advances of ResNet. The non-linear age-related variance and the aging transformation between age groups are efficiently captured. Given the log-likelihood objective function, high-quality age-progressed faces can be produced. In addition to a simple preprocessing step, geometric constraints are implicitly embedded during the learning process. The evaluations in both quality of synthesized faces and cross-age verification showed the robustness of our TNVP. In the next chapter, the generative capabilities, robustness, and efficiency of our four models will be validated via both qualitative and quantitative experiments.

Chapter 8

Experimental Results

This chapter will first briefly introduce main features of the databases used in our evaluations of the four proposed models. They consist of two in-the-wild databases; two indoor databases with a wide range of illuminations and poses; three public aging databases. Moreover, we further collect a large-scale aging face database to train the models and analyzing aging effects. By using these databases with numerous challenging factors, we aim to show the robustness and efficiency of our proposed models. Then, in the next four sections, the generative capabilities, the robustness and efficiency of our DAM, RDAM, TRBM based age progression, and TNVP are validated in terms of both facial representation and reconstruction.

8.1 Evaluating Databases

With the aim of building a model that can represent face texture in a wide range of variations, different in-the-wild databases are chosen to evaluate the four proposed models. These databases contain unconstrained facial images collected from various multimedia resources. These facial images have considerable resolutions and contain numerous variations such as poses, occlusions and expressions.

8.1.1 In-the-wild face databases

LFPW [10] contains 1400 images in total with 1100 training and 300 testing images. However, a part of it is no longer accessible. Therefore, in the experiments, only 811 training and 224 testing images, the available remaining, are used. Each facial image is annotated with 68 landmark points provided by 300-W competition [105].

Helen [66] provides a high-resolution dataset with 2000 images used for training and 330 images for testing. The variations consist of pose changing from $-30°$ to $30°$; several types of expression such as neutral, surprise, smile, scream; and occlusions. Similar to LFPW, all faces in Helen are also annotated with 68 landmark points.

AR [76] contains 134 people (75 males and 59 females) and each subject has 26 frontal images (14 normal images with different lighting and expressions, six occluded images with sunglasses and six for scarves).

EURECOM [79] consists of facial images of 52 people (38 males and 14 females). Each person has different expressions, lighting and occlusion conditions. We only use images wearing sunglasses in our experiments.

8.1.2 Aging Databases

FG-NET[1] is a popular face aging database. There are 1002 face images of 82 subjects with age ranges from 0 to 69 years. The annotations in FG-NET are also 68 landmarks in the same format as LFPW and Helen databases.

Cross-Age Celebrity Dataset (CACD) [19] provides a large-scale dataset with 163446 images and the age ranging from 14 to 62. This dataset is collected from the Internet using keywords formed by the names of 2000 celebrities and the year (i.e. from 2004 to 2013). The annotations for this database are limited with 16 landmarks.

MORPH [102] provides a large-scale dataset with two albums of passport style images. The MORPH-I includes 1690 images from 515 subjects and the age ranges from 15 to 68. The MORPH-II contains 55134 photos of 13000 subjects. In our experiments, MORPH-I is used for evaluation.

Table 8.1: Features of collected AginG Faces in the Wild (AGFW).

Number of images	18685
Age Range	From 10 to 64 years old
Image(s)/subject	1
Sources	(1) The search engine using different keywords (i.e. male 20 years old, etc.) (2) The Productive Aging Laboratory (PAL) database. (3) Mugshot images: that are accessible from public domains.

8.1.3 Aging Database collection

In order to train the model and to analyze the aging effects, a large-scale dataset named AginG Faces in the Wild (AGFW) is further collected. Moreover, to ensure the consistency of the collected data, the tag names and the age-related information of these images are also considered. The resulting dataset consists of 18,685 images with the age ranging from 10 to 64 years. It is then decomposed into 11 age groups with the age span of 5 years. On average, each age group consists of 1700 images of different people in the same age group. The Productive Aging Laboratory (PAL) Face database [80] is also included in our collected dataset.

8.2 Face modeling with Deep Apprearance Models

In this section, the generative capabilities of our DAM is validated in terms of both facial reconstruction and representation via four applications, i.e. facial super-resolution reconstruction; facial off-angle reconstruction; facial occlusion removal and facial age estimation. The experiments are also made to be more challenging by including numerous variations in poses, occlusions and impulsive noise. Comparing to other methods such as PCA-based AAM and bicubic interpolation, our method achieves better reconstructions without blurring effects or spreading out the errors caused by occlusions or noise. Then, in section 8.2.4, a shape fitting experiment to evaluate our proposed Deep Appearance Models in its ability of synthesizing new face images is also presented. Its performance in term of point-to-point error is compared with AAM and other face alignment methods such as RCPR [16].

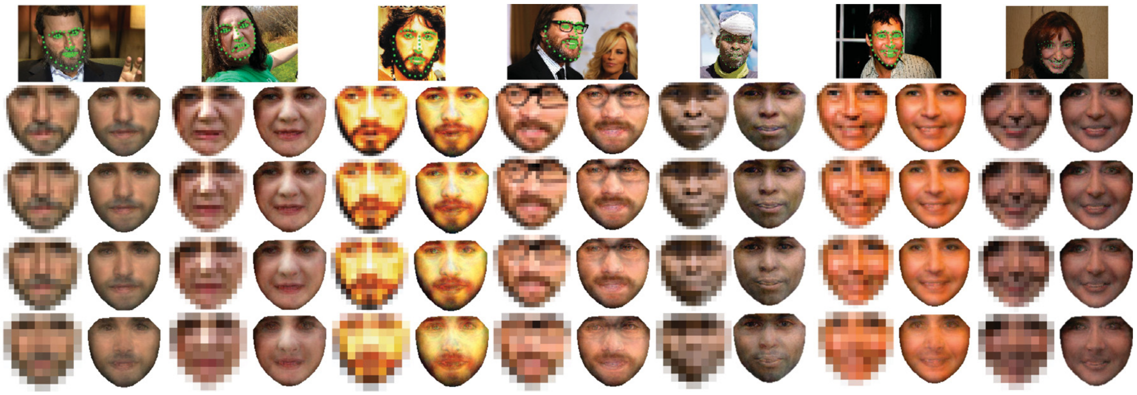


Figure 8.1: Facial image super-resolution reconstruction at different scales of down-sampling. The 1st row: original image, the 2nd row to the 5th row: down-scaled images with factors of 4, 6, 8, 12 (left) and reconstructed facial images using DAM (right).

8.2.1 Facial Super-resolution Reconstruction

The proposed DAM method is evaluated in its capability to recover high-resolution face images given their very low-resolution versions. Moreover, since LFPW and Helen databases also include numerous variations in poses, expressions and occlusions, the experiment becomes more challenging. The proposed method is very potential in dealing with the problem of super-resolution in various conditions of facial poses and occlusions.

In order to train the DAM model, 811 training images from LFPW and 2000 images from Helen database are combined into one training set. The coordinates of facial landmarks are normalized to zero mean before setting as observations to train the shape model. In the texture modeling, shape-free images are first extracted by warping faces into the texture domain $\mathcal{D}$. The size of the shape-free image is set to 117×120 pixels based on the mean shape of the training data. Then texture model is trained to learn the facial variations represented in these shape-free images.

During the testing phase, since the number of visible units in the texture model are fixed, the testing low-scale facial shape-free image is first resized to 117×120 using bicubic interpolation method. Then both the shape and the shape-free image are clamped to DAM. After 50 epoches in the alternating Gibbs updates, the face texture is reconstructed based on the current states of hidden unit $\mathbf{h}_g^{(1)}$. Different magnification factors α are used for evaluating the quality of DAM reconstructions. Testing images are down-sampled in different magnification levels ranging from 4

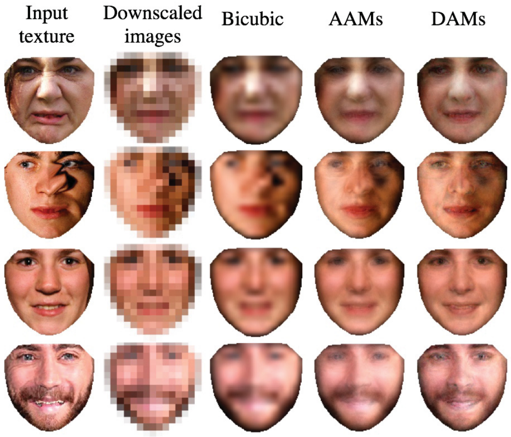


Figure 8.2: Results of average RMSEs over 4 images: Bicubic interpolation (RMSE = 19.68); PCA-based AAM reconstruction (RMSE = 19.96); (d) Deep Appearance Models reconstruction (RMSE = 20.44).

Table 8.2: The average RMSEs of reconstructed images using different methods against LFPW and Helen databases with $\alpha = 16$

Methods	LFPW	Helen
Bicubic	19.53	22.13
AAM [128]	19.74	22.3
DAM (Ours)	**19.24**	**21.24**

to 12. They are then used as inputs to the reconstruct module using our approach. Figure 8.1 shows the reconstruction results using the DAM approach. Remarkable results are achieved using DAM with very low-resolution input images, i.e. 10×10 pixels with the magnification factor $\alpha = 12$.

Comparisons against Baseline Methods: The proposed approach is also compared with two base-line methods, i.e. bicubic interpolation method and PCA-based AAM [128]. Root Mean Square Error (RMSE) is used as a performance measurement. RMSE is a common metric that is usually used for evaluating image recovery task. Although this metric is not always reliable for rating image quality visually [133], it could provide a qualitative view for comparing DAM and other methods.

From the results shown in Figure 8.2, the method gives better reconstruction results in visualization than the others. However, the RMSE results are not much better as shown in Table 8.2. This is because RMSE cannot fully evaluate the quality of reconstructed images in the task of image

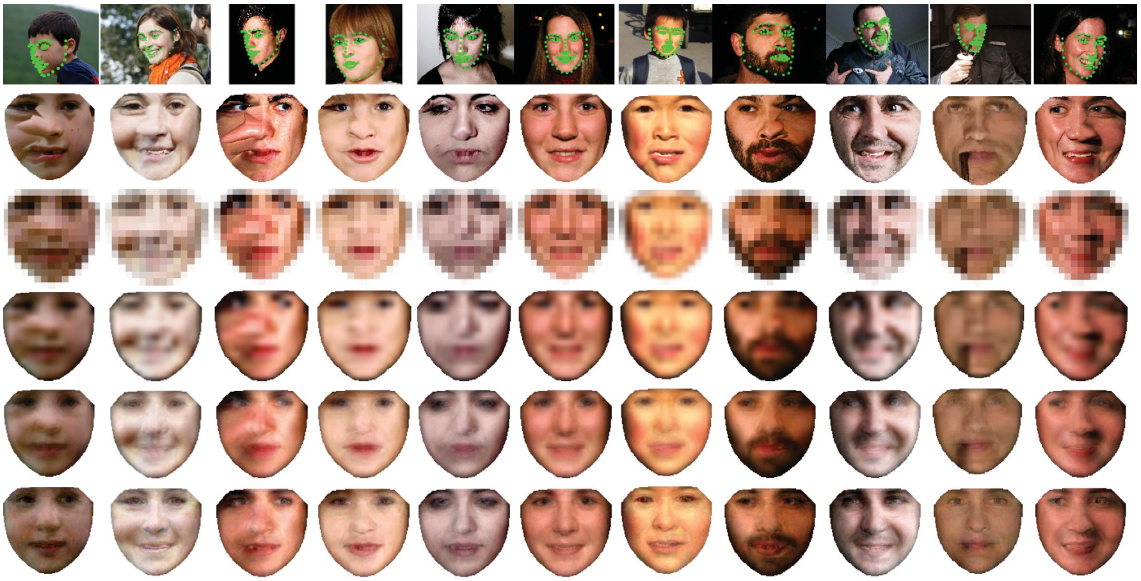


Figure 8.3: Facial image super-resolution. The original images (first row) are warped to shape-free images in texture domain (second row); then they are down-sampled by a factor of 8 from 117×120 to 15×15 (third row) The next three rows are the high-resolution reconstructed using Bicubic method (the fourth row), PCA-based AAM (the fifth row) and Deep Appearance Models (the sixth row).

super-resolution [139]. Especially, we don't have the ground-truth for RMSE evaluation in these databases. For example, in the cases of occlusions and poses in those databases, although the reconstructed images obtained using PCA-based AAM and bicubic methods are very blurry, their RMSEs are still low. This is because the reconstructed images still contains occlusion components or pose features which are quite similar to the original ones. Figure 8.3 illustrates further reconstruction results obtained using bicubic method, PCA-based AAM method and DAM approach. The PCA-based AAM method is trained using the same dataset as DAM and the length of texture parameter vector is 200, the highest level used in [128]).

Comparisons against Other Super-resolution Methods: For further evaluations, DAM are compared with other super-resolution methods. Two types of approaches are chosen for comparisons, i.e. image super-resolution and face hallucination. The main difference between these two approaches is that the former is designed for images in general while the latter is more specific for facial images. Figure 8.4 compares the reconstructed faces using DAM against sparse representation based image super-resolution (ScSR) [139] and Structured Face Hallucination (SFH) [137]

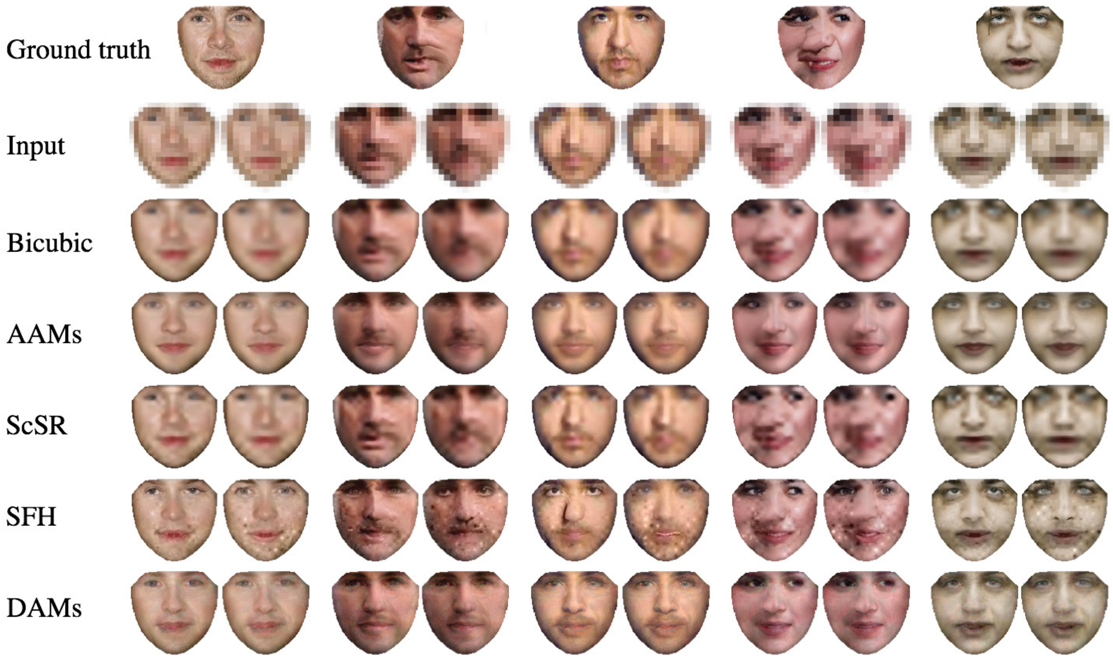


Figure 8.4: Comparisons of different facial image super-resolution methods. The 1st row: ground truth faces. The 2nd row: down-scaled images with factors of 6 (left) and 8 (right). From the 3rd row to the 7th row: reconstructed faces using bicubic, PCA-based AAM, ScSR [139], SFH [137] and DAM, respectively.

methods. For each face, the low-resolution (LR) faces (i.e. LR_6 and LR_8) are obtained by down-sampling the ground truth face with factors of 6 and 8. Their high-resolution (HR) reconstructed faces (i.e. HR_6 and HR_8) of different methods are shown in the left and right columns, respectively. The results of bicubic and PCA-based AAM are also presented in this figure. The resolution of LR_6 is 20×20 and that of LR_6 is 15×15.

It is clear in the figure that SFH performs better than ScSR in term of reconstruction details. This is because SFH was already trained with the face structure and contour's statistical priors. However, some noisy and blocky effects are still remained in the reconstructed faces of SFH. Especially, when parts of face images are blurred due to the effects of warping operator, artifacts may appear in its final results. Meanwhile, remarkable results can be achieved by DAM in terms of keeping fine details without noisy effects. In addition, these results aslo show the advantages of DAM when dealing with higher magnigication factor α. Whereas all four methods fail to produce high quality reconstructions when α increases from 6 to 8, DAM still perform well and generate faces with

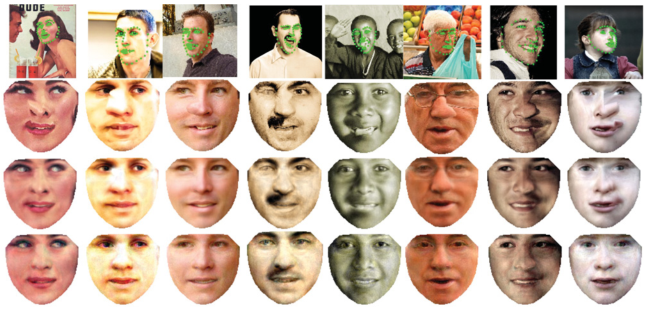


Figure 8.5: Facial off-angle reconstruction: the 1st row: original image, the 2nd row: shape-free image, the 3rd row: PCA-based AAM reconstruction [128], and the 4th-row: DAM reconstruction

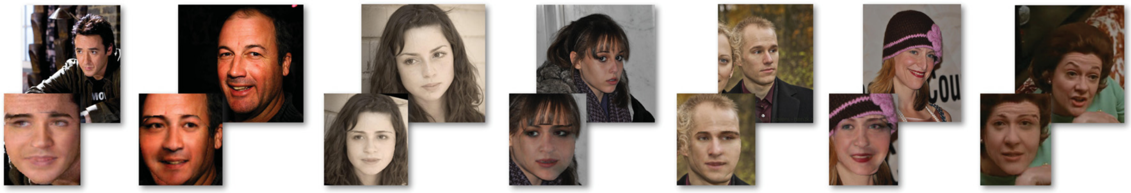


Figure 8.6: Face Frontalization: Top: input faces and Bottom: frontalized faces reconstructed using DAM.

consistent quality.

8.2.2 Facial off-angle Reconstruction and Occlusion Removal

This section illustrates the ability of DAM to deal with facial poses and occlusions.

Facial off-angle Reconstruction Using the same trained model as in the previous experiment, facial images with different poses are represented in Figure 8.5.

Comparing to AAM, our DAM achieve better reconstructions especially in the invisible regions of extreme poses. These regions in shape-free images are blurry and noisy due to the non-linear warping operator. Therefore, the errors are spread out in the reconstructions of PCA-based AAM approaches. Meanwhile, the generative capability of our proposed DAM method can solve those challenging cases. From the results, it is easy to see that the blurry effects are effectively removed in DAM reconstructions.

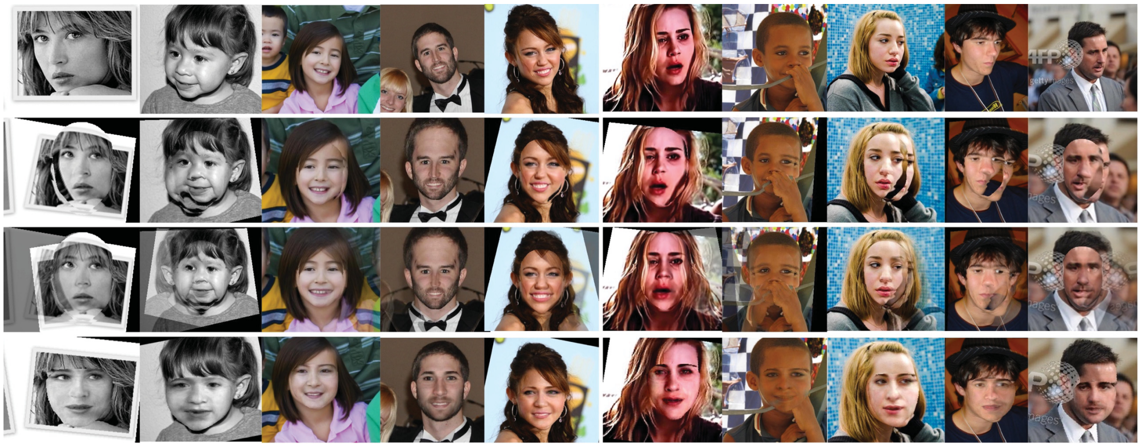


Figure 8.7: Comparisons between DAM and Face Frontalization approach [40]. The 1st row: input faces; the 2nd and 3rd rows: synthesized frontal view before and after applying soft symmetry [40]; the 4th row: frontalized faces produced by DAM.

Face Frontalization: Next, this ability of DAM approach is further emphasized on the face frontalization problem. Given an input face with pose, the process of "frontalization" is to synthesize the frontal view of that face. Notice that the facial photos are unconstrained and the subjects are not required to already be in the training data. Once again, in order to produce aesthetic frontal view, not only poses but other factors such as expressions and occlusions are needed to be taken into account. The frontalization can help to boost the performance of other subsequent processes such as face recognition, verification, gender estimation [40], etc. Figure 8.6 represents the frontalized views of input faces with different poses and expressions given in the top row.

The reconstruction results are also compared with the recent frontalization work [40] against LFPW and Helen databases in the Figure 8.7. From the second and third rows, one can see that the approach in [40] achieves good reconstructions when the input poses are not so extreme (i.e. not greater than 30 degrees). However, in case of extreme poses (i.e. the first two and the last three faces) or occlusions (the 7th face), even when the symmetry property is used, the full faces can not be reconstructed aesthetically. On the other hand, the results in the last row show that DAM can effectively synthesize the frontal views of these faces without further applying the soft symmetry property. Since the face priors are already learned, DAM are able to produce more natural faces instead of duplicating the information from known side to the other side.

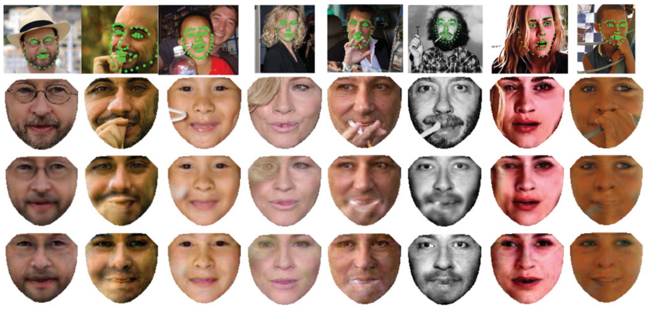


Figure 8.8: Occlusion removal: the 1st row: original image, the 2nd row: shape-free image, the 3rd row: PCA-based AAM reconstruction still remains with occlusion and blurring effects, and the 4th-row: DAM reconstruction can help to remove the occlusion

Facial Occlusion Removal: Similarly, DAM also show their capability in the problem of facial occlusion removal. In Figure 8.8, the occlusions, e.g. hands, glasses, hair, etc., can be removed successfully without blurring effects. More interestingly, the occlusions are removed from faces without loosing facial features. For example, glasses are totally removed without making beard blurred as in the PCA-based AAM reconstruction.

Using occluded faces as references and measuring the reconstruction quality by RMSE cannot illustrate the modeling capabilities of DAM. To get a better evaluation protocol, we select a subset of 174 occluded faces of the first 29 subjects, i.e. 15 males and 14 females, from AR database [76]. We employ DAM to reconstruct these occluded faces and then use their corresponding neutral faces, i.e. frontal face without occlusions, as references to compute the RMSE. In this testing set, each subject includes two faces with scarf and four other faces with both illumination and scarf. The average RMSE of DAM is 45.08 while that of PCA-based AAM is 47.36. The trained models in DAM and AAM use LFPW and Helen databases. This experiment shows that DAM achieve better reconstructions, i.e. closer to the neutral faces, compared to AAM.

8.2.3 Facial Age Estimation

Besides some other previous age estimation approaches [31, 72], the proposed DAM are employed to this problem to further demonstrate their robustness and effectiveness.

Evaluation on reconstructed images: Since the texture is an important factor to predict a person's age given his facial image, this experiment will evaluate how good the reconstructed image

Table 8.3: The MAEs (years) of different methods against impulsive noise

Methods	No noise	Noise range			
		25	50	100	150
AAM [128]	6.14	6.15	6.11	**6.13**	6.47
DAM	**5.67**	**5.81**	**5.56**	6.14	**6.18**

Table 8.4: The MAEs (years) of different methods against low-resolution testing faces

Methods	Magnification factor α			
	2	4	6	8
Bicubic	5.96	6.95	7.15	7.21
AAM [128]	6.13	6.33	6.44	6.69
DAM	**5.91**	**6.00**	**6.11**	**6.21**

is as well as how much aging information is retained by the model.

To make this task more challenging, we add noise to the testing facial image and then predict the age of that person using “clean” reconstructed face from DAM. For the evaluation system, we re-implemented the age estimation system presented in [72] and trained it with 802 images from FG-NET. The remaining 200 images were used for testing. To generate noisy testing images, all pixels of facial images were mixed with uniform noise ranged within $[-r, r]$.

A similar experiment is set up as follows: given the low-resolution testing face, the system will predict the age of that person using his high-resolution reconstructed face. The Mean Absolute Errors (MAEs) of different methods against noise and low-resolution testing faces are represented in Table 8.3 and Table 8.4, respectively. The performance in terms of Cumulative Scores (CS) is illustrated in Figure 8.9. From these results, in both cases, the smallest error is achieved with DAM model. Therefore, the proposed model produces better reconstructed results under the effects of noise and low-resolution factor.

Evaluation on model features: Beside the ability of generalizing the faces, DAM can produce a higher level representation for both facial shape and texture. Therefore, instead of using pixel values, we extracted the model parameters as described in Section 4.5 and evaluated them with the age estimation system. For the AAM features, the number of features for shape and texture was chosen so that 93% of variations are retained. Table 8.5 lists the MAEs of four different inputs: reconstructed image of DAM (**DAM-Rec**) and AAM (**AAM-Rec**), model parameters extracted from

Table 8.5: Comparison of age estimation results on FG-NET database with four different features

Inputs	MAEs (years)
DAM-Mod	**5.28**
AAM-Mod	5.35
DAM-Rec	5.67
AAM-Rec	6.14

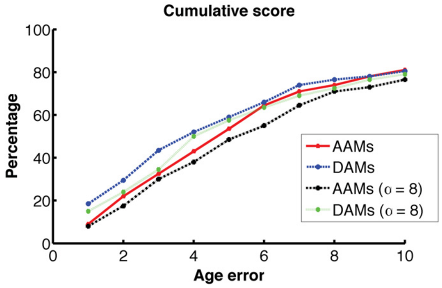


Figure 8.9: Cumulative scores of using reconstructed images from original-scaled and down-sampled images with a factor of 8

AAM (**AAM-Mod**) and DAM (**DAM-Mod**). Not surprisingly, our DAM feature achieves the lowest MAEs as compared with AAM features.

8.2.4 Shape Fitting in DAM

This section presents the experiments with the shape fitting on LFPW database. The model configurations are kept the same as in previous sections except it is now trained with 811 training images of LFPW. For evaluation and comparision, we use the average distance of each landmark to its ground truth position normalized by face size as in [128]. Moreover, in order to remove the effect of face detection error during fitting step, the bounding boxes provided in [105] are used for initialization. Then the mean shape with 68 landmarks is simply placed inside the face's bounding box to start the fitting process. The proposed method is compated with two other fitting strategies, i.e. AAM and RCPR [16], and the results are presented in Table 8.6. The Cumulative Error Distribution (CED) curves are also showed in Figure 8.10. From these results, we can see that DAM achieve performance comparable to other face alignment methods.

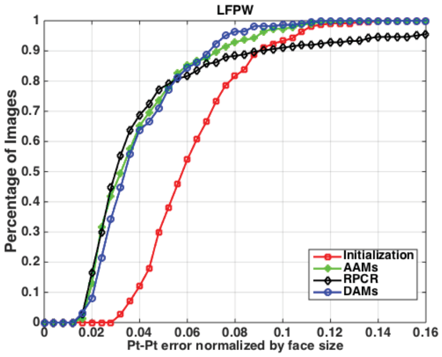


Figure 8.10: Cumulative Error Distribution (CED) curves of LFPW database.

Table 8.6: The fitting errors using different methods against LFPW database

Methods	Fitting Error
Initialization	0.0618
Fast-SIC [128]	0.0391
RCPR [16]	0.0505
DAM (Ours)	0.0398

Table 8.7: Computational time of DAM and AAM in three stages

Stages	DAM	AAM [128]
Training	12.87 hrs	564.06 s
Fitting (per image)	19.17 s	2.28 s
Reconstruction (per image)	0.53 s	0.023 s

8.2.5 Computational Costs

The computational costs of DAM, i.e. training, fitting and reconstruction stages are discussed in this section. Both LFPW and Helen databases are combined to use in this evaluation. The numbers of training and testing images are 2811 and 554, respectively. The method is implemented in Matlab environment and runs in a system of Core i7-2600 @3.4GHz CPU, 8.00 GB RAM. The shape contains 68 landmarks and the appearance is represented in a vector of 9652 dimensions. Each layer was trained using Contrastive Divergence learning in 600 epochs. It is noted that the current version is implemented without using parallel processing. The computational costs of both DAM and AAM are shown in Table 8.7.

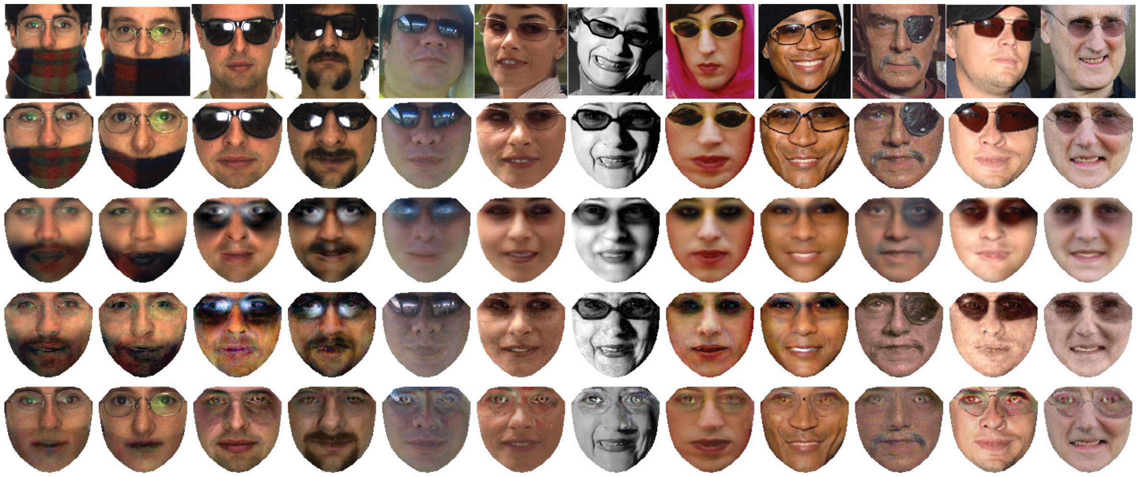


Figure 8.11: Reconstruction results on images with occlusions (i.e. sunglasses or scarves) in LFPW, Helen and AR databases. The first row: input images, the second row: shape-free images, from the third to fifth rows: reconstructed results using AAM, DAM, and RDAM, respectively

8.3 Face Modeling with Robust Deep Appearance Models

In this section, we validate the ability of RDAM to handle extreme cases of occlusions and poses as well as model fitting in RDAM.

8.3.1 Facial Occlusion Removal

This section demonstrates the ability of RDAM to handle extreme cases of occlusions such as sunglasses or scarves. RDAM are trained in two steps: pre-train each layer and train the whole model. The training set includes 1000 “clean” (i.e. faces without occlusion and pose) and 200 posed images from LFPW and Helen; 534 “clean”, 95 sunglasses, and 95 scarf images from 95 subjects in AR; 104 images from 52 subjects in EURECOM. During the pre-training step, we separately train the shape model; “clean” texture model with clean images; and a binary mask RBM with masks generated from occluded and posed images. After that, we can train the texture model with pre-initialized weights of the “clean” texture model and mask RBM. The joint layer is later trained with all training images. Each step above is trained using Contrastive Divergence learning in 600 epochs on a system of Xeon@3.6GHz CPU, 32.00GB RAM. The computational costs (without parallel processing) are as follows. The training time is 14.2 hours. Fitting on average is 17.4s.

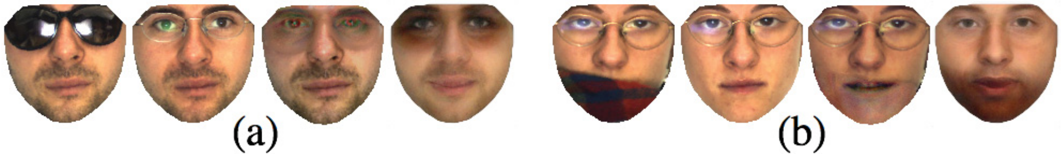


Figure 8.12: Reconstruction results on images with sunglasses (a) or scarves (b) in AR database. The images are input shape-free, ground truth shape-free, reconstructed results using RDAMs and RPCA [96], respectively.

Table 8.8: The average RMSEs of reconstructed images using different methods on LFPW and AR databases with sunglasses (SG) and scarf (SF)

Methods	AAM [128]	DAM [88]	**RDAM**
LFPW	12.91 (18.98)	11.15 (14.98)	**8.58** (23.98)
AR - SG	56.55	55.48	**41.67**
AR - SF	63.16	60.96	**47.65**

Reconstructing faces on average is 1.53s.

As shown in Figure 8.11, RDAM can remove those occlusions successfully without leaving any severe artifact comparing to AAM and DAM. The comparison between RPCA approach [96] and the proposed RDAM in both cases of sunglasses and poses is also shown Figure 8.12. We evaluate the reconstruction quality in terms of Root Mean Square Error (RMSE) on LFPW, Helen, AR and EURECOM databases. For AR database, we choose two subsets of 210 images with sunglasses and 210 images with scarves from 38 subjects not in the training set, i.e. 30 males and eight females. The corresponding normal face images, i.e. frontal and without occlusions, of the same person are used as the references to compute the RMSE. For LFPW and Helen databases, we select a subset of 23 images with sunglasses and 100 images with some occlusions around the mouth. A mask is used to ignore occluded/corrupted pixels in the testing images so that we have an unbiased metric. The average masked-RMSEs of AAM, DAM and RDAM are shown in Table 8.8. The average unmasked-RMSEs are also reported for reference (i.e. the numbers inside the brackets). From these results, one can see that the RDAM outperforms AAM and DAM in terms of both reconstruction quality and RMSE metric. Note that the unmasked-RMSE is always higher than masked-RMSE since some corrupted pixels are recovered during reconstruction. Since our RDAM can recover more corrupted/occluded pixels, it makes the un-masked RMSE higher than the ones from AAM and DAM.

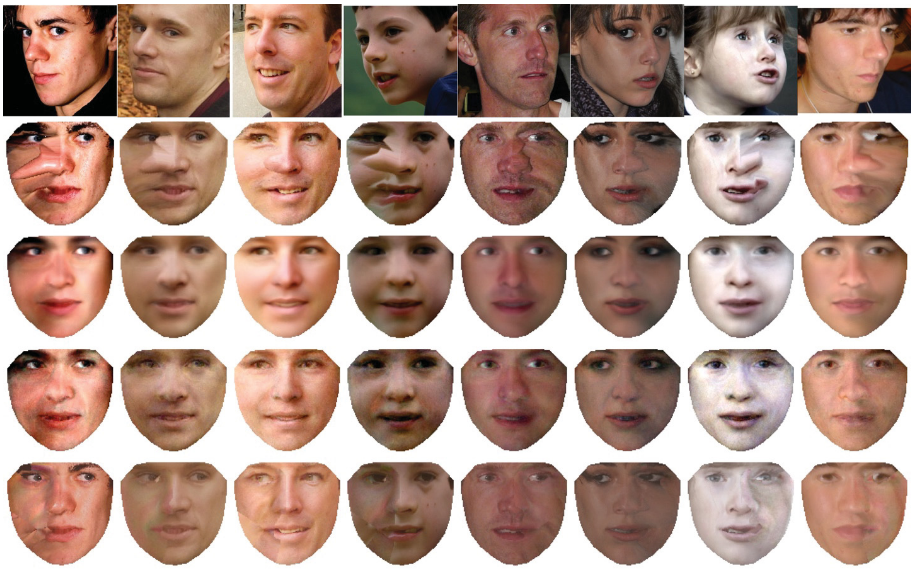


Figure 8.13: Facial pose recovery results on images from LFPW and Helen databases. The first row is the input images. The second row is the shape- free images. From the third to fifth rows are AAM, DAM and RDAM reconstruction, respectively.

8.3.2 Facial Pose Recovery

This section illustrates the capability of RDAM to deal with facial poses. Using the same pre-trained model presented in Section 8.3.1, the texture model was trained using 280 images with different pose variations from LFPW and Helen databases. The reconstruction results of facial images with different poses are presented in Figure 8.13. In this experiment, our RDAM also achieves the best reconstruction results comparing to AAM and DAM especially in the cases of extreme poses (more than $45°$). Our proposed RDAM method can handle those extreme poses in a more natural way. From Figure 8.13, RDAM give reconstructed faces that look more similar to the original faces while DAM or AAM make the face look younger or change its identity.

8.3.3 Model Fitting in RDAM

The aim of this experiment is only to evaluate the performance of different model fitting algorithms that are described in section 5.5. Those algorithms are employed to find optimal parameters for the models that give the best reconstructed results. We evaluated our model fitting algorithms incorporating a corrupted pixel mask with the baseline fitting methods without using the mask on the

Table 8.9: The average MSE between estimated shape and ground truth shape (68 landmark points) on sunglasses (SG) and scarves (SF) images. Tested on about 300 images (23 images from LFPW database and 268 images from AR database)

Type	Method	Initial	With Mask	Without Maks
SG	FA	0.0406	**0.0353**	0.0361
	IC	0.0406	**0.038**	0.039
	FC	0.0406	**0.0372**	0.0373
SF	FA	0.0874	0.0873	**0.0849**
	IC	0.0874	**0.0853**	0.0864
	FC	0.0874	0.0873	**0.0849**

Table 8.10: The fitting time and the average MSE of estimated shapes (68 points) on sunglasses (SG) and scarves (SF) images.

Methods	Initial	DAM	RDAM	Fast-SIC	AOMs
MSE - SG	0.195	0.1732	0.1664	0.1218	0.1705
MSE - SF	0.211	0.0947	0.0756	0.0756	0.1705
Fitting time		19.17s	17.4s	2.28s	1.26s

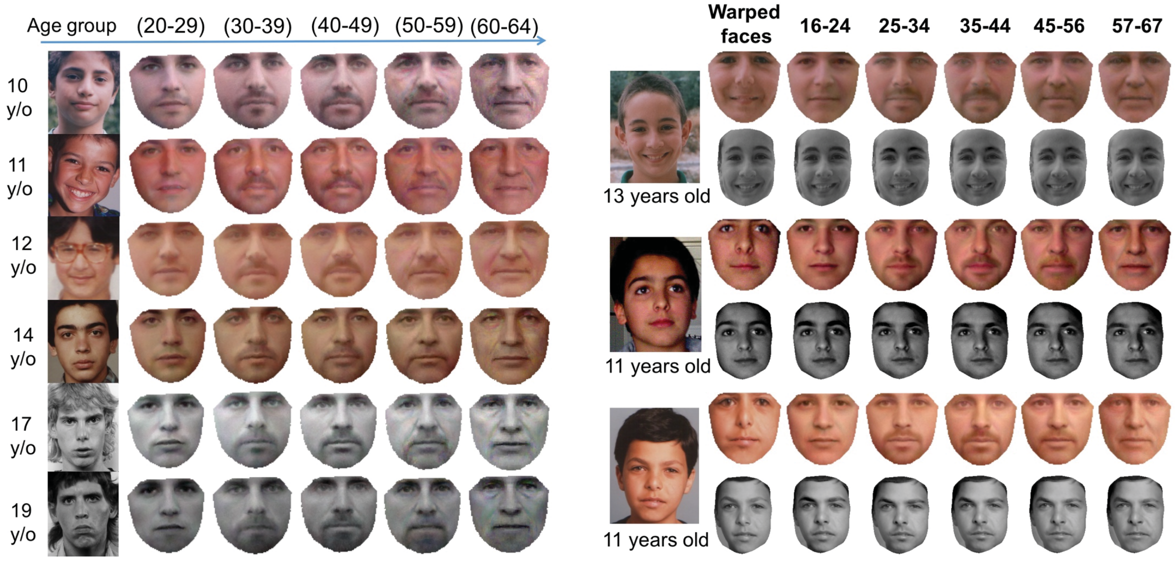
LFPW and the AR databases. Three model fitting algorithms (i.e. Forward Additive (FA), Inverse Compositional (IC) and Forward Compositional (FC)) are compared on two types of occlusions including sunglasses (SG) and scarf (SF). The average errors are reported in Table 8.9. We further compare our results with Active Orientation Models (AOMs) [129] and Fast-SIC [128] as in Table 8.10. The initial shape is the mean shape placed inside the faces bounding box. RDAM achieves comparable performance compared to other methods.

8.4 Face Age Progression with Temporal Restricted Boltzmann Machines

In this section, we evaluate the efficiency and flexibility of our TRBM based system in both age progression and regression applications. We next demonstrate the generality and robustness of our model with in the wild data.

8.4.1 Age Progression

In order to train the RBMs for reference sequence generation, the AGFW dataset is decomposed into 11 age groups with the age span of 5 (i.e. age 10-14, 15-19, ..., 60-64). On average, each age



(a) (b)

Figure 8.14: (a) **Age progression results**. Given an input image in age range 10-19, the system automatically reconstructs age-progressed images in various age ranges. (b) **Comparisons between our approach and IAAP** [57]. For each case, the input face image (1st column) is aligned and normalized to frontal face (2nd column). From the 3rd to the 7th column: the progressed images corresponding to several age groups using our approach (the row above) and IAAP (the row below).

group consists of 1700 images. These images are then used for constructing the set of RBMs as represented in Section 6.3. For training the TRBM based age progression component, we select a subset of 572 celebrities from the CACD dataset and also classify their images into 11 age groups with the age span of 5. Then for each person, one image per age group is randomly selected. This process results in a training data with 572 sequences. Since the images are collected from 2004 to 2013, the longest sequence consists of only three images.

All training images are then aligned and normalized as presented in section 6.2. The size of the normalized image is set to 95 × 95 pixels based on the reference shape generated in the alignment step. The TRBM based age progression model is then employed to learn the aging transformation between faces. After all components are trained, we run our system on every face over 10 years old of FG-NET and MORPH databases. Figure 8.14a illustrates the age-progressed faces reconstructed by our model. Notice that both FG-NET and MORPH databases are not part of our training data.

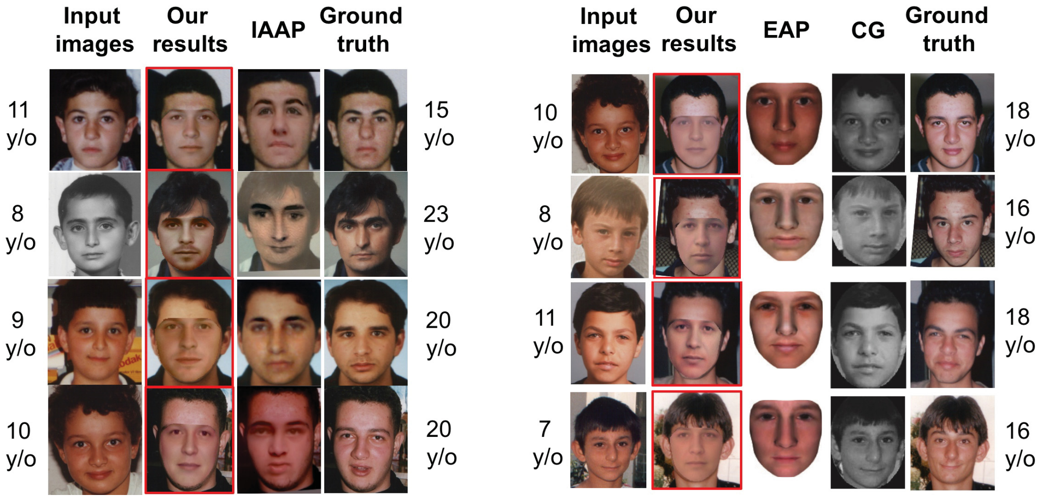
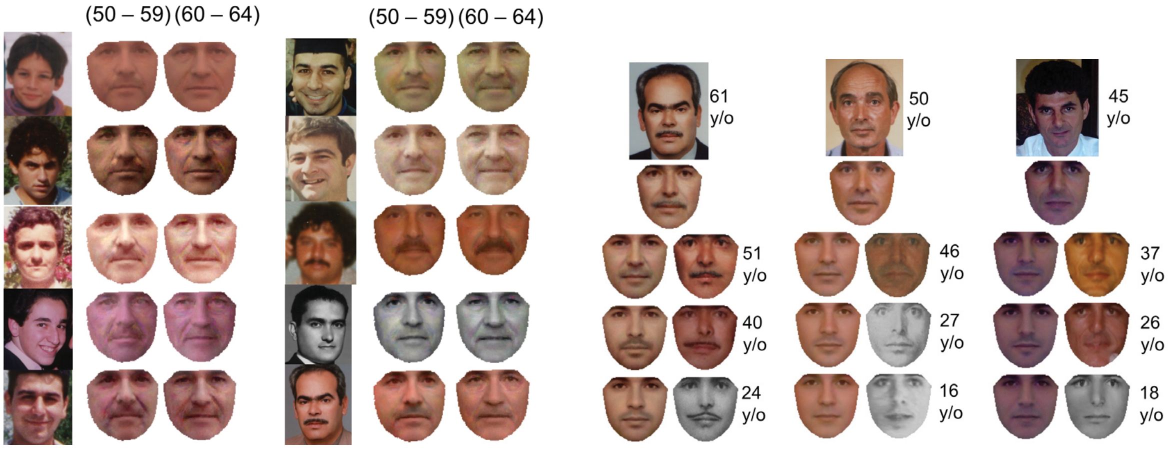


Figure 8.15: Comparisons between our appproach and other age progression approaches: IAAP [57], EAP [110] and CG [100].

Our age-progressed sequences are compared with the recent age progression work, Illumination-Aware Age Progression (IAAP) [57] against FG-NET database in Figure 8.14b. From these sequences, one can see that IAAP approach synthesizes very similar faces among different age groups. Moreover, since the texture difference between average faces is used as the main source for aging process, the synthesized faces usually look younger than those from their own age groups. Meanwhile, more nonlinear aging features in each age group are still kept in the reconstructed results of our approach. In addition, one can easily see that our age-progressed sequences are able to better reflect the face changes during the aging process (i.e. the appearance of beard in the middle stages and wrinkle in the later stages). For further evaluations, we compare our proposed model with other approaches including IAAP; Exemplar based Age Progression (EAP) [110] and Craniofacial Growth (CG) model [100] in Figure 8.15. The ground truth images are also provided for comparisons. It should be noted that since our model is trained using the collected data with ages ranging from 10 to 64, in cases where the IAAP uses input images at ages less than 5, we choose images of the same individuals with age close to 10 as input for our system.



(a) Age progression "in the wild"

(b) Age regression results

Figure 8.16: (a) **Age progression "in the wild"** with other variations in the input images such as poses, illuminations, expressions. (b) **Age regression results**. For each case, the input image (1st row) is normalized to frontal face (2nd row). From the 3rd row to 5th row: the age-regressed images generated by our model (left) and the ground truth images with the corresponding ages (right).

8.4.2 Age Progression "in the Wild"

In order to validate the robustness of our model, in this experiment, we focus on input images that include different variations such as poses, expressions, illuminations. Blurry images are also considered. Figure 8.16a illustrates age-progressed images that are automatically reconstructed by our model. From these results, one can see that although other non-linear variations also present in the input images, remarkable results can still be achieved by our model in terms of fine aging details without any quality reduction.

8.4.3 Age Regression

We next emphasize the flexibility of our proposed model by evaluating its capability to generate the younger faces of an individual given his/her current appearance. The results of this application can be easily obtained using our model by simply keeping the same training process as in previous experiments except the training sequences are reversed. The faces at younger ages are represented in Figure 8.16b.

Table 8.11: The MAEs (years) of Age Estimation System on Ground Truth and Age-progressed Results

Inputs	Dataset	MAEs
Ground Truth faces (set A)	FGNET	5.89
Synthesized faces (set B)	FGNET	5.96
IAAP 's synthesized faces (set B')	FGNET	6.29
Ground Truth faces (set C)	MORPH	4.84
Synthesized faces (set D)	MORPH	5.17

8.4.4 Automatic Age Estimation

One challenge of the face data "in the wild" comes from the age labels of the input images. In most cases, this information is incorrect or unavailable. Thus, it causes lots of difficulties for age progression process in later stage. Far apart from previous age progression systems, the effectiveness and scalability of our proposed model are further increased by integrating an age estimation system to the proposed framework. In this way, given a face image, our system can do age progression without any further information.

Besides some other previous age estimation approaches [54, 74, 75, 88], in this work, we re-implement the method in [72] which is among the state-of-the-art age estimators reported in [85]. Moreover, this approach is modified with three-group classification in the first step (youths, adults, and elders) before constructing three Support Vector Regression (SVR) based aging functions. In order to train this age estimator, we randomly select 802 images from FG-NET and 1000 images from MORPH as the training data. The remaining images of these two databases are used for testing. The Mean Absolute Errors (MAEs) achieved are 5.86 years for FG-NET and 4.84 years for MORPH. By incorporating this age estimator to our age-progression framework, the need for age label is alleviated and, therefore, making the whole framework fully automatic.

8.4.5 Age Accuracy of Age-progressed Results

This section illustrates the accuracy of our synthesized results in term of age perceived. In other words, this experiment aims at assessing whether the age-progressed faces are perceived to be at the target ages. In this evaluation, the trained age estimation system in the previous experiment is adopted to compare the accuracies on the ground-truth and age-progressed faces. From the testing set of FG-NET database, we select all images above 10 years old and consider them as the ground truth images. This forms the set A consisting of 135 images. Each photo of an individual in set A is then progressed to the later ages where the ground truth faces are available. This process results in the set B of 194 age-progressed images. In order to compare with IAAP method, we apply this process using IAAP and obtain the set B'. For a large scale evaluation, we further generate a test set using MORPH database. Let the test set of MORPH as in section 8.4.4 be set C. For each individual in the testing data, we synthesize four aged images accross three decades. This gives us 1421 images that compose set D. The MAEs of the age estimation system on these test sets are listed in Table 8.11. These results show that the age estimation accuracies of our age-progressed images are comparable to those of ground truth images. Therefore, our proposed model is able to generate the age-progressed faces at the target ages.

8.5 Temporal Non-Volume Preserving Approach for Facial Age-Progression and Age Invariant Face Recognition

In this section, we firstly represent the implementation details of TNVP model. Then we evaluate our TNVP via both qualitative and quantitative experiments. The large-scale face verification benchmark on Megaface challenge 1 is also employed to further demonstrate the efficiency and robustness of our TNVP model.

8.5.1 Implementation details

In order to train our TNVP age progression model, we first select a subset of 572 celebrities from CACD as in the training protocol of [89]. All images of these subjects are then classified into 11 age groups ranging from 10 to 65 with the age span of 5 years (i. e. 10-14, 15-19, . . . , 55-59, 60-65). Next, the aging sequences for each subject are constructed by collecting and combining all image pairs that cover two successive age groups of that subject. This process results in 6437 training sequences. All training images from these sequences and the AGFW dataset are then preprocessed as presented in Section 7.2. After that, a two-step training process is applied to train our TNVP age progression model. In the first step, using faces from AGFW, all mapping functions (i.e. $\mathcal{F}_1, \mathcal{F}_2$)

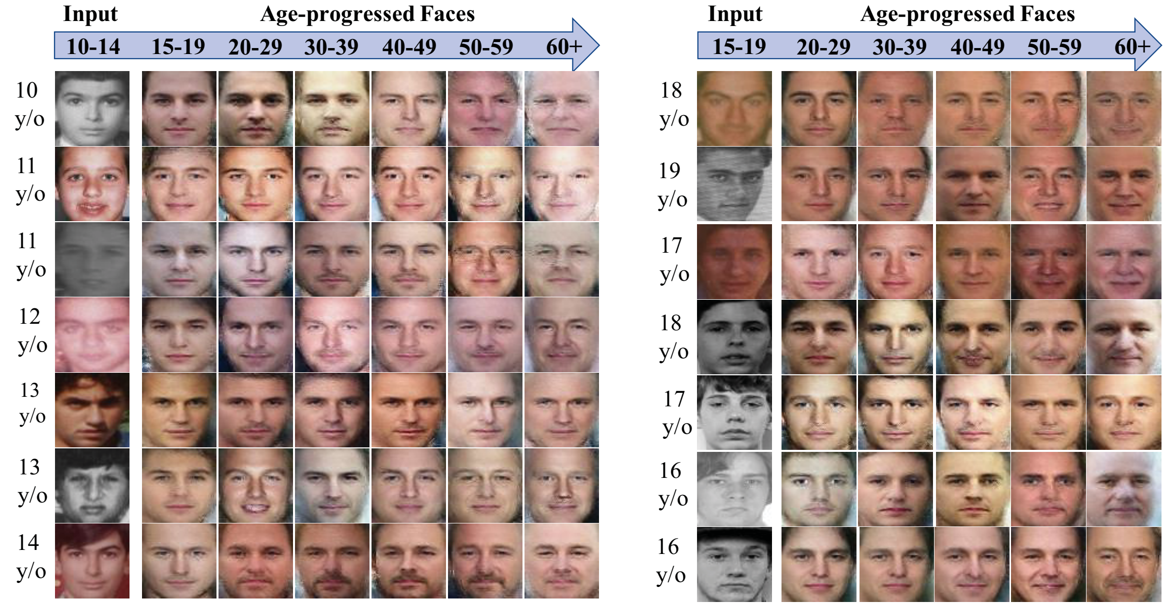


Figure 8.17: Age Progression Results against FG-NET and MORPH. Given input images, plausible age-progressed faces in different age ranges are automatically synthesized. **Best viewed in color.**

are pretrained to obtain the capability of face interpretation and high-level feature extraction. Then in the later step, our TNVP model is employed to learn the aging transformation between faces presented in the face sequences.

For the model configuration, the number of units for each mapping function is set to 10. In each mapping unit f_i, two Residual Networks with rectifier non-linearity and skip connections are set up for the two transformations $\mathcal{S}$ and $\mathcal{T}$. Each of them contains 2 residual blocks with 32 feature maps. The convolutional filter size is set to 3×3. The training time for TNVP model is 18.75 hours using a machine of Core i7-6700 @3.4GHz CPU, 64.00 GB RAM and a single NVIDIA GTX Titan X GPU and TensorFlow environment. The training batch size is 64.

8.5.2 Age Progression

After training, our TNVP age progression system is applied to all faces over 10 years old from FG-NET and MORPH. As illustrated in Figure 8.17, given input faces at different ages, our TNVP is able to synthesize realistic age-progressed faces in different age ranges. Notice that none of the images in FG-NET or MORPH is presented in the training data. From these results, one can easily see that our TNVP not only efficiently embeds the specific aging information of each age group to

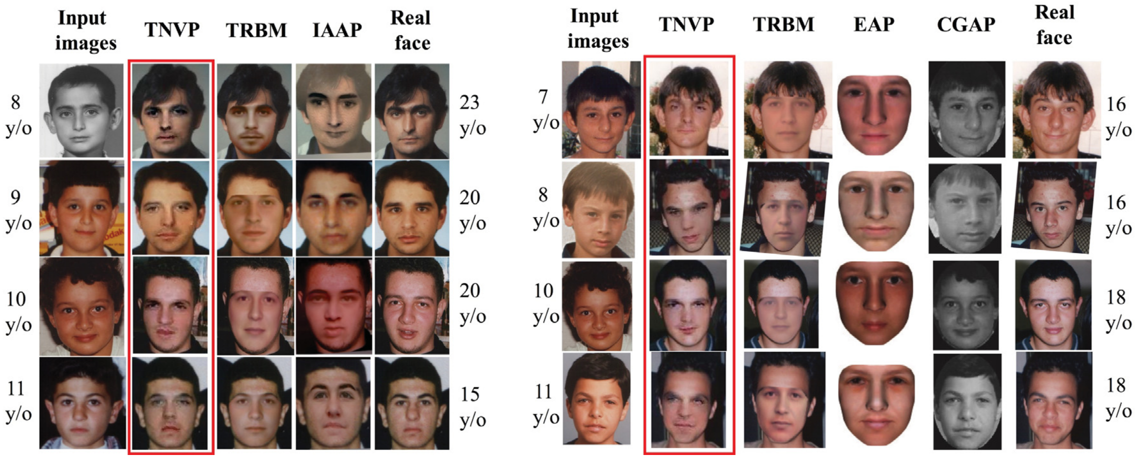
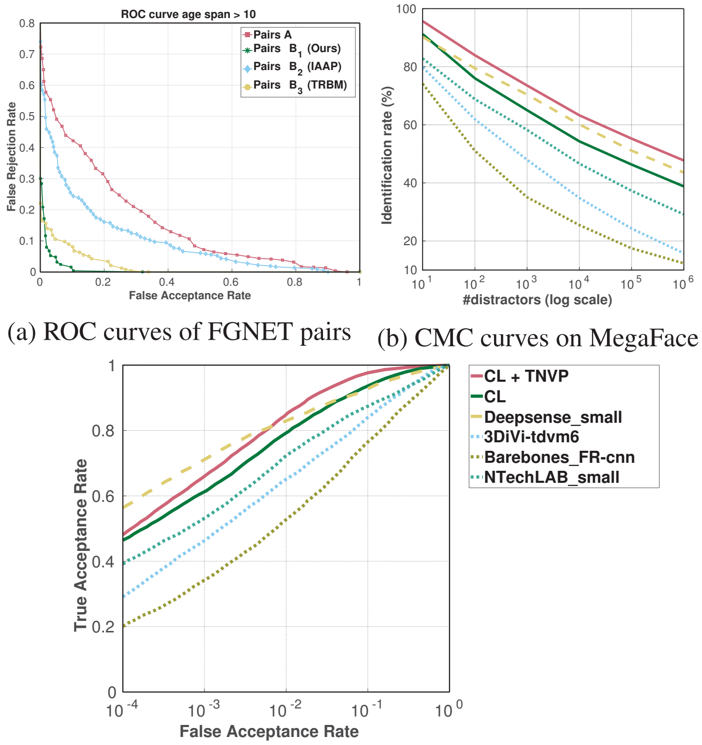


Figure 8.18: Comparisons between our TNVP against other approaches: IAAP [57], TRBM-based [89], Exemplar based (EAP) [110], and Craniofacial Growth (CGAP) [100] models. **Best viewed in color.**

the input faces but also robustly handles in-the-wild variations such as expressions, illumination, and poses. Particularly, beards and wrinkles naturally appear in the age-progressed faces around the ages of 30-49 and over 50, respectively. The face shape is also implicitly handled in our model and changes according to different individuals and age groups. Moreover, by avoiding the ℓ_2 reconstruction loss and taking the advantages of maximizing log-likelihood, sharper synthesized results with aging details are produced by our proposed model. We compare our synthesized results with other recent age progression works whose results are publicly available such as IAAP [57], TRBM-based model [89] in Figure 8.18. The real faces of the subjects at target ages are provided for reference. Other approaches, i.e. Exemplar based Age Progression (EAP) [110] and Craniofacial Growth (CGAP) model [100], are also included for further comparisons. Notice that since our TNVP model is trained using the faces ranging from 10 to 64 years old, we choose the ones with ages close to 10 years old during the comparison. These results again show the advantages of our TNVP model in term of efficiently handling the non-linear variations and aging embedding.

8.5.3 Age-Invariant face verification

This experiment validates the effectiveness of our TNVP model by showing the performance gain for cross-age face verification using our age-progressed faces. In both testing protocols, i.e. small-scale with images pairs from FG-NET and large-scale benchmark on Megaface Challenge



(a) ROC curves of FGNET pairs

(b) CMC curves on MegaFace

(c) ROC curves on MegaFace

Figure 8.19: From left to right: (a) ROC curves of face verification from 1052 pairs synthesized from different age progression methods; (b) ROC and (c) CMC curves of different face matching methods and the improvement of CL method using our age-progressed faces (under the protocol of MegaFace challenge 1).

1, we show that our aged faces can provide significant improvements on top of the face matching model without re-training on cross-age databases. We employ the deep face recognition model [134], named Center Loss (CL), which is among the state-of-the-art for this experiment.

Under the *small-scale protocol*, in FG-NET database, we randomly pick 1052 image pairs with the age gap larger than 10 years of either the same or different person. This set is denoted as $\mathbf{A}$ consisting of a positive list of 526 image pairs of the same person and a negative list of 526 image pairs of two different subjects. From each image pair of set $\mathbf{A}$, using the face with younger age, we synthesize an age-progressed face image at the age of the older one using our proposed TNVP model. This forms a new matching pair, i.e. the aged face vs. the original face at older age. Applying this process for all pairs of set $\mathbf{A}$, we obtain a new set denoted as set $\mathbf{B_1}$. To compare with IAAP [127] and TRBM [89] methods, we also construct two other sets of image pairs similarly and

Table 8.12: Rank-1 Identification Accuracy with one million Distractors (MegaFace Challenge 1 - FGNET). Protocol “small” means $\leq$0.5M images trained. “Cross-age” means trained with cross-age faces (e.g. in CACD, MORPH, etc.).

Methods	**Protocol**	**Cross-age**	**Accuracy**
Barebones_FR	Small	Y	7.136 %
3DiVi	Small	Y	15.78 %
NTechLAB	Small	Y	29.168 %
DeepSense	Small	Y	43.54 %
CL [134]	Small	**N**	38.79%
CL + TNVP	Small	**N**	**47.72%**

denote them as set $\mathbf{B_2}$ and $\mathbf{B_3}$, respectively. Then, the False Rejection Rate-False Acceptance Rate (FRR-FAR) is computed and plotted under the Receiver Operating Characteristic (ROC) curves for all methods (Fig. 8.19a). Our method achieves an improvement of **30**% on matching performance over the original pair (set **A**) while IAAP and TRBM slightly increase the rates.

In addition, our model is also experimented on the *large-scale Megaface* [58] challenge 1 with FGNET test set. Practical face recognition models should achieve high performance against having gallery set of millions of distractors and probe set of people at various ages. In this testing, 4 billion pairs are generated between the probe and gallery sets where the gallery includes one million distractors. Thus, only improvements on Rank-1 identification rate with one million distractors and verification rate at low FAR are meaningful [58]. Fig. 8.19b shows Rank-1 identification rates as the number of distractors increasing and the rates with one million distractors are shown in Table 8.12. We compute the TAR-FAR and show ROC curves[1] in Fig. 8.19c. The model from DeepSense achieves the best performance under the cross-age training set while the CL model [142] trained solely on CASIA WebFace dataset having $<$ 0.49M images without cross-age information. From these results, we show that face matching models can directly benefit from our TNVP model to improve their robustness against aging effects. Particularly, by using our age-progressed images without re-training, the CL model [142] not only obtains **10**% improvements but also outperforms other models trained with a small training set as shown in Table 8.12.

[1] The results of other methods are provided in MegaFace website.

Chapter 9

Conclusions

Motivating from the advantages of deep learning approaches, the main aims of this thesis are to demonstrate that the learning deep generative models with non-linear structure and latent variables organized in hidden layers can efficiently embed wide-range variations and structures in complex data. Four designed principles behind the four proposed deep models presented in the thesis are (1) the non-linear structure with many layers of latent variables is able to efficiently interpreting large and non-linear face variations; (2) the relationships between shape and texture of a face or between faces are efficiently embedded in latent space; (3) the probabilistic graphical model with log-likelihood objective function can produce better image synthesis quality compared to regular reconstruction loss function; and (4) deep convolutional networks can help to improve the highly non-linear feature generation.

The thesis consists of two main parts focusing on (1) single face modeling under large variations, and (2) face sequence modeling to synthesize the age-progressed faces. In the first part of the thesis, two novel deep models, named Deep Appearance Models and Robust Deep Appearance Models, are introduced to overcome the disadvantages of classical linear model such as Active Appearance Models. The proposed models have shown their potential in both tasks of learning high-level representation and face reconstruction under various challenging conditions. In their main structures, three crucial components represented in hierarchical layers are modeled using Deep Boltzmann Machines (DBM) to robustly capture the variations of facial shapes and appearances. Furthermore, by incorporating a binary mask separating “clean” and corrupted pixels, RDAM is able to efficiently handle the occluded face areas and, therefore, produces more plausible reconstruction results. These proposed approaches are evaluated in various applications to demonstrate their robustness and capabilities, e.g. facial super-resolution reconstruction, facial off-angle reconstruction, facial occlusion removal and age estimation using challenging face databases: Labeled Face Parts in the Wild (LFPW), Helen, AR, EURECOM and FG-NET.

The second part of the thesis focuses on developing deep generative models that are able to interpret the temporal relationship between images in a face sequence. Two novel models (i.e. Temporal Restricted Boltzmann Machines based and Temporal Non-volume Preserving models) are introduced and applied to solve the face age progression task. Thanks to the log-likelihood objective function together with the probabilistic graphical structures, these two models have shown their advantages not only in efficiently capturing the non-linear age related variances but also producing age-progressed faces with more aging details. Furthermore, the structure of TNVP can be transformed into a deep convolutional network while keeping the advantages of probabilistic models with tractable log-likelihood density estimation. The proposed approaches are evaluated in both synthesizing age-progressed faces and cross-age face verification and consistently shows the state-of-the-art results in various face aging databases, i.e. FG-NET, MORPH, our collected large-scale aging database named AginG Faces in the Wild (AGFW), and Cross-Age Celebrity Dataset (CACD). A large-scale face verification on Megaface challenge 1 is also performed to further show the advantages of our proposed approaches.

Future works Several extensions and applications can be further developed using the ideas of this thesis.

- **Better learning of temporal embedding.** Although Reinforcement Learning (RL) origins from solving sequential decision making tasks, the integration of RL and deep learning models in recent years has created several breakthroughs and applied in many other applications such as game playing agents, robotic control, visual tracking, etc. With the advantages of RL in discovering the temporal relationships between states for decision making and our deep generative models in synthesizing the new images, the combination between them could be a potential extension. For example, one can treat an input face image and its age as a "starting state" and face at the target age as a "goal", and then utilize RL techniques to automatically discover the relationship between these two face images. By this way, the flexibility in learning the temporal embedding can be improved significantly.

- **The integration of prior knowledge for aging model.** In chapters 6 and 7, we only considered one input image at a time for the proposed models. In reality, it is possible to have more input information such as images from the subject's family or other images of that subject in the same age group. Integrating these prior knowledge could potentially help to produce better synthesized results.

- **Deep generative model for 3D data.** Due to the limitation of information provided in 2D images especially in challenging cases such as poses, or the shape changing during aging process, an extension to 3D input data is also another possible direction for our deep generative models.

Bibliography

[1] *FG-NET Aging Database*. http://www.fgnet.rsunit.com.

[2] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *Signal Processing, IEEE Transactions on*, 54(11): 4311–4322, 2006.

[3] A. M. Albert, K. Ricanek, and E. Patterson. A review of the literature on the aging adult skull and face: Implications for forensic science research and applications. *Intl. Journal of Forensic Science*, 172:1–9, 2007.

[4] B. Amberg, A. Blake, and T. Vetter. On compositional image alignment, with an application to active appearance models. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1714–1721. IEEE, 2009.

[5] R. Anderson, B. Stenger, V. Wan, and R. Cipolla. Expressive visual text-to-speech using active appearance models. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3382–3389. IEEE, 2013.

[6] G. Antipov, M. Baccouche, and J.-L. Dugelay. Face aging with conditional generative adversarial networks. *arXiv preprint arXiv:1702.01983*, 2017.

[7] E. Antonakos, J. Alabort-i Medina, G. Tzimiropoulos, and S. Zafeiriou. Hog active appearance models. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 224–228. IEEE, 2014.

[8] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[9] Y. Bando, T. Kuratate, and T. Nishita. A simple method for modeling wrinkles on human skin. In *Computer Graphics and Applications, 2002. Proceedings. 10th Pacific Conference on*, pages 166–175. IEEE, 2002.

[10] P. N. Belhumeur, D. W. Jacobs, D. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 545–552. IEEE, 2011.

[11] Y. Bengio. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

[12] A. C. Berg and S. C. Justo. Aging of orbicularis muscle in virtual human faces. In *IV*, pages 164–168. IEEE, 2003.

[13] D. Berthelot, T. Schumm, and L. Metz. Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017.

[14] C. M. Bishop. *Pattern recognition and machine learning*. Chapter 8, 2006.

[15] L. Boissieux, G. Kiss, N. M. Thalmann, and P. Kalra. *Simulation of skin aging and wrinkles with cosmetics insight*. Springer, 2000.

[16] X. P. Burgos-Artizzu, P. Perona, and P. Dollár. Robust face landmark estimation under occlusion. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1513–1520. IEEE, 2013.

[17] D. M. Burt and D. I. Perrett. Perception of age in adult caucasian male faces: Computer graphic manipulation of shape and colour information. *Proceedings of the Royal Society of London B: Biological Sciences*, 259(1355):137–143, 1995.

[18] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011.

[19] B.-C. Chen, C.-S. Chen, and W. H. Hsu. Cross-age reference coding for age-invariant face recognition and retrieval. In *ECCV*, 2014.

[20] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2172–2180, 2016.

[21] C.-C. Chiu and S. Marsella. A style controller for generating virtual human behaviors. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 3*, pages 1023–1030. International Foundation for Autonomous Agents and Multiagent Systems, 2011.

[22] T. F. Cootes and C. J. Taylor. An algorithm for tuning an active appearance model to new data. In *BMVC*, pages 919–928. Citeseer, 2006.

[23] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Interprettting Face Images using Active Appearance Models. In *Proc. of the* 3^{rd} *Intl. Conf. on Automatic Face and Gesture Recognition*, pages 300–305, 1998.

[24] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Trans. on pattern analysis and machine intell. (TPAMI)*, 23(6):681–685, 2001.

[25] E. L. Denton, S. Chintala, R. Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in neural information processing systems*, pages 1486–1494, 2015.

[26] L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[27] R. Donner, M. Reiter, G. Langs, P. Peloschek, and H. Bischof. Fast active appearance model search using canonical correlation analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1690, 2006.

[28] G. J. Edwards, T. F. Cootes, and C. J. Taylor. Face recognition using active appearance models. In *Computer VisionECCV98*, pages 581–595. Springer, 1998.

[29] M. Ehrlich, T. J. Shields, T. Almaev, and M. R. Amer. Facial attributes classification using multi-task representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 47–55, 2016.

[30] S. A. Eslami, N. Heess, C. K. Williams, and J. Winn. The shape boltzmann machine: a strong model of object shape. *International Journal of Computer Vision*, 107(2):155–176, 2014.

[31] Y. Fu and T. S. Huang. Human age estimation with regression on discriminative aging manifold. *Multimedia, IEEE Transactions on*, 10(4):578–584, 2008.

[32] K. Fukushima and S. Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pages 267–285. Springer, 1982.

[33] Y. Ge, D. Yang, J. Lu, B. Li, and X. Zhang. Active appearance models using statistical characteristics of gabor based texture representation. *Journal of Visual Communication and Image Representation*, 24(5):627–634, 2013.

[34] X. Geng, Z.-H. Zhou, and K. Smith-Miles. Automatic age estimation based on facial aging patterns. *PAMI*, 29(12):2234–2240, 2007.

[35] M. Germain, K. Gregor, I. Murray, and H. Larochelle. Made: Masked autoencoder for distribution estimation. In *ICML*, pages 881–889, 2015.

[36] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[37] R. Gross, I. Matthews, and S. Baker. Generic vs. person specific active appearance models. *Image and Vision Computing*, 23(12):1080–1093, 2005.

[38] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017.

[39] D. Haase, E. Rodner, and J. Denzler. Instance-weighted transfer learning of active appearance models. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1426–1433. IEEE, 2014.

[40] T. Hassner, S. Harel, E. Paz, and R. Enbar. Effective face frontalization in unconstrained images. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2015. URL http://www.openu.ac.il/home/hassner/projects/frontalize.

[41] C. Häusler and A. Susemihl. Temporal autoencoding restricted boltzmann machine. *arXiv preprint arXiv:1210.8353*, 2012.

[42] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, June 2016.

[43] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

[44] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[45] G. E. Hinton and T. J. Sejnowski. Optimal perceptual inference. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 448–453. Citeseer, 1983.

[46] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

[47] X. Hou, S. Z. Li, H. Zhang, and Q. Cheng. Direct appearance models. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–828. IEEE, 2001.

[48] D. H. Hubel and T. N. Wiesel. Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, 195(1):215–243, 1968.

[49] M. J. Huiskes, B. Thomee, and M. S. Lew. New trends and ideas in visual concept detection: the mir flickr retrieval evaluation initiative. In *Proceedings of the international conference on Multimedia information retrieval*, pages 527–536. ACM, 2010.

[50] H. Jia and A. M. Martinez. Face recognition with occlusions in the training and testing sets. In *Automatic Face & Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on*, pages 1–6. IEEE, 2008.

[51] H. Jia and A. M. Martinez. Support vector machines in face recognition with occlusions. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 136–141. IEEE, 2009.

[52] Z. Jiang, Z. Lin, and L. S. Davis. Learning a discriminative dictionary for sparse coding via label consistent k-svd. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1697–1704. IEEE, 2011.

[53] S. Z. Joan Alabort-i Medina. Bayesian active appearance models. In *Computer Vision and Pattern Recognition, 2014. CVPR 2014. IEEE Conference on*, pages 3438–3445. IEEE, 2014.

[54] F. Juefei-Xu, K. Luu, M. Savvides, T. D. Bui, and C. Y. Suen. Investigating age invariant face recognition based on periocular biometrics. In *IJCB*, pages 1–7. IEEE, 2011.

[55] A. Karpathy. Convolutional neural networks for visual recognition, 2017. URL `http://cs231n.github.io`.

[56] I. Kemelmacher-Shlizerman and S. M. Seitz. Collection flow. In *CVPR*, pages 1792–1799. IEEE, 2012.

[57] I. Kemelmacher-Shlizerman, S. Suwajanakorn, and S. M. Seitz. Illumination-aware age progression. In *CVPR*, pages 3334–3341. IEEE, 2014.

[58] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *CVPR*, 2016.

[59] D. E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.

[60] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[61] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images, 2009.

[62] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[63] A. Lanitis, C. J. Taylor, and T. F. Cootes. Toward automatic simulation of aging effects on face images. *PAMI*, 24(4):442–455, 2002.

[64] H. Larochelle and I. Murray. The neural autoregressive distribution estimator. In *AISTATS*, volume 1, page 2, 2011.

[65] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*, 2015.

[66] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang. Interactive facial feature localization. In *Computer Vision–ECCV 2012*, pages 679–692. Springer, 2012.

[67] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[68] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[69] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *TPAMI*, 33(5):978–994, 2011.

[70] B. D. Lucas, T. Kanade, et al. An iterative image registration technique with an application to stereo vision. In *IJCAI*, volume 81, pages 674–679, 1981.

[71] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 94–101. IEEE, 2010.

[72] K. Luu, K. Ricanek, T. D. Bui, and C. Y. Suen. Age estimation using active appearance models and support vector machine regression. In *Biometrics: Theory, Applications, and Systems, 2009. BTAS'09. IEEE 3rd International Conference on*, pages 1–5. IEEE, 2009.

[73] K. Luu, C. Suen, T. Bui, and J. K. Ricanek. Automatic child-face age-progression based on heritability factors of familial faces. In *BIdS*, pages 1–6. IEEE, 2009.

[74] K. Luu, T. D. Bui, and C. Y. Suen. Kernel spectral regression of perceived age from hybrid facial features. In *FG'11*, pages 1–6. IEEE, 2011.

[75] K. Luu, K. Seshadri, M. Savvides, T. D. Bui, and C. Y. Suen. Contourlet appearance model for facial age estimation. In *IJCB*, pages 1–8. IEEE, 2011.

[76] A. Martınez and R. Benavente. The ar face database. *Rapport technique*, 24, 1998.

[77] I. Matthews and S. Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60(2):135–164, 2004.

[78] R. Memisevic and G. Hinton. Unsupervised learning of image transformations. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[79] R. Min, N. Kose, and J.-L. Dugelay. Kinectfacedb: A kinect database for face recognition. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 44(11):1534–1548, Nov 2014. ISSN 2168-2216. doi: 10.1109/TSMC.2014.2331215.

[80] M. Minear and D. C. Park. A lifespan database of adult facial stimuli. *Behavior Research Methods, Instruments, & Computers*, 36(4):630–633, 2004.

[81] R. Mittelman, B. Kuipers, S. Savarese, and H. Lee. Structured recurrent temporal restricted boltzmann machines. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1647–1655, 2014.

[82] A. Mollahosseini and M. H. Mahoor. Bidirectional warping of active appearance model. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, pages 875–880. IEEE, 2013.

[83] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[84] R. Navarathna, S. Sridharan, and S. Lucey. Fourier active appearance models. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1919–1926. IEEE, 2011.

[85] M. L. Ngan and P. J. Grother. Face recognition vendor test (frvt) - performance of automated age estimation algorithms. NIST Interagency Report 7995, 2014.

[86] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 689–696, 2011.

[87] C. Nhan Duong, K. G. Quach, and T. D. Bui. Are sparse representation and dictionary learning good for handwritten character recognition? In *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, pages 575–580. IEEE, 2014.

[88] C. Nhan Duong, K. Luu, K. Gia Quach, and T. D. Bui. Beyond principal components: Deep boltzmann machines for face modeling. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4786–4794, June 2015.

[89] C. Nhan Duong, K. Luu, K. Gia Quach, and T. D. Bui. Longitudinal face modeling via temporal deep restricted boltzmann machines. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5772–5780, June 2016.

[90] C. Nhan Duong, K. G. Quach, K. Luu, T. H. N. Le, and M. Savvides. Temporal non-volume preserving approach to facial age-progression and age-invariant face recognition. In *The IEEE Conference on Computer Vision (ICCV)*, pages 3735–3743, October 2017.

[91] G. Papandreou and P. Maragos. Adaptive and constrained algorithms for inverse compositional active appearance model fitting. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[92] E. Patterson, K. Ricanek, M. Albert, and E. Boone. Automatic representation of adult aging in facial images. In *Proc. IASTED Intl Conf. Visualization, Imaging, and Image Processing*, pages 171–176, 2006.

[93] E. Patterson, A. Sethuram, M. Albert, and K. Ricanek. Comparison of synthetic face aging to age progression by forensic sketch artist. In *IASTED International Conference on Visualization, Imaging, and Image Processing, Palma de Mallorca, Spain*, pages 247–252, 2007.

[94] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 313–318. ACM, 2003.

[95] D. Pizarro, J. Peyras, and A. Bartoli. Light-invariant fitting of active appearance models. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–6. IEEE, 2008.

[96] K. G. Quach, C. Nhan Duong, and T. D. Bui. Sparse representation and low-rank approximation for robust face recognition. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 1330–1335. IEEE, 2014.

[97] K. G. Quach, C. Nhan Duong, K. Luu, and T. D. Bui. Robust deep appearance models. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 390–395. IEEE, 2016.

[98] K. G. Quach, C. N. Duong, K. Luu, and T. D. Bui. Non-convex online robust pca: Enhance sparsity via p-norm minimization. *Computer Vision and Image Understanding*, 158:126–140, 2017.

[99] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[100] N. Ramanathan and R. Chellappa. Modeling age progression in young faces. In *CVPR*, volume 1, pages 387–394. IEEE, 2006.

[101] N. Ramanathan and R. Chellappa. Modeling shape and textural variations in aging faces. In *FG'08.*, pages 1–8. IEEE, 2008.

[102] K. Ricanek Jr and T. Tesafaye. Morph: A longitudinal image database of normal adult age-progression. In *FGR 2006.*, pages 341–345. IEEE, 2006.

[103] M. Rosca, B. Lakshminarayanan, D. Warde-Farley, and S. Mohamed. Variational approaches for auto-encoding generative adversarial networks. *arXiv preprint arXiv:1706.04987*, 2017.

[104] D. Rowland, D. Perrett, et al. Manipulating facial appearance through shape and color. *Computer Graphics and Applications, IEEE*, 15(5):70–76, 1995.

[105] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. A semi-automatic methodology for facial landmark annotation. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, pages 896–903. IEEE, 2013.

[106] R. Salakhutdinov and G. Hinton. An efficient learning procedure for deep boltzmann machines. *Neural computation*, 24(8):1967–2006, 2012.

[107] R. Salakhutdinov and G. E. Hinton. Deep boltzmann machines. In *Intl. Conf. on Artificial Intell. and Statistics*, pages 448–455, 2009.

[108] R. R. Salakhutdinov. Learning in markov random fields using tempered transitions. In *Advances in neural information processing systems*, pages 1598–1606, 2009.

[109] J. Saragih and R. Goecke. A nonlinear discriminative approach to aam fitting. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[110] C.-T. Shen, W.-H. Lu, S.-W. Shih, and H.-Y. M. Liao. Exemplar-based age progression prediction in children faces. In *ISM*, pages 123–128. IEEE, 2011.

[111] X. Shu, J. Tang, H. Lai, L. Liu, and S. Yan. Personalized age progression with aging dictionary. In *ICCV*, December 2015.

[112] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[113] P. Smolensky. Information processing in dynamical systems: Foundations of harmony theory. 1986.

[114] N. Srivastava and R. Salakhutdinov. Multimodal learning with deep boltzmann machines. In *Advances in neural information processing systems*, pages 2222–2230, 2012.

[115] J. Sung and D. Kim. Pose-robust facial expression recognition using view-based 2D + 3D AAM. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 38(4):852–866, 2008.

[116] J. Suo, S.-C. Zhu, S. Shan, and X. Chen. A compositional and dynamic model for face aging. *PAMI*, 32(3):385–401, 2010.

[117] J. Suo, X. Chen, S. Shan, W. Gao, and Q. Dai. A concatenational graph evolution aging model. *PAMI*, 34(11):2083–2096, 2012.

[118] I. Sutskever and G. E. Hinton. Learning multilevel distributed representations for high-dimensional sequences. In *International Conference on Artificial Intelligence and Statistics*, pages 548–555, 2007.

[119] I. Sutskever, G. E. Hinton, and G. W. Taylor. The recurrent temporal restricted boltzmann machine. In *Advances in Neural Information Processing Systems*, pages 1601–1608, 2009.

[120] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[121] X. Tan, S. Chen, Z.-H. Zhou, and J. Liu. Face recognition under occlusions and variant expressions with partial similarity. *Information Forensics and Security, IEEE Transactions on*, 4(2):217–230, 2009.

[122] Y. Tang, R. Salakhutdinov, and G. Hinton. Robust boltzmann machines for recognition and denoising. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2264–2271. IEEE, 2012.

[123] G. W. Taylor and G. E. Hinton. Factored conditional restricted boltzmann machines for modeling motion style. In *Proceedings of the 26th annual international conference on machine learning*, pages 1025–1032. ACM, 2009.

[124] G. W. Taylor, G. E. Hinton, and S. T. Roweis. Modeling human motion using binary latent variables. In *Advances in neural information processing systems*, pages 1345–1352, 2006.

[125] G. W. Taylor, L. Sigal, D. J. Fleet, and G. E. Hinton. Dynamical binary latent variable models for 3d human pose tracking. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 631–638. IEEE, 2010.

[126] K. T. Taylor. *Forensic Art and Illustration*. CRC Press, 2000.

[127] M.-H. Tsai, Y.-K. Liao, and I.-C. Lin. Human face aging with guided prediction and detail synthesis. *Multimedia tools and applications*, 72(1):801–824, 2014.

[128] G. Tzimiropoulos and M. Pantic. Optimization problems for fast aam fitting in-the-wild. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 593–600. IEEE, 2013.

[129] G. Tzimiropoulos, J. Alabort-i Medina, S. Zafeiriou, and M. Pantic. Active orientation models for face alignment in-the-wild. *TIFS*, 9(12):2024–2034, 2014.

[130] L. Van Der Maaten and E. Hendriks. Capturing appearance variation in active appearance models. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 34–41. IEEE, 2010.

[131] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3360–3367. IEEE, 2010.

[132] W. Wang, Z. Cui, Y. Yan, J. Feng, S. Yan, X. Shu, and N. Sebe. Recurrent face aging. In *CVPR*, pages 2378–2386, 2016.

[133] Z. Wang and A. C. Bovik. Mean squared error: love it or leave it? a new look at signal fidelity measures. *Signal Processing Magazine, IEEE*, 26(1):98–117, 2009.

[134] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, pages 499–515. Springer, 2016.

[135] Y. Wu, Z. Wang, and Q. Ji. Facial feature tracking under varying facial expressions and face poses based on restricted boltzmann machines. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3452–3459. IEEE, 2013.

[136] J. Xing, Z. Niu, J. Huang, W. Hu, and S. Yan. Towards multi-view and partially-occluded face alignment. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1829–1836. IEEE, 2014.

[137] C.-Y. Yang, S. Liu, and M.-H. Yang. Structured face hallucination. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1099–1106. IEEE, 2013.

[138] H. Yang, D. Huang, Y. Wang, H. Wang, and Y. Tang. Face aging effect simulation using hidden factor analysis joint sparse representation. *TIP*, 25(6):2493–2507, 2016.

[139] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution via sparse representation. *Image Processing, IEEE Transactions on*, 19(11):2861–2873, 2010.

[140] M. Yang and L. Zhang. Gabor feature based sparse representation for face recognition with gabor occlusion dictionary. In *Computer Vision–ECCV 2010*, pages 448–461. Springer, 2010.

[141] M. Yang, L. Zhang, X. Feng, and D. Zhang. Fisher discrimination dictionary learning for sparse representation. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 543–550. IEEE, 2011.

[142] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014.

[143] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[144] M. D. Zeiler, G. W. Taylor, L. Sigal, I. Matthews, and R. Fergus. Facial expression transfer with input-output temporal restricted boltzmann machines. In *Advances in Neural Information Processing Systems*, pages 1629–1637, 2011.

[145] J. Zhao, M. Mathieu, and Y. LeCun. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*, 2016.

[146] J. Zhu, S. C. Hoi, and M. R. Lyu. Real-time non-rigid shape recovery via active appearance models for augmented reality. In *Computer Vision–ECCV 2006*, pages 186–197. Springer, 2006.