PhD Dissertation

International Doctorate School in Information and Communication Technologies

DISI - University of Trento

HUMAN FACE AND BEHAVIOR ANALYSIS

Wei Wang

Advisor:

Prof. Nicu Sebe

Università degli Studi di Trento

April 2018

Publications

This thesis consists of the following publications:

- Chapter 2:
  Wei Wang, Yan Yan, L. Nie, L. Zhang, Stefan Winkler, Nicu Sebe: Sparse Code Filtering for Action Pattern Mining. Asian Conference on Computer Vision (ACCV), 2016

- Chapter 3
  Wei Wang, Sergey Tulyakov, NicuSebe: Recurrent convolutional face alignment. Asian Conference on Computer Vision (ACCV), 2016
  Wei Wang, Sergey Tulyakov, Nicu Sebe: Recurrent Convolutional Shape Regression. IEEE Transaction on Pattern Analysis and Machine Intelligence (TPAMI). 2018.

- Chapter 4
  Wei Wang, Zhen Cui,Yan Yan, Jiashi Feng,Shuicheng Yan, Xiangbo Shu, NicuSebe: Recurrent Face Aging. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
  Wei Wang, Yan Yan, Zhen Cui, Jiashi Feng, Shuicheng Yan, Nicu Sebe."Recurrent Face Aging with Hierarchical AutoRegressive Memory." IEEE Transaction on Pattern Analysis and Machine Intelligence (TPAMI). 2018.

- Chapter 5
  Wei Wang, Xavier Pineda, Dan Xu, Pascal Fua, Elisa Recci, Nicu Sebe: Every Smile is Unique: Landmark-Guided Diverse Smile Generation. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018
  Wei Wang, Xavier Pineda, Dan Xu, Elisa Recci, Nicu Sebe: Enforcing Diversity in Landmark-Guided Smile Generation. *submitted to* International Journal of Computer Vision (IJCV), 2018 *under review.*

The following are the papers published during the course of the Ph.D but not included in this thesis:

- **Wei Wang**, Xavier Pineda, Dan Xu, Elisa Recci, Nicu Sebe: Learning How to Smile: Expression Video Generation with Conditional Adversarial Recurrent Nets. *submitted to* IEEE Transaction on Pattern Analysis and Machine Intelligence (TPAMI), 2018 *under major revision*.

- **Wei Wang**, Yan Yan, Feiping Nie, Shuicheng Yan, Nicu Sebe: Flexible Manifold Learning with Optimal Graph for Image and Video Representation. IEEE Transaction on Image Processing (TIP), 2018

- Dan Xu, **Wei Wang**, Hao Tang, Nicu Sebe: Structured Attention Guided Convolutional Neural Fields for Monocular Depth Estimation. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018

- Siyuan Hao, **Wei Wang**, Yuanxin Ye, Lorenzo Bruzzone: A Deep Network Architecture for Super-resolution aided Hyperspectral Image Classification with Contrastive Class-wise Loss. IEEE Transaction on GeoScience and Remote Sensing (TGRS), 2018

- Siyuan Hao, **Wei Wang**, Yan Yan, Lorenzo Bruzzone: Class-wise Dictionary Learning for Hyperspectral Image Classification. Neurocomputing, 2017

- Siyuan Hao, **Wei Wang**, Yuanxin Ye, Tingyuan Nie, Lorenzo Bruzzone: Two-stream Deep Architecture for Hyperspectral Image Classification. IEEE Transaction on GeoScience and Remote Sensing (TGRS), 2017

- X. Li, Z. Jie, **Wei Wang**, C. Liu, J. Yang, X. Shen, Z. Lin, Q. Chen, S. Yan, J. Feng: FoveaNet: Perspective-aware Urban Scene Parsing. IEEE International Conference on Computer Vision. (ICCV), 2017

- S. Xiao, J. Feng, L. Liu, X. Nie, **Wei Wang**, S. Yan, A. Kassim: Recurrent 3D-2D Dual Learning for Large-pose Facial Landmark Detection. IEEE International Conference on Computer Vision. (ICCV), 2017

- **Wei Wang**, Yan Yan, Stefan Wrinkle, Nicu Sebe: Category Specific Dictionary Learning for Attribute Specific Feature Selection. IEEE Transaction on Image Processing (TIP), 2016

- **Wei Wang**, YanYan, Feiping Nie, Xavier Alameda Pineda, Shuicheng Yan, NicuSebe: Projective Unsupervised Flexible Embedding with Optimal Graph. British Machine Vision Conference, (BMVC), 2016

- **Wei Wang**, Yan Yan, Luming Zhang, Richang Hong, Nicu Sebe: Collaborative Sparse Coding for Multi-view Action Recognition." Proceedings of the IEEE Multimedia Magazine. 2016

- **Wei Wang**, Yan Yan, NicuSebe: Attribute guided dictionary learning. Proceedings of the ACM on International Conference on Multimedia Retrieval (ICMR), 2015

Abstract

*Human face and behavior analysis are very important research topics in the field of computer vision and they have broad applications in our everyday life. For instance, face alignment, face aging, face expression analysis and action recognition have been well studied and applied for security and entertainment. With these face analyzing techniques (*e.g.*, face aging), we could enhance the performance of cross-age face verification system which now has been used for banks and electronic devices to recognize their clients. With the help of action recognition system, we could better summarize the user uploaded videos or generate logs for surveillance videos. This could help us retrieve the videos more accurately and easily.*

The dictionary learning and neural networks are powerful machine learning models for these research tasks. Initially, we focus on the multi-view action recognition task. First, a class-wise dictionary is pre-trained which encourages the sparse representations of the between-class videos from different views to lie close by. Next, we integrate the classifiers and the dictionary learning model into a unified model to learn the dictionary and classifiers jointly.

*For face alignment, we frame the standard cascaded face alignment problem as a recurrent process by using a recurrent neural network. Importantly, by combining a convolutional neural network with a recurrent one we alleviate hand-crafted features to learn task-specific features. For human face aging task, it takes as input a single image and automatically outputs a series of aged faces. Since human face aging is a smooth progression, it is more appropriate to age the face by going through smooth transitional states. In this way, the intermediate aged faces between the age groups can be generated. Towards this target, we employ a recurrent neural network. The hidden units in the RFA are connected autoregressively allowing the framework to age the person by referring to the previous aged faces. For smile video generation, one person may smile in different ways (*e.g. *closing/opening the eyes or mouth). This is a one-to-many image-to-video generation problem, and we introduce a deep neural architecture named conditional multi-mode network (CMM-Net) to approach it. A multi-mode recurrent generator is trained to induce diversity and generate K different sequences of video frames.*

Keywords Face Aging, Face Alignment, Video Generation, Multi-view Action Recognition, Dictionary Learning, Deep Learning, Recurrent Neural Network.

Acknowledgements

The past years as a PhD student are very important for my life. During this period of time, I was very lucky to meet a lot of amazing people who taught me a lot and inspired me to purse my goals. I have also been able to travel to many incredible places, attend conferences, make friends and I really enjoy it. I will cherish these good memories forever.

I would like to thank all the people encountered during this fabulous journey. First of all, my advisor Nicu Sebe, who helped me navigate through the thunderstorms and found the Eden of research. I'm really grateful for your trust, mentorship and support. 'Grazie mile' is definitely far from enough to express my gratitude. I hope I could live up to your expectations and I feel so lucky to have you as my advisor. Next, I would like to thank Yan Yan, who helped me to transform from a green hand to a skilled researcher and helped me to finish my first work. I would also like to thank Shuicheng Yan and Jiashi Feng, who host me at Learning and Vision Lab in National University in Singapore during my first and second internship, and I made a lot of friends there.

I'm very grateful and proud to be a member of MHUG group and I have spent a very good time with my lovely group members. I would like to thank my colleagues from our MHUG family. Thanks, Elisa, Xavi, Yan, Sergey, Enver, Gloria, Mihai, John, Dan, Hao, Ionut, Andrea and Aliaksandr. I will never forget the time when we stay up for the conference deadlines.

I would also like to express my gratitude to my family, especially my wife, Siyuan. I hope I can make you proud of me. Without the support of my family through this time, I would not have lived such a happy and colorful life.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this section, we present the motivations behind our work, and describe the contributions chapter by chapter, including references to the corresponding publications.

1.1 Motivations

Human face and behavior analysis is a fundamental and challenging problem in computer vision. To analyze human behaviors in videos, action recognition has received increasing attention during the last decade. Action recognition has wide applications, such as human-computer interactive games, search engines, and online video surveillance systems. Videos can be summarized by labels if the actions can be annotated automatically. Then a search engine can make better recommendations (*e.g., finding dunks in basketball games*). Usually, the same action observed from different viewpoints has considerable differences. Therefore, an efficient method to extract robust view-invariant features is essential for multi-view action recognition. In the real world, user uploaded videos are usually recorded from different views. Therefore, how to design an action classifier which is robust to view changes is a very challenging problem. One of the solutions is to learn an effective video representation which is robust to view changes.

Apart from human behavior analysis, human face analysis is another very important research field, such as face aging, and face expression analysis. Face aging, also known as age progression, is attracting more and more research interest. It has wide applications in various domains including cross-age face verification Park et al. [2010] and finding lost children. When people grow old, it makes the face verification task more challenging because of the gap between the young and old faces of the same person. Thus, modeling the aging process of human faces is important for cross-age face verification and recognition. In recent years, face aging has witnessed many breakthroughs and a number of face aging models have been proposed. However, it remains a very challenging task in practice. Generally, face aging follows some common patterns of the human aging process. For kids, the main appearance change is the shape change

caused by cranium growth. For adults, the appearance change is mainly reflected in wrinkles Suo et al. [2012].

Besides face aging, facial expression analysis is another important research topic as facial expressions are one of the –if not *the*– most prominent non-verbal signals for human communication Vinciarelli et al. [2009]. The automatic recognition of facial expressions has been studied for the last decades. Indeed, a plethora of discriminative approaches, aiming to learn the boundaries between various categories in different video sequence representation spaces, were proposed to tackle the recognition of facial expressions. Naturally, these approaches focus on recognizing the dynamics of the different signals/expressions. Even if their performance is, specially lately, very impressive, these methods do not posses the ability to reproduce the dynamics of the patterns they accurately classify. How to generate realistic facial expressions is a scientific challenge yet to be soundly addressed, and learning to generate facial expressions can help us better understand the dynamics of facial action units.

For face analysis (*e.g.*, face aging, facial expression generation), one of the fundamental techniques is face alignment which is usually employed as the preprocessing step. For instance, when we do face verification, the faces need to be aligned first. Besides, face pose and expression normalization is also a very important preprocessing step. The current approaches usually rely on hand-crafted features, and they could not employ the power of neural networks.

In this thesis, we investigate both human behavior and human face analysis. My Ph.D research consists of two parts. The first part focuses on multi-view action recognition problem where we employ the dictionary learning method to mine the representation of action patterns which are robust to view changes. The second part focuses on face analysis which includes face alignment, face aging and smile video generation. All of these tasks are closely related to sequence processing. Therefore, we employ the recurrent neural networks, such as the Long Short Term Memory (LSTM) and the Gated Recurrent Unit (GRU) to solve these problems.

1.2 Contributions and Structure of the Thesis

Chapter 2 introduces human behavior analysis. We will focus on multi-view action recognition task which employs dictionary learning methods. Chapters 3, 4, 5 present human face analysis (*i.e., face alignment, face aging and face expression generation*) with deep recurrent neural networks. Chapter 6 presents the conclusions and the future research directions. This thesis makes the following contributions towards human behavior and face analysis.

Contribution 1 (Chapter 2). *For multi-view action recognition, we integrate the classifiers and the class-wise sparse coding process into a unified collaborative filtering (CF) framework to mine the discriminative sparse codes which are robust to view changes.*

In Chapter 2, we study the multi-view action recognition for human behavior analysis. Various approaches have been proposed to encode the videos that contain actions, among which self-similarity matrices (SSMs) have shown very good performance by encoding the dynamics of the video. However, SSMs become sensitive when there is a very large view change. In this chapter, we tackle the multi-view action recognition problem by proposing a sparse code filtering (SCF) framework which can mine the action patterns. First, a class-wise sparse coding method is proposed to make the sparse codes of the between-class data lie close by. Then we integrate the classifiers and the class-wise sparse coding process into a collaborative filtering (CF) framework to mine the discriminative sparse codes and classifiers jointly. The experimental results on several public multi-view action recognition datasets demonstrate that the presented SCF framework outperforms other state-of-the-art methods.

Contribution 2 (Chapter 3). *For face alignment, we frame the standard cascaded face alignment problem as a recurrent process by using a recurrent neural network with the gated recurrent unit, and we propose a novel recurrent convolutional face alignment method to solve this problem.*

In Chapter 3, we propose a new Recurrent Convolutional Face Alignment method. Mainstream direction in face alignment is now dominated by cascaded regression methods. These methods start from an image with an initial shape and build a set of shape increments by computing features with respect to the current shape estimate. These shape increments move the initial shape to the desired location. Despite the advantages of the cascaded methods, they all share two major limitations: (i) shape increments are learned separately from each other in a cascaded manner, (ii) the use of standard generic computer vision features such SIFT, HOG, does not allow these methods to learn problem-specific features. We propose a new Recurrent Convolutional Face Alignment method that overcomes these limitations. We frame the standard cascaded alignment problem as a recurrent process and learn all shape increments jointly, by using a recurrent neural network with the gated recurrent unit. Importantly, by combining a convolutional neural network with a recurrent one we alleviate hand-crafted features, widely adopted in the literature and thus allowing the model to learn task-specific features. Moreover, both the convolutional and the recurrent neural networks are learned jointly. Experimental evaluation shows that the proposed method has better performance than the state-of-the-art methods, and further support the importance of learning a single end-to-end model for face alignment.

Contribution 3 (Chapter 4). *For face aging, we propose a Recurrent Face Aging (RFA) framework which takes as input a single image and automatically outputs a series of aged faces. The*

hidden units in the RFA are connected autoregressively allowing the framework to age the person by referring to the previous aged faces.

In Chapter 4, we propose a Recurrent Face Aging (RFA) framework which takes as input a single image and automatically outputs a series of aged faces. The hidden units in the RFA are connected autoregressively allowing the framework to age the person by referring to the previous aged faces. Due to the lack of labeled face data of the same person captured in a long range of ages, traditional face aging models split the ages into discrete groups and learn a one-step face transformation for each pair of adjacent age groups. Since human face aging is a smooth progression, it is more appropriate to age the face by going through smooth transitional states. In this way, the intermediate aged faces between the age groups can be generated. Towards this target, we employ a recurrent neural network whose recurrent module is a hierarchical triple-layer gated recurrent unit which functions as an autoencoder. The bottom layer of the module encodes the input to a latent representation, and the top layer decodes the representation to a corresponding aged face. The experimental results demonstrate the effectiveness of our framework.

Contribution 4 (Chapter 5). *To train a computer to make someone smile in different ways (*e.g. *closing/opening the eyes or mouth), we introduce a deep neural architecture named conditional multi-mode network (CMM-Net). CMM-Net is trained to induce diversity and generate different sequences of landmark images which are then translated into video sequences.*

In Chapter 5, we propose a video generation framework which can generate diverse smile videos with only one input neutral face and smile label. People never smile twice the same way, and still we are able to recognize smiles in a blink of an eye. The diversity in human smiles is notorious: in this paper we wonder if we can train a computer to make someone smile in different ways (*e.g.* closing/opening the eyes or mouth). This is a one-to-many image-to-video generation problem, and we introduce a deep neural architecture named conditional multi-mode network (CMM-Net) to approach it. The smile dynamics are captured thanks to an embedded landmark representation learned by means of a variational auto-encoder. A conditional recurrent network is used to generate a sequence conditioned to a class label (*e.g.* posed smile). This first sequence is then fed to the multi-mode recurrent landmark generator trained to induce diversity and generate K different sequences of landmark images. Finally, the landmark sequences are translated into video sequences. The experimental results demonstrate the effectiveness of the proposed CMM-Net in enforcing diversity in landmark-guided smile generation.

Chapter 2

Multi-view Action Recognition

2.1 Introduction

Action recognition has wide applications, such as human-computer interactive games, search engines, and online video surveillance systems. Videos can be summarized by labels if the actions can be annotated automatically. Then a search engine can make better recommendations (*e.g., finding dunks in basketball games*). Usually, the same action observed from different viewpoints has considerable differences. Therefore, an efficient method to extract robust view-invariant features is essential for multi-view action recognition. The features can be roughly grouped into two types, the 2D features Cai et al. [2014] and 3D features Vemulapalli et al. [2014].

Many works employed 3D models to tackle the multi-view action recognition problem. First, the geometric transitions are utilized to obtain projections across different viewpoints. Then the observations are compared with the projections to find the viewpoint that best matches the observations Lv and Nevatia [2007]. However, how to accurately find body joints to build the 3D model remains an open problem. Besides, the built model has too many degree-of-freedom parameters, which must be carefully calibrated. Moreover, the model requires high resolution videos to locate body joints and sometimes may require mocap data Peursum et al. [2007]. An alternative solution for multi-view action recognition is to design view-invariant 2D features. Farhadi and Tabrizi [2008] proposed split-based representations by clustering the similar video frames into splits. The split-based representations can be transferred among different viewpoints as the change dynamics of the multi-view videos are the same. Similarly, Junejo et al. [2011] employed SSMs to encode the frame-to-frame relative changes. However, the SSMs are robust to view changes only to a certain extent.

In this paper, to tackle the multi-view problem, we propose a class-wise sparse coding approach to maintain label consistency. We employ SSM feature to represent each video. The sparse coding learns a dictionary from SSM representations of the video collections. The dic-

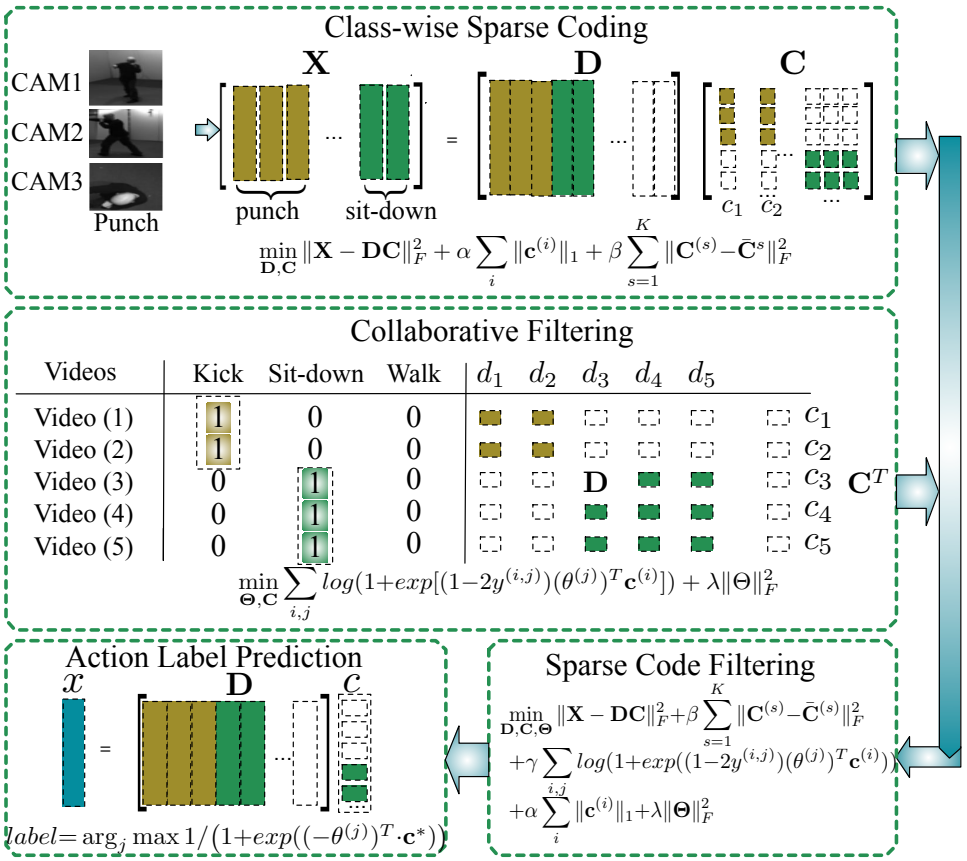


Figure 2.1: Overview of sparse code filtering: (top) Class-wise sparse coding. (middle) Collaborative Filtering. (bottom right) Sparse code filtering framework. (bottom left) Label prediction.

tionary consists of typical action patterns, and each video is encoded to a code as a linear combination of action patterns. The label consistency is achieved by penalizing the within class variance of the codes. Thus, the codes of the within-class videos will lie close by, and accordingly, only the view-invariant action patterns will be learned while the view-dependent information will be suppressed. Then we rely on the codes as video features to do action classification.

To further improve the discriminative power of the codes, we integrate the class-wise sparse coding and classifiers training process into a unified CF framework as shown in Fig.5.2. This is because CF can link the dictionary and classifiers together which can optimize them jointly. The dictionary can be adjusted for the classifiers while the classifiers can be adjusted for the dictionary collaboratively. In this way, the learned action patterns in the dictionary can be more discriminative with respect to different actions. Thus, we derive a novel sparse code filtering framework. In the sparse code filtering scheme, each action class is regarded as an user. For the classical collaborative filtering, the entry in the rating matrix (e.g., ranges from 0 to 5)

describes how much a user likes the product. In our scheme, however, the entry in the rating matrix, ranging from 0 to 1, represents the probability that a video belongs to an action class. The sparse code filtering framework provides a trade-off between the dictionary reconstruction error and the classification error which derive from the class-wise sparse coding and the logistic classifiers respectively.

To summarize, our work makes the following contributions: (i) We propose a class-wise sparse coding approach to maintain the label consistency by encouraging the sparse codes of the multi-view videos within the same action class to lie close by. (ii) We propose a novel sparse code filtering framework in which the classifiers and dictionary can be optimized collaboratively. Thus, the view-invariant and class-discriminant sparse codes can be learned. (iii) The proposed sparse code filtering framework has a good generalization property and can be applied to other pattern recognition tasks.

2.2 Related Work

2.2.1 Action Recognition

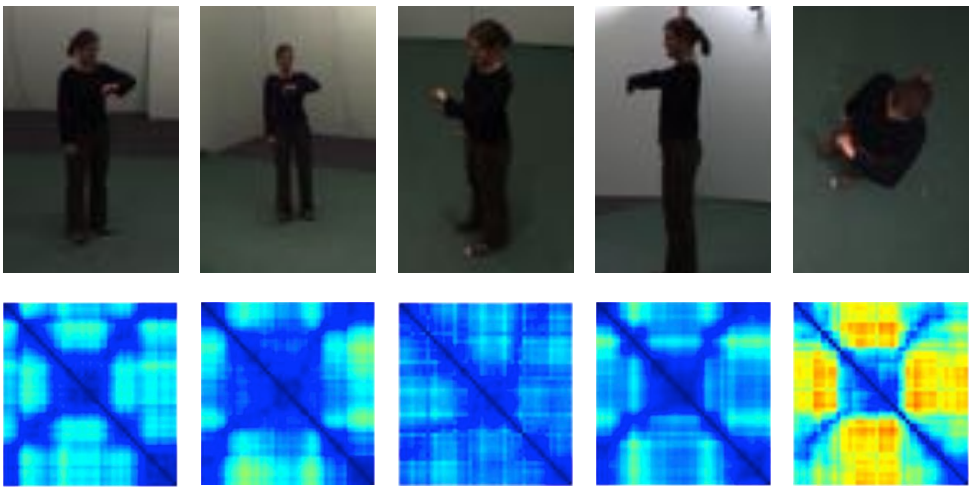


Figure 2.2: SSM features extracted from different views.

Many 3D and 2D based approaches are proposed for action recognition. Through reconstructing 3D human bodies, features can be adapted across different viewpoints through geometric transformation. Weinland et al. [2007] projected 3D poses into 2D to obtain arbitrary views and employed an exemplar-based HMM to model view transformations. A similar idea is proposed in Natarajan and Nevatia [2008] which employed CRF instead of HMM. Except for designing the 3D models, some works focus on designing view-invariant classifiers, such as linear discriminant analysis Yan et al. [2014] and latent multi-task learning Mahasseni and

Todorovic [2013]. Matikainen et al. [2012] suggested training models for all the views and then utilizing recommender system to find the suitable model. But the approach in Matikainen et al. [2012] requires huge amount of training samples from different viewpoints. Recently, the recurrent neural network is also applied for the action recognition task Du et al. [2015] as it is good at dealing with signal sequences with various lengths Wang et al. [2016b,d]. However, these methods can only tackle the single-view action recognition task.

To achieve view invariance in 2D models, many works try to extract view-invariant features. Farhadi and Tabrizi [2008] proposed a split-based representation by clustering video frames into splits. Then videos can be represented by the statistics of the splits, and the split transfer mapping across views can be learned. Based on 2D features, the transfer learning model requires no 3D human reconstructions. Recently, a more robust view-invariant descriptor, self-similarity matrix (SSM) Junejo et al. [2011] has been proposed. It is relatively stable over the viewpoint changes compared with other features Junejo et al. [2008]. Similarly to Farhadi and Tabrizi [2008], this descriptor encodes the relative changes between pairs of frames, and completely discards the absolute features of each single frame. SSMs can be calculated using different low-level features which have similar properties.

Fig.2.2 shows the examples from the action videos and their corresponding SSM features. From Fig.2.2, we can observe that the SSM features from the 4 side cameras are visually similar, while the feature from camera 5 (on the ceiling) is quite different. Yan et al. [2014] revealed that SSMs became less reliable when there was a very large view change. Based on SSMs, Joint Self-Similarity Volume (SSV) was introduced by Sun et al. [2011] which utilized Joint Recurrence Plot (JRP) theory to extend SSM. But different from Junejo et al. [2008], the SSM defined in Sun et al. [2011] is the recurrence-plot matrix of the vector representation of each single frame.

2.2.2 Sparse Coding

Sparse coding, also known as dictionary learning, aims to construct efficient representations of data as a combination of a few typical patterns (dictionary bases). Wang et al. [2016f, 2015] used the sparse coding for attribute detection. Raina et al. [2007] showed that sparse coding significantly improved classification performance. Luo et al. [2013] employed sparse coding for action recognition from depth maps. However, their approach is restricted to the videos which can provide depth information. Qiu et al. [2011] selected a set of more compact and discriminative bases from the dictionary using Gaussian Process. Guha and Ward [2012] investigated different sparse coding strategies, namely, an overall dictionary for all the classes, different dictionaries for each class, and their concatenation. But view changes were not considered. Zheng and Jiang [2013] proposed view-specific sparse coding. But sufficient training data from each viewpoint are required. Besides, the label information is discarded. Thus, it can not preserve la-

bel consistency. In our work, we add within-class variance into the loss function to preserve the label-consistency. The learned class-wise dictionary can be considered as a more label-smooth feature space compared with the original video feature space.

2.2.3 Collaborative Filtering

Collaborative filtering is widely used in recommendation systems of commercial websites, such as Amazon and eBay, to recommend products to their consumers. The most attractive characteristic of CF is that it can learn a good set of features automatically Goldberg et al. [1992], which does not require hand-designed features. Taking the movie recommendation system for example, each movie has its own features and each user has its own specific feature preference weights. Given the movie-user rating matrix, CF learn a good set of features for each movie and feature preference weights for each user jointly. During the CF learning process, the features will be adjusted for the feature preference weights for each user, and feature preference weights will also be adjusted for the features iteratively. Inspired by the movie recommendation system, we employ a CF framework to learn class-wise dictionary and classifiers jointly. Thus, the codes and classifiers can be adjusted to better fit each other. The experimental results demonstrate the effectiveness of our framework.

The rest of the paper is organized as follows. We propose the class-wise dictionary learning approach in the Section 3. Section 4 presents our sparse code filtering scheme. The experiments are described in Section 5. We conclude our paper in Section 6.

2.3 Class-wise Sparse Coding

The input of the sparse model is the descriptors for n_v videos, where each video is represented by a d-dimension vector $\mathbf{x}_d$. Let $\mathbf{X}_{d\times n_v}$ be the matrix by stacking the all the training video descriptors. In our model, $\mathbf{x}_d$ is the SSM feature. The outputs are the dictionary $\mathbf{D}$ and sparse codes $\mathbf{C}$. The loss of the *classical* sparse coding model, which considers reconstruction error and sparsity, is defined as:

$$\mathbf{L}(\mathbf{X};\mathbf{D},\mathbf{C})=\|\mathbf{X}-\mathbf{D}\mathbf{C}\|_F^2+\alpha\sum_{i=1}^{n_v}\|\mathbf{c}^{(i)}\|_1 \tag{2.1}$$

In Eq. 2.1, $\mathbf{D}_{d\times n}$ represents the learned dictionary and each column vector in the dictionary represents a typical action pattern, n is the number of typical patterns, $\mathbf{C}_{n\times n_v}$ is the sparse code matrix, whose i-th column, $\mathbf{c}^{(i)}$, is the sparse code of sample i. l_1-norm is a lasso constraint which encourages sparsity, and α balances the reconstruction error and the sparsity penalty.

In order to mine the view-invariant patterns of the SSM feature, we propose a class-wise sparse coding method to encourage the sparse codes of the multi-view within-class videos to

lie close by. The closeness is measured by the within-class variance. Given the class labels of the training data, we try to reduce the within-class variance during the learning process. The within-class variance is measured by the Euclidean distance between the videos and their class center. The loss of the class-wise sparse coding model is defined as follows,

$$\mathbf{L}(\mathbf{X};\mathbf{D},\mathbf{C}) + \beta \sum_{s=1}^{K} \|\mathbf{C}^{(s)} - \bar{\mathbf{C}}^{s}\|_F^2 \tag{2.2}$$

The second term in Eq. 2.2 measures the within-class variance. This term enforces the multi-view within-class videos to have similar sparse codes. K is the number of action classes. $\mathbf{C}^{(s)}$ represents a video collection. s is the class index. Each column vector in $\mathbf{C}^{(s)}$ is the sparse code of the video which belong to action class s. Each column vector in $\bar{\mathbf{C}}^{s}$ is the mean of all the column vectors in $\mathbf{C}^{(s)}$. $\bar{\mathbf{C}}^{s}$ has the same size as $\mathbf{C}^{(s)}$. β is the weight of within-class variance penalty.

2.4 Sparse Code Filtering

2.4.1 Joint Action Learning

The input to our learning scheme is (1) the learned sparse codes for n_v videos, each represented as a n-dimension vector $\mathbf{c}^{(i)} \in \mathbb{R}^n, i = 1, 2, ..., n_v$. (2) the binary action label matrix for all the videos, which is represented as $\mathbf{Y}_{n_v \times n_a}$, n_a is the number of actions. The item $y^{(i,j)}$, is either 1 or 0, which denotes whether or not video i belongs to action class j.

We learn all action classifiers simultaneously in a multi-task learning setting, where each *task* represents one action. The output is the parameter matrix $\mathbf{\Theta}_{n \times n_a}$ whose column vector $\theta^{(j)}$ denotes the parameters of the classifier of action j. In our model, we employ logistic regression classifiers. Given the sparse code matrix and binary action label matrix ($\mathbf{C}_{n \times n_v}, \mathbf{Y}_{n_v \times n_a}$), the loss function is defined as:

$$\mathbf{L}(\mathbf{C}, \mathbf{Y}; \mathbf{\Theta}) = \sum_{i,j} log(1 + exp((1 - 2y^{(i,j)})(\theta^{(j)})^T \mathbf{c}^{(i)})) \tag{2.3}$$

Each action classifier has an tuple $\theta^{(j)}$ whose element $\theta_k^{(j)}$ corresponds to the *weight* of the sparse code which is tied to the k-th typical pattern in the dictionary.

2.4.2 Formulation of Sparse Code Filtering

Usually, the dictionary and classifiers are trained separately. Thus, there is no guarantee that the learned patterns in the dictionary can serve the classification task well. In order to mine the class-discriminative action patterns, we propose a sparse code filtering (SCF) scheme. In

our scheme, the prediction function is logistic function whose output denotes the probability that a video belongs to an action. Besides, the parameters are learned by minimizing both the dictionary reconstruction error and classification error. Thus, the dictionary and classifiers are optimized jointly. The learned sparse codes are expected to be view-invariant and class-discriminative. By integrating all the tasks, we can obtain the following loss function:

$$\mathbf{L}(\mathbf{X},\mathbf{Y};\mathbf{D},\mathbf{C},\mathbf{\Theta})=\mathbf{L}(\mathbf{X};\mathbf{D},\mathbf{C})+\gamma\mathbf{L}(\mathbf{C},\mathbf{Y};\mathbf{\Theta})+\beta\sum_{s=1}^{K}\|\mathbf{C}^{(s)}-\bar{\mathbf{C}}^{(s)}\|_F^2+\lambda\|\mathbf{\Theta}\|_F^2 \tag{2.4}$$

In Eq. 2.4, γ balances the dictionary reconstruction error and the classification error, the Frobenius norm of $\mathbf{\Theta}$ is employed to prevent overfitting. By minimizing the loss function, Eq. 2.4, a view-invariant and class-discriminative dictionary $\mathbf{D}$, and an action classification parameter matrix $\mathbf{\Theta}$ are learned jointly.

Optimization

The input of the SCF framework is video descriptor matrix and binary action label matrix: $[\mathbf{X},\mathbf{Y}]$. The outputs are the dictionary, sparse codes, and parameter matrix for the classifiers: $[\mathbf{D},\mathbf{C},\mathbf{\Theta}]$. We propose the following algorithm (Alg. 1) to solve the framework. When only one variable is left to optimize and the rest are fixed, the problem becomes convex. Thus, we optimize the variables alternatively by fixing the rest.

Algorithm 1: Solution Structure

Initialization: $\mathbf{D}\leftarrow\mathbf{D}_0$, $\mathbf{C}\leftarrow\mathbf{C}_0$, $\mathbf{\Theta}\leftarrow\mathbf{\Theta}_0$
repeat
  fix $\mathbf{D},\mathbf{\Theta}$, *update* $\mathbf{C}$:
  for $\mathbf{C}^{(s)}\in\mathbf{C}$ **do**
    $ratio\leftarrow 1$
    while $ratio > threshold$ **do**
      run **FISTA(modified)**
      update $ratio$
    end while
  end for
  fix $\mathbf{D},\mathbf{C}$, *update* $\mathbf{\Theta}$:
  $\mathbf{parallelgradientdescent}$
  fix $\mathbf{C},\mathbf{\Theta}$, *update* $\mathbf{D}$:
  $\mathbf{least-squaressolution}$
until converges

Initialization in Algorithm 1: we employ k-means clustering to find k centroids as the bases in dictionary $\mathbf{D}_0$. $\mathbf{\Theta}_0$ and $\mathbf{C}_0$ are set to $\mathbf{0}$.

The loop in Algorithm 1 consists of three parts:

1. Fix $\mathbf{C}$, $\mathbf{\Theta}$, Optimize $\mathbf{D}$. In Eq. (2.4), only the first term is related to $\mathbf{D}$, and it is a least square problem when the other parameters are fixed. By setting the derivative of Eq. (2.4) equal to $\mathbf{0}$ with respect to $\mathbf{D}$, we can obtain:

$$(\mathbf{DC}-\mathbf{X})\mathbf{C}^T{=}0 \Rightarrow \mathbf{D}{=}\mathbf{X}\mathbf{C}^T(\mathbf{C}\mathbf{C}^T)^{-1} \tag{2.5}$$

   Then we employ the following equation to update $\mathbf{D}$:

$$\mathbf{D} = \mathbf{X}\mathbf{C}^T(\mathbf{C}\mathbf{C}^T + \lambda\mathbf{I})^{-1} \tag{2.6}$$

   λ is a small constant and it guarantees that the matrix $\mathbf{C}\mathbf{C}^T{+}\lambda\mathbf{I}$ is invertible in case $\mathbf{C}\mathbf{C}^T$ is singular.

2. Fix $\mathbf{D}$, $\mathbf{C}$, Optimize $\mathbf{\Theta}$. When $\mathbf{D}$ & $\mathbf{C}$ are fixed, we employ the parallel gradient descent method to tackle the problem. Since $\theta^{(j)}$ are independent from each other, we optimize them in parallel. The updating formula is as follows:

$$\theta^{(j)} = \theta^{(j)} - \delta\frac{\partial}{\partial\theta^{(j)}}\mathbf{L}(\mathbf{X},\mathbf{Y};\mathbf{D},\mathbf{C},\mathbf{\Theta}) \tag{2.7}$$

3. Fix $\mathbf{D}$, $\mathbf{\Theta}$, Optimize $\mathbf{C}$. Beck and Teboulle [2009] proposed the Fast Iterative Soft-Thresholding Algorithm (FISTA) to solve the classical dictionary learning problem. A soft-threshold step is incorporated into FISTA to guarantee the sparseness of the solution. The complexity for the classical ISTA method is $O(1/k)$, in which k denotes the iteration times. FISTA converges in function values as $O(1/k^2)$, which is much faster. FISTA optimizes $\mathbf{c}^{(i)}{\in}\mathbf{C}$ independently. However, in our model, $\mathbf{c}^{(i)}$ and $\mathbf{c}^{(j)}$ within the same action class depend on each other. Thus, $\mathbf{c}^{(i)}$, $\mathbf{c}^{(j)}$ must be updated jointly until all of them converge. Thus, we decompose our objective function and modify the original FISTA algorithm to tackle the decomposed sub-objectives.

   In Eq. (2.4), the sparse code matrices with respect to different action classes are independent. Thus, when updating $\mathbf{C} = [\mathbf{C}^{(1)},...,\mathbf{C}^{(K)}]$, we decompose the objective function into K sub-objectives, shown as follows:

$$\min_{\mathbf{C}}\sum_{s=1}^{K}\mathbf{L}(\mathbf{C}^{(s)}) = \sum_{s=1}^{K}\min_{\mathbf{C}^{(s)}}\mathbf{L}(\mathbf{C}^{(s)}) \tag{2.8}$$

Thus, the original objective function is decomposed into K sub-objective functions with respect to each action class. The following shows the details of the deduction of decomposition of Eqn. 2.4. The first two terms in Eqn. 2.4 can be reformulated as follows:

$$\begin{cases} \mathbf{L}(\mathbf{X};\mathbf{D},\mathbf{C}) = \sum_{s=1}^{K} \left(\|\mathbf{D}\mathbf{C}^{(s)} - \mathbf{X}^{(s)}\|^2 + \alpha \|\mathbf{C}^{(s)}\|_1 \right) \\ \mathbf{L}(\mathbf{C}, \mathbf{Y}; \mathbf{\Theta}) = \sum_{s=1}^{K} \sum_{j=1}^{n_a} \log(1+\exp(1-2y^{(i,j)})(\theta^{(j)})^T \mathbf{C}^{(s)}) \end{cases} \tag{2.9}$$

Putting the transformed terms from Eq. (2.9) back into the loss function Eq. (2.4), we can obtain a new form of the objective function. Because $\mathbf{D}$ and $\mathbf{\Theta}$ are fixed, the term $\lambda\|\mathbf{\Theta}\|_F^2$ becomes a constant. By removing the constant term, we can obtain the loss function as Eq. (2.8) where

$$\begin{aligned} \mathbf{L}(\mathbf{C}^{(s)}) = & \|\mathbf{D}\mathbf{C}^{(s)} - \mathbf{X}^{(s)}\|_F^2 + \alpha\|\mathbf{C}^{(s)}\|_1 + \beta\|\mathbf{C}^{(s)} - \bar{\mathbf{C}}^s\|_F^2 \\ & + \gamma \sum_{j=1}^{n_a} \log(1+\exp(1 - 2y^{(i,j)})(\theta^{(j)})^T \mathbf{C}^{(s)}) \end{aligned} \tag{2.10}$$

The modified FISTA algorithm is applied to solve the sub-objective functions. The details of the modified FISTA algorithm is as follows:

In the classical dictionary learning model, the sparse codes of training data are independent from each other. Thus, each $\mathbf{c}$ can be optimized independently. However, our new sub-objective needs to optimize a group of training data jointly because these data have dependencies among each other as shown in Eq. (2.10). For training data $\mathbf{x}^{(i)} \in \mathbf{X}^{(s)}$ in the equation above, its sparse code $\mathbf{c}^{(i)}$ ($\mathbf{c}^{(i)} \in \mathbf{C}^{(s)}$) dependents on other $\mathbf{c}^{(k)}$ ($\mathbf{c}^{(k)} \in \mathbf{C}^{(s)}$). We modify the classical FISTA algorithm to optimize the sub-objectives jointly.

When update $\mathbf{C}^{(s)}$, instead of updating $\mathbf{c}^{(i)}$ independently, all $\mathbf{c}^{(i)} \in \mathbf{C}^{(s)}$ are updated simultaneously using the following form,

$$\mathbf{c}^{(i)} := \mathbf{c}^{(i)} - \delta \frac{\partial L}{\partial \mathbf{c}^{(i)}} \tag{2.11}$$

This updating procedure of $\mathbf{C}^{(s)}$ will repeat until it converges. Then we apply a soft-threshold step to set the entries in $\mathbf{C}^{(s)}$ whose absolute value is less than the threshold to 0. We repeat the process above until the whole algorithm converges.

Label Prediction

As shown in Fig.5.2, in the classical CF framework, when the features of a new movie are given, its ratings by different users can be predicted based on the movie features and the learned

feature preference weights. The basic underlying assumption of CF is that users will rate movies which share the similar features with similar scores Goldberg et al. [2001] as we assume that the preferences of the users remain the same. Similarly, each action class can be regarded as one user, and the action videos can be regarded as the movies. The label prediction for a new video $\mathbf{x}$ consists of two steps: sparse coding and probability calculation.

$$\mathbf{c}^* = \arg_{\mathbf{c}} \min \mathbf{L}(\mathbf{x}, \mathbf{D}; \mathbf{c}) \tag{2.12}$$

$$label = \arg_j \max 1/\big(1 + exp((-\theta^{(j)})^T \cdot \mathbf{c}^*)\big) \tag{2.13}$$

First, given the dictionary $\mathbf{D}$, and video descriptor $\mathbf{x}$ which is the SSM feature, the sparse code $\mathbf{c}^*$ of the new video is calculated by solving the classical sparse coding model as shown in Eq. (2.12). Then the probability that the new video belongs to action class j can be calculated. The action label is the one which maximizes the probability as shown in Eq. (2.13).

2.5 Experiments

2.5.1 Datasets

We evaluate our framework on three largest public *multi-view* action recognition datasets, as shown in Fig.2.3, which are the IXMAS dataset Weinland et al. [2006], the NIXMAS dataset, and the OIXMAS dataset Weinland et al. [2010] in which the actions are partially occluded. IXMAS dataset consists of 12 action classes, (*e.g., check watch, cross arms, scratch head, sit down, get up, turn around, walk, wave, punch, kick, point and pick up*). Each action is performed 3 times by 11 actors and is recorded by 5 cameras which observe the actions from 5 different viewpoints. The NIXMAS dataset is recorded with different actors, cameras, and viewpoints, and about 2/3 of the videos have objects which partially occlude the actors. Overall, it contains 1148 sequences.

2.5.2 Implementation Details

The sparse code filtering is based on SSM descriptors using HOG/HOF features to describe each individual frame. Each video is represented by a 500-dimension vector. Fig.2.2 shows an example from IXMAS dataset and the corresponding extracted SSM feature. In our experiments, the dictionary size is set to $[600, 700, ..., 1000]$, and all regularization parameters α, β, γ, λ are tuned from $[10^{-3}, 10^{-2}, ..., 10^{3}]$.

We employ two settings for the experiment, which are *multi-view* setting and *cross-view* setting. For the *multi-view* setting, we have access to the videos from all the viewpoints for

Figure 2.3: Multi-view action recognition datasets.

training, and use the standard experimental protocol described in Huang et al. [2012]: two-thirds and one-third split for training and testing. This experimental protocol is widely used for action recognition. For the *cross-view* setting, one camera view is missing in the training data and we train the model using the data from other four camera views. Then we perform prediction on the missing view.

2.5.3 Baselines

To evaluate the contribution of the class-wise sparse coding (CWSC), we put the raw features and the codes into two classification scheme: (1) standard radial basis kernel SVM Junejo et al. [2011] which learns each action classifiers separately, and (2) the multi-task learning approach Yan et al. [2014] which learns the action classifiers jointly. The codes and the classifiers are learned separately. We name the two baselines which take the codes as input as (3) CWSC+SVM, and (4) CWSC+MTL, and they are employed as baselines. Then through the comparison between (CWSC+MTL) and our SCF framework, we can observe the extra gain we obtained by training the class-wise dictionary and classifiers jointly. (5) We also choose some other action recognition baselines, such as Li and Zickler [2012], Yan et al. [2014], and Huang et al. [2012].

2.5.4 Results

Multi-view Action Recognition.

For the multi-view setting, we use the standard two-thirds and one-third split for training and testing. Table 2.1 shows the mean action recognition accuracy of all the cameras using different approaches.

Table 2.1: Multi-view action recognition accuracy of different approaches for 3 datasets.

Methods	IXMAS	OIXASM	NIXMAS
SVM Junejo et al. [2011]	0.6425	0.4809	0.5680
CWSC+SVM	0.6537	0.5235	0.6026
MTL Yan et al. [2014]	0.6883	0.5608	0.6163
CWSC+MTL	0.6889	0.6082	0.6228
Farhadi and Tabrizi [2008]	0.5810	-	-
Huang et al. [2012]	0.5730	-	-
Liu and Shah [2008]	0.7380	-	-
Reddy et al. [2009]	0.7260	-	-
Li and Zickler [2012]	0.8120	-	-
Baumann et al. [2016]	0.8055	-	-
Ashraf et al. [2014]	0.8140	-	-
SCF	**0.8594**	**0.7803**	**0.8083**

We observe that the baselines CWSC+SVM and CWSC+MTL outperform SVM and MTL with raw features respectively. This indicates that the class-wise sparse coding can help encode the view-invariant action patterns which preserve the label consistency. From Table 2.1, we can also observe that our method has the best performance. This is because our sparse code filtering scheme optimizes the classifiers and dictionary jointly, and it helps learn a class-wise label-discriminative dictionary. Fig.2.4 shows some qualitative results on IXMAS dataset for our proposed SCF framework and multi-task learning approach for multi-view action recognition.

Cross-view Action Recognition

Table 2.2, 2.3 and 2.4 show the performances of different approaches on IXMAS, OIXMAS, and NIXMAS dataset.

From Table 2.2, 2.3 and 2.4, we can observe that our framework achieves better performance

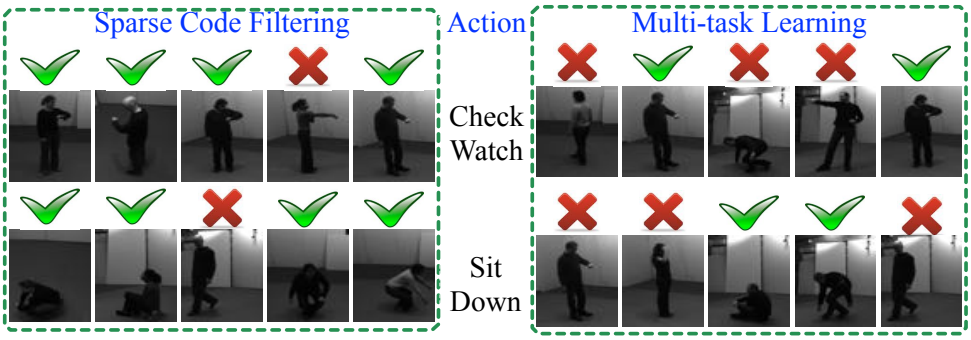


Figure 2.4: Qualitative results on IXMAS dataset.

Table 2.2: Cross-View action recognition performance on the IXMAS dataset

	Missing Viewpoints					
Methods	Cam 1	Cam 2	Cam 3	Cam 4	Cam 5	Avg
Junejo et al. [2011]	0.6663	0.6554	0.6500	0.6243	0.4963	0.6185
CWSC+SVM	0.6880	0.6577	0.6701	0.6187	0.5110	0.6291
Yan et al. [2014]	0.7554	0.7462	0.7710	0.6973	0.6332	0.7206
CWSC+MTL	0.7559	0.8257	0.8003	0.7759	0.6417	0.7599
SCF	**0.8285**	**0.8322**	**0.8053**	**0.7941**	**0.7384**	**0.7997**

Table 2.3: Cross-View action recognition performance on the OIXMAS dataset

	Missing Viewpoints					
Methods	Cam 1	Cam 2	Cam 3	Cam 4	Cam 5	Avg
Junejo et al. [2011]	0.5639	0.6250	0.5472	0.4677	0.4423	0.5292
CWSC+SVM	0.5688	0.6477	0.6001	0.5087	0.4511	0.5553
Yan et al. [2014]	0.5422	0.6540	0.5070	0.5171	0.4730	0.5387
CWSC+MTL	0.5535	0.6826	0.5366	0.5401	0.4867	0.5599
SCF	**0.6080**	**0.6980**	**0.6573**	**0.6957**	**0.5850**	**0.6512**

compared with other baselines which shows the effectiveness of our learned dictionary. It is also interesting to notice that the fifth camera always has low action recognition accuracy regardless of the classification methods. One reasonable explanation is that the fifth camera is placed on the ceiling, and the motion dynamics of different actions observed from this camera are visually

Table 2.4: Cross-View action recognition performance on the NIXMAS dataset

	Missing Viewpoints					
Methods	Cam 1	Cam 2	Cam 3	Cam 4	Cam 5	Avg
Junejo et al. [2011]	0.6410	0.6532	0.5912	0.5924	0.5322	0.6020
CWSC+SVM	0.6759	0.6951	0.6226	0.6387	0.5560	0.6377
Yan et al. [2014]	0.7170	0.6993	0.7542	0.6911	0.6792	0.7082
CWSC+MTL	0.7198	0.7391	0.7559	0.7176	0.6879	0.7240
SCF	**0.8080**	**0.7980**	**0.7573**	**0.7357**	**0.7050**	**0.7608**

similar with each other.

2.5.5 Parameter Tuning

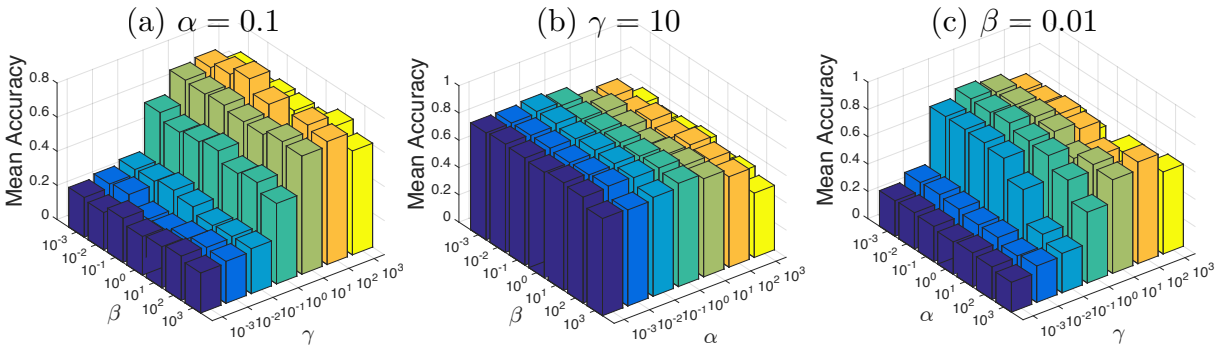


Figure 2.5: Sensitivity study of different regularization parameters on IXMAS dataset.

Fig.2.5 shows the sensitivity study of regularization parameters γ, α, β and λ. In our model, γ balances the dictionary learning loss and the classification loss, α balances the reconstruction error and the sparsity penalty, β provides the trade-off between the dictionary reconstruction loss and intra-class variance penalty, and λ is employed to prevent overfitting of the classifiers. The optimal classification performance can be obtained when dictionary size is set to 800. We observe that the performance changes little (within 0.0015) when we set λ to the different values. So we focus on the other 3 parameter. As shown in Fig.2.5(a), when γ is fixed, the mean accuracy varies subtly along the axis of β. However, when β is fixed, the mean accuracy changes dramatically along the axis of β. Thus, γ is more sensitive than β. Similarly, Fig.2.5(b) shows that α is more sensitive than β, and Fig.2.5(c) shows that γ is more sensitive than α. Thus, we obtain the importance of these parameters $\gamma{>}\alpha{>}\beta{>}\lambda$.

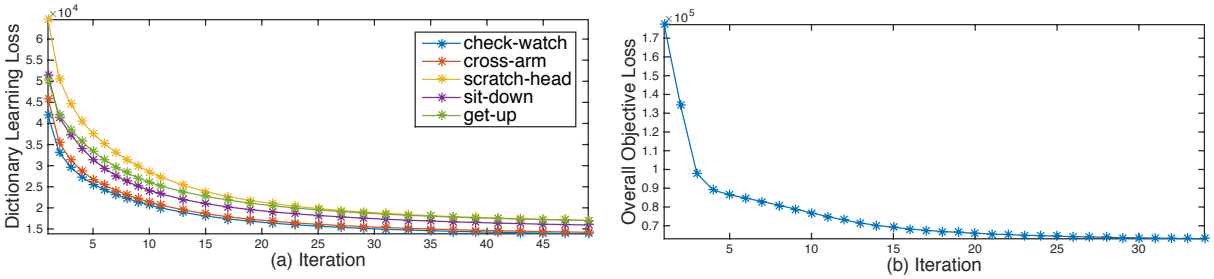


Figure 2.6: Convergence of the sparse code filtering algorithm on IXMAS dataset.

We also analyzes the convergence of our algorithm. Fig.2.6 plots the convergence curves of the objectives. Fig.2.6(b) shows that Alg. 1 converges in 30 iterations. Fig.2.6(a) plots the convergence curves when updating $\mathbf{C}^{(s)}$ for action classes. It shows that the class-wise dictionary learning converges very fast.

2.6 Conclusion

In this paper, we propose a novel sparse code filtering framework for multi-view action recognition. First, a class-wise dictionary is learned by encoding label information into the sparse coding process. We integrate class-wise sparse coding and classifier learning into a CF framework. Thus, the classifiers and dictionary are optimized jointly, and they can be adapted for each other. The extensive experimental results illustrate that our proposed method outperforms other important baselines for multi-view action recognition. In the future work, we will take the correlation between the classifiers into consideration. For example, we can suppress the urge of feature sharing between classifiers by adding a l_1 norm penalty to the classifier parameters.

Chapter 3

Face Alignment

3.1 Introduction

Face alignment methods trace their lineage from Active Shape Models Cootes and Taylor [1992, 1993] and Active Appearance Models (AAM) Cootes et al. [2001b], developed a couple of decades ago. These works first build a statistical shape and appearance models of the face, and during testing use numerical optimization techniques to find a set of parameters of the statistical model that could have generated the query face. Todays mainstream face alignment methods belong to Cascaded Regression Methods (CRM) group Cao et al. [2013]; Xiong and De La Torre [2013]; Yang and Patras [2013]; Tzimiropoulos [2015]; Zhu et al. [2015a]; Xiong and Torre [2015]. These methods operate in a cascaded fashion, *i.e.* starting from an initial shape and producing several shape increments that move the initial shape closer to the desired location. Shape increments are learned in a supervised manner during training stage. Formally CRMs operate in the following fashion:

$$\Delta \mathbf{S}_{t+1} = \mathbf{R}_t(\mathbf{F}_t(\mathbf{I}, \hat{\mathbf{S}}_t)), \tag{3.1}$$

$$\hat{\mathbf{S}}_{t+1} = \hat{\mathbf{S}}_t + \Delta \mathbf{S}_{t+1}, \tag{3.2}$$

where $\mathbf{I}$ denotes a 2D image, $\mathbf{F}_t(\mathbf{I}, \hat{\mathbf{S}}_t)$ represents the feature values extracted using the previous shape estimate $\hat{\mathbf{S}}_t$, $\Delta \mathbf{S}_{t+1}$ is a shape update produced by the t-th regressor $\mathbf{R}_t$ in the cascade. To initialize the pipeline the average face shape over all images in the training set $\bar{\mathbf{S}}$ is taken. The feature extraction function ($\mathbf{F}_t(\cdot, \cdot)$) and a set of regressors ($\mathbf{R}_t(\cdot)$) constitute the main ingredients of a CRM framework. The final outcome of the CRMs writes as:

$$\hat{\mathbf{S}}(T) = \bar{\mathbf{S}} + \sum_{t=1}^{T} \Delta \mathbf{S}_t, \tag{3.3}$$

where T is the total number of layers in the cascade. In order to frame a task at hand as a cascaded regression problem, one has to decide upon the feature extraction function ($\mathbf{F}_t(\cdot, \cdot)$),

as well as to select a proper regression function ($\mathbf{R}_t(\cdot)$). Various features have been explored by the community *e.g.* HoG Xiong and Torre [2015], SIFT Xiong and De La Torre [2013]; Tzimiropoulos [2015], pixel differences Tulyakov and Sebe [2015]; Kazemi and Josephine [2014]; Jeni et al. [2015], local binary features Ren et al. [2014], as well as different regression functions have been tried: linear regression Xiong and De La Torre [2013]; Jeni et al. [2015], random ferns Doll et al. [2010], regression trees Tulyakov and Sebe [2015]; Kazemi and Josephine [2014]. This brings to light two major limitations of the CRMs, that we are going to remove in this work: (i) manually designed features and (ii) relative independence of the regressors at the different layers in the cascade.

Hand-crafted computer vision features, such as HoG features for pedestrian detection Dalal and Triggs [2005], SIFT features for object recognition Lowe [1999], attribute detection Wang et al. [2016f, 2015] have played an important role in many application domains for a long time since they offer illumination, rotation and scaling invariance. These features, however, represent a generic image transformation that lacks any domain specific knowledge. Many works, have tackled this problem by *selecting* best features out of an overcomplete set Tulyakov and Sebe [2015]; Kazemi and Josephine [2014]; Ren et al. [2014]. However, this feature selection is suboptimal, since it is still performed on a generated set. Recently, it has been shown for object detection Krizhevsky et al. [2012], tracking Wang and Yeung [2013], image labeling Simonyan and Zisserman [2014] and other fields that features learned for a specific problem using deep convolutional neural networks show much better performance. Moreover, features learned for image classification often generalize well for different tasks, showing the ability of CNNs, such as AlexNet Krizhevsky et al. [2012], VGGNet Chatfield et al. [2014] and GoogleNet Szegedy et al. [2015], to learn a generic image representation.

The second limitation of the CRMs is the independence of the regressors at every level of the cascade. One can argue the regressor at time t is learned by using the output of the previous regressor at time $t-1$, with the final prediction given by Eq. 3.3. This however, affects only the feature computation (see Eq. 3.1), while the regressors themselves are learned independently. It has been shown in Xiong and De La Torre [2013] that a single regressor is not capable of arriving at the desired location in a single step. As shown in Eq. 3.3 the final prediction of the cascade $\hat{\mathbf{S}}(T)$ is a function of the number of layers in the cascade T. One can think of $\hat{\mathbf{S}}(T)$ as a sequence of measurements of some stochastic process. It has been recently shown that Recurrent Neural Network are extremely powerful in modeling the sequential inputs and outputs Auli et al. [2013]. In order to model long time-varying sequences, various RNN units have been proposed. In particular, long-short term memory cells and later Gated Recurrent Units have proven to be efficient in modeling time-varying processes and sequence-to-sequence learning Cho et al. [2014]. Additionally, it has been shown that using a CNN for feature extraction and an RNN for classification brings extra advantages Pinheiro and Collobert [2013]; Liang and Hu

[2015]; Lai et al. [2015].

This discussion naturally brings us to the main contribution of this work. We present a unified face alignment framework that features end-to-end learning starting from raw pixel values. We replace the manually hand-crafted features $\mathbf{F}_t(\cdot,\cdot)$ by learning a patch-based CNN. In contrast with boosted regression methods, where one has a sequence of regressors $\{\mathbf{R}_1(\cdot)$, $\mathbf{R}_2(\cdot), ..., \mathbf{R}_T(\cdot)\}$, our method learns a single recurrent module trained jointly with the CNN which can generate the regressor $\mathbf{R}(\cdot)$ recursively based on the input data and the memory of the recurrent module. We would like to highlight for the reader, that the parameters of both the CNN module and the RNN module are learned jointly. Additionally, we show that our model is capable of generalizing beyond the learned number of recurrent iterations, being able to automatically decide when to stop iterating. The experimental evaluation we detail in Section 3.4 proves that learning a task-specific end-to-end model brings higher accuracy than that of the available state-of-the-art.

3.2 Related work

In this section we review relevant works in face alignment as well as discuss recent advances in the neural-network learning important to formulate our Recurrent Convolutional Face Alignment (RFCA) method.

3.2.1 Face alignment

According to the widely accepted classification, methods for face alignment can be grouped into three broad categories Wang et al. [2014]: Active Appearance Models (AAM), Constrained Local Models (CLM) Saragih et al. [2011]; Baltrusaitis et al. [2012]; Yu et al. [2013], and Cascaded Regression Methods (CRM). Initial works on face alignment such as ASMs Cootes and Taylor [1992, 1993] and AAMs Cootes et al. [2001b,a], build a parametric statistical shape and appearance models from a set of training faces. These methods show reasonable accuracy when the testing image is close to the training distribution. However, they fail to generalize to an unseen subject Gross et al. [2005]. Although such methods still attract the attention of researchers Tzimiropoulos and Pantic [2013]; Fanelli et al. [2013], the more recent Cascaded Regression Methods have shown higher accuracy at impressive frame rates Ren et al. [2014]; Kazemi and Josephine [2014]. In the following we will mostly detail this latter group of works.

Initially CRMs were introduced in the medical image processing community for anatomic structure prediction Zhou and Comaniciu [2007]. Since then they have been extensively exploited by the computer vision community with many seminal works proposed in the literature. Currently this avenue of research represents the mainstream direction of the deformable shape fitting. In Doll et al. [2010] a method for cascaded pose regression was introduced. The authors

used a *pose-indexed features* and learned a sequence of weak-regressors (random ferns in their case) to regress a deformable shape from an image. In order to compute pose-indexed features one has to provide the current belief regarding the shape. This naturally brings some form of pose invariance to the framework. Later, these ideas were extended to regress the whole face shape Cao [2012]. Importantly, it was shown that regressing the whole shape imposes the result to lie in the space constructed by all the training images.

The supervised descent method (SDM) Xiong and De La Torre [2013] further extends the cascaded framework to generic non-linear optimization problems: face alignment, template tracking and camera calibration. SDM learns a sequence of descent directions that applied sequentially solve the optimization problem. The authors replace the feature extraction part with SIFT Lowe [2004] and achieve impressive results by using linear regressors in the layers of the cascade. A downside of SDM, is its inability to generalize well to non-frontal poses, requiring to train separate regressors depending on the detected head pose. This constraint is relaxed in Xiong and Torre [2015] by introducing a global SDM to automatically learn several descent maps at every level of the cascade to handle complex cost-functions. These ideas were extended in Tzimiropoulos [2015], where the authors learn both the Jacobian and the Hessian matrices, in a manner inspired by the Gauss-Newton optimization method. Similarly to the original SDM, the authors use hand-crafted SIFT features extracted around the keypoints locations. SDM-based methods have become popular in various applications of face analysis Tulyakov et al. [2016b] and are used in several commercially available face alignment systems[1]. A different strategy for feature extraction is presented in Tulyakov and Sebe [2015]; Ren et al. [2014]; Kazemi and Josephine [2014]. Instead of employing hand-crafted features (*e.g.*, HoG, SIFT), they perform feature selection using a framework of regression trees. Alleviating the need to compute hand-crafted features, these works reach impressive processing speed.

Multiple CRM-based 3D methods have been proposed. In Cao et al. [2014a], an extension of Cao [2012] is introduced to fit a 2D-3D parametric shape model. Similar ideas were explored in Jourabloo and Liu [2015], where a cascaded coupled regressor is introduced to obtain the camera projection matrix and the 3D landmarks of the face. The work in Tulyakov and Sebe [2015] proposes to include the third dimension directly into the learning pipeline. They used a large generated set of training faces and showed that considering a face shape as a 3D object gives better results. An interesting work in Jeni et al. [2015] uses binary features to track a large number of points on the face, with subsequent 3D deformable model fitting to obtain a 3D mesh of the face. Notably, this work shows impressive frame rates for the whole pipeline as well as the tracking accuracy comparable to purely 3D methods Tulyakov et al. [2014].

From a higher perspective the aforementioned methods have two independent steps: (i) fea-

[1] http://www.humansensing.cs.cmu.edu/intraface/index.php
http://www.zface.org/

ture extraction and (ii) applying a sequence of regressors. Typically the first step is performed by using some hand-crafted features such as SIFT Xiong and De La Torre [2013]; Tzimiropoulos [2015], HOG Xiong and Torre [2015]. Some form of feature learning is employed in Ren et al. [2014]; Tulyakov and Sebe [2015]; Kazemi and Josephine [2014], while the levels of the cascade in the second step still remain independent. This requires a researcher to use a trial-and-fail approach in selecting which features and which regressors work the best.

In contrast, the method presented in our study is end-to-end. By learning convolutional filters, RCFA does not require manual supervision in defining feature extraction functions. Additionally, our method replaces a cascade of independent regressors by a single recurrent model, where all iterations are leaned jointly. This formulation merges the two steps of the typical CRM pipeline into a single unified framework, simultaneously trained using the available data.

3.2.2 Recurrent and convolutional neural networks

Recurrent Neural Networks (RNN) have become increasingly popular to learn complex dynamic systems, because of their impressive capability to recurrently operate with sequential input. During each recurrence of the traditional RNN Schuster and Paliwal [1997], an input signal is mapped to the hidden state, which is passed forward to the next recurrence. This way, the information of the previous states is memorized and persists during the whole process. Therefore, RNNs have proven to have an advantage in modeling sequences with long-term dependencies. During the last decade, we have seen a lot of success in applying RNNs to various application domains, such as generating text description of videos Venugopalan et al. [2015], image caption generation Karpathy and Fei-Fei [2015], face aging Wang et al. [2016c], machine translation Auli et al. [2013] and speech recognition Graves et al. [2013].

Given the success of RNNs, a lot of RNN variants have been explored, such as the Long Short Term Memory (LSTM) Hochreiter and Schmidhuber [1997] networks, Gated Recurrent Unit (GRU) Cho et al. [2014], and Clockwork RNN Koutnik et al. [2014]. All these architectures consist of a chain of repeated modules, where each module contains several gates, controlling the information flow in the network and states, memorizing necessary information for future recurrences. Although the combination of gates/states varies depending on the selected architectures, each subsequent iteration is performed similarly, by processing a new input using the memory of the current state. These architectures show varying performances for different tasks. In Jozefowicz et al. [2015] it was shown that, in general, GRU-based models feature superior performance compared with other architectures.

Convolutional Neural Networks (CNN) have recently demonstrated notable success in multiple tasks, such as image classification Krizhevsky et al. [2012], super-resolution Dong et al. [2014], as well as image segmentation Liang et al. [2015]. One of the main advantages of CNNs, is that they do not require human supervision to design feature transformation. Their

feature representations have shown to provide significantly higher performance, compared to commonly adopted hard-crafted features, in numerous application domains. Thus, it is very promising to combine the RNN architecture together with the CNN architecture into a hybrid architecture. This hybrid architecture has been successfully applied to many tasks, such as scene labeling Pinheiro and Collobert [2013], object recognition Liang and Hu [2015], and text classification Lai et al. [2015].

3.3 Method

The overview of the proposed Recurrent Convolutional Face Alignment method is given in Fig. 3.1. The framework mainly consists of two parts, the *recurrent module* and the *convolutional module*. During each recurrent iteration t the current shape estimate $\mathbf{h}_t$ is imposed onto the image and the convolutional neural network is applied to the patches extracted around the points of the shape. The output of the last layer of the CNN is passed to the RNN as an input. During the first iteration, the average shape of all the images in the training set is set to the initial shape estimate: $\widehat{\mathbf{h}}_0 = \bar{\mathbf{S}}$.

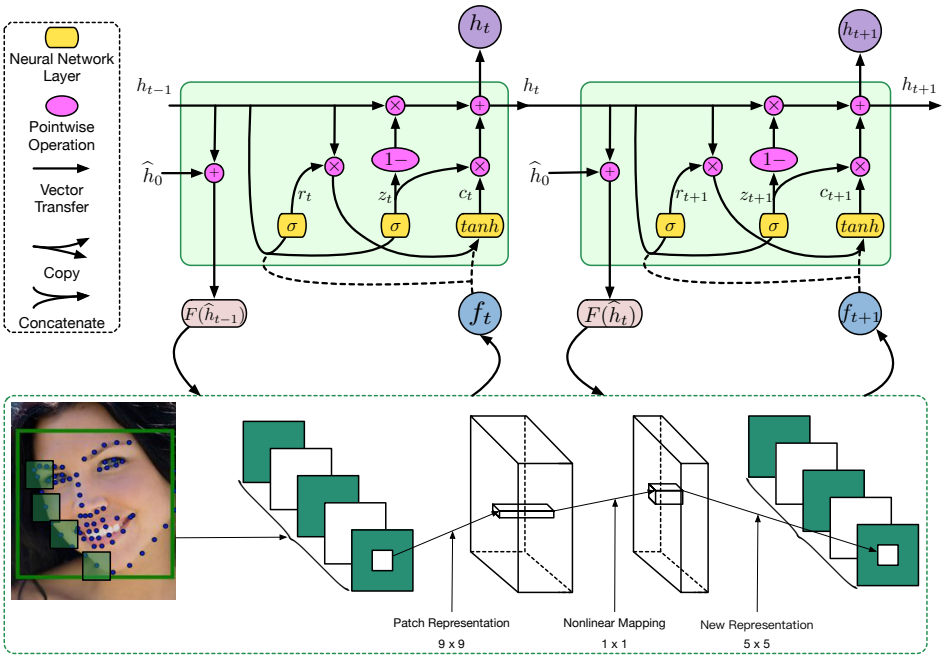


Figure 3.1: The overview of the proposed approach. Top: the RNN with gated recurrent units unrolled in time. Bottom: the CNN architecture used for feature extraction. Note that feature extraction is performed at every recurrent iteration.

3.3.1 Recurrent module

In the current study we use an RNN with GRU module for its simplicity and superior performance as compared to other RNN types Jozefowicz et al. [2015]. The structure of two recurrent iterations is given in the top row of Fig. 3.1. A GRU contains two gates and one state. The gates are the *reset* gate and the *update* gate. The *hidden* state $\mathbf{h}_t$ represents the relative movement or increment of the landmark positions after the adjustment in t-th iteration. Then the predicted position after t iterations is $\widehat{\mathbf{h}}_t = \widehat{\mathbf{h}}_0 + \mathbf{h}_t$.

The feature extraction function $f(t) = \mathbf{F}(\mathbf{I}, \widehat{\mathbf{h}}_t)$ is performed using a super resolution convolutional neural network (SRCNN), described in Section 3.3.

The *reset* gate $\mathbf{r}_t$ controls whether the adjustment from the previous recurrence should be ignored, *i.e.* if $\mathbf{r}_t$ is close to 0, the information of the previous adjustment operation will be forced to be discarded. Then the unit will focus on its current features without referring to the previous operation. To sum up, the reset gate allows the unit to remember or drop the adjustment operation from the previous operation.

The *update* gate $\mathbf{z}_t$ has two functions. The first one is to control what to forget from the previous operation which is implemented by the term $\mathbf{z}_t$, and the second one is to control the acceptance of the new input operation which is implemented by the term $1 - \mathbf{z}_t$. If $\mathbf{z}_t$ is set to 0, $1 - \mathbf{z}_t$ will be 1. This means that all the information from the previous operation will be kept and the new input will be totally discarded. Thus, the new adjustment operation will be exactly the same as the previous operation. However, if $\mathbf{z}_t$ is set to 1, $1 - \mathbf{z}_t$ will be 0, and the next operation will be based only on the new input operation without referring to the previous adjustment operation.

The described process is schematically presented in Fig. 3.1, where a single recurrent iteration is governed by the following equations:

$$\begin{aligned}
\mathbf{z}_t &= \sigma(\mathbf{W}_{zh}\mathbf{h}_{t-1} + \mathbf{W}_{zf}f_t + \mathbf{b}_z) \\
\mathbf{r}_t &= \sigma(\mathbf{W}_{rh}\mathbf{h}_{t-1} + \mathbf{W}_{rf}f_t + \mathbf{b}_r) \\
\mathbf{c}_t &= tanh(\mathbf{W}_{ch}\mathbf{r}_t \odot \mathbf{h}_{t-1} + \mathbf{W}_{cf}f_t + \mathbf{b}_c) \\
\mathbf{h}_t &= (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \mathbf{c}_t
\end{aligned} \tag{3.4}$$

where $\odot$ represents element-wise multiplication, and $\mathbf{c}_t$ is the new increment candidate created by the $tanh$ layer that could be added to the current shape increment using the following rule:

$$\mathbf{h}_t = (1-\mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \mathbf{c}_t. \tag{3.5}$$

If the reset gate is always activated, the system will have only the short-term memory, since the calculation of the new increment candidate ignores the previous increments and focuses on the current input only. If the update gate is not activated, the system can have the long-term memory and the previous increments will be memorized.

Within this framework, the RNN acts as a refinement process which tries to find the optimal shape increment by *gradually* changing the previous shape. We use T recurrent steps to train RCFA. In order for the RNN to focus on the later iterations we define a series of weights $\mathbf{w}=[w_1, w_2, ...w_T]$ each one for a single recurrent iteration. These weights increase monotonically, therefore forcing the recurrent network to adjust the shape slowly and penalizing the model more for the error during the later recurrent steps. Formally, the loss writes as:

$$J = \sum_{i=1}^{n} \sum_{t=1}^{T} w_t \|(\widehat{\mathbf{h}}_0 + \mathbf{h}_t^i) - \mathbf{h}_*^i\|_F^2, \tag{3.6}$$

where $\widehat{\mathbf{h}}_0$ is the initial shape estimate, *i.e.* the average shape, $\mathbf{h}_t^i$ is the predicted shape increment after t iterations, $\mathbf{h}_*^i$ is the target shape, the superscript i defines the ith image in the mini-batch of n images. The final shape after t steps is obtained as $\widehat{\mathbf{h}}_0 + \mathbf{h}_t^i$. During training, for each face image $\mathbf{I}^i$, the initial shape $\widehat{\mathbf{h}}_0$ is sampled several times by adding noise to the mean shape.

3.3.2 Convolutional module

We employ the super resolution convolutional neural network (SRCNN) for feature extraction Dong et al. [2014]. We apply the SRCNN to the pixel values around the landmarks position Fig. 3.1. We denote the patch around a landmark location as $\mathbf{Y}$, and use it as an input for the SRCNN. The SRCNN consists of three convolution layers, formulated as the following operations:

$$\begin{aligned} F_1(\mathbf{Y}) &= \max(0, \mathbf{W}_1 * \mathbf{Y} + \mathbf{B}_1) \\ F_2(\mathbf{Y}) &= \max(0, \mathbf{W}_2 * F_1(\mathbf{Y}) + \mathbf{B}_2) \\ F_3(\mathbf{Y}) &= \mathbf{W}_3 * F_2(\mathbf{Y}) + \mathbf{B}_3 \end{aligned} \tag{3.7}$$

where $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{W}_3$ and $\mathbf{b}_1$, $\mathbf{b}_2$, $\mathbf{b}_3$ represent the filters and biases respectively. The Rectified Linear Unit (ReLU) is employed as the activation function for the first two convolution layers. The dimensions of $\mathbf{W}_1$ are set to $c \times f_1 \times f_1 \times n_1 = [1 \times 9 \times 9 \times 64]$, where c is the number of channels of the input image, f_1 is the filter size, and n_1 is the number of filters which also corresponds to the number of feature maps. $\mathbf{W}_2$ is of the size $n_1 \times 1 \times 1 \times n_2 = [64 \times 1 \times 1 \times 32]$ and $\mathbf{W}_3$ has the size of $n_2 \times f_3 \times f_3 \times c = [32 \times 5 \times 5 \times 3]$. The first layer can be regarded as PCA where each filter works as a basis and projects the input $\mathbf{Y}$ to a high-dimension vector. The second layer has the filter size of 1×1, and this layer can be understood as a non-linear mapping operation which maps an $n-1$ dimensional vector to a n_2 dimensional vector. Originally, the last layer in the SRCNN works as an averaging filter which projects the n_2 dimensional vector to a high-resolution patch, and take the average of the overlapping high-resolution patches. However, instead of projecting the n_2 dimensional vector to a high-resolution patch, the last

layer in our network will project the n_2 dimensional vector to a feature space which can is then passed to the recurrent module.

3.3.3 Supervised descent method as GRU

In this section we show that the proposed RCFA method is a generalization of the widely adopted Supervised Descent Method Xiong and De La Torre [2013]. Given a set of images $[\mathbf{I}^1, \mathbf{I}^2, ...\mathbf{I}^i, ..., \mathbf{I}^n]$, $\mathbf{h}^i$ denotes the positions of the landmarks in image $\mathbf{I}^i$. $\mathbf{F}$ is a feature extraction function, and $\mathbf{F}(\mathbf{h}^i)$ represent the extracted features. Let $\mathbf{y}_*^i{=}\mathbf{F}(\mathbf{h}_*^i)$ be the ground-truth features extracted at the manually labeled landmark positions $\mathbf{h}_*^i$. Then we have the following objective function for face alignment with respect to image $\mathbf{I}^i$,

$$\min \|\mathbf{F}(\mathbf{h}^i) - \mathbf{y}_*^i\|_2^2. \tag{3.8}$$

SDM applies the gradient descent rule to Eq. 3.8, and yields the following discrete update equation:

$$\begin{aligned} \mathbf{h}_t^i &= \mathbf{h}_{t-1}^i - \mathbf{R}_{t-1}(\mathbf{F}(\mathbf{h}_{t-1}^i) - \mathbf{y}_*^i) \\ &= \mathbf{h}_{t-1}^i - \mathbf{R}_{t-1}\mathbf{F}(\mathbf{h}_{t-1}^i) + \mathbf{R}_{t-1}\mathbf{y}_*^i, \end{aligned} \tag{3.9}$$

where $\mathbf{R}_{t-1}{=}\alpha\mathbf{F}'(\mathbf{x}_{t-1}^i)$, and $\mathbf{R}_{t-1}$ is regarded as a regressor. Thus, instead of calculating the derivatives, the SDM learns a descend direction from the available training data.

However, Eq. 3.9 has an inconsistency problem, *i.e.* $\mathbf{y}_*^i$ is only available in the training phase and it is unknown in the testing phase. Therefore, Eq. 3.9 could not be used to calculate the position of the landmarks. To solve this inconsistency problem, $\mathbf{y}_*^i$ is replaced by $\overline{\mathbf{y}}_* = (\sum_i \mathbf{y}_*^i)/n$. By defining $\mathbf{b}_{t-1} = \mathbf{R}_{t-1}\overline{\mathbf{y}}_*$ we obtain the new update equation:

$$\mathbf{h}_t^i = \mathbf{h}_{t-1}^i - \mathbf{R}_{t-1}\mathbf{F}(\mathbf{h}_{t-1}^i) + \mathbf{b}_{t-1}, \tag{3.10}$$

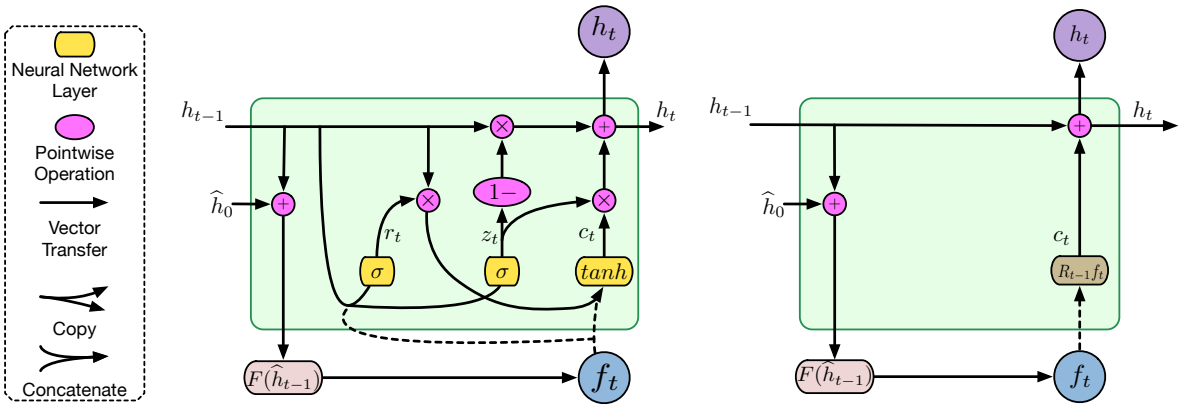
which solves the inconsistency problem. During the training phase, $\mathbf{h}_t^i$ is set to $\mathbf{h}_*^i$ as our goal is to make $\mathbf{h}_t^i$ equal to the target $\mathbf{h}_*^i$. The loss is defined as:

$$\sum_i \|\mathbf{h}_*^i - \mathbf{h}_{t-1}^i + \mathbf{R}_{t-1}\mathbf{F}(\mathbf{h}_{t-1}^i) - \mathbf{b}_{t-1}\|^2 \tag{3.11}$$

where $\mathbf{h}_0^i$ is obtained using Monte Carlo integration.

Thus, Eq. 3.11 can be considered as a special case of Eq. 3.6, making the SDM a special case of our GRU network. As shown in Fig. 3.2 (b), the traditional linear regressor is equivalent to GRU if the *update* gate and *reset* gate are removed. Finally, if we replace the $tanh$ layer with the regressor $\mathbf{R}$, we obtain the formula for the shape increment $\mathbf{h}_t$ for the image $\mathbf{I}^i$, as follows:

$$\mathbf{h}_t^i = \mathbf{h}_{t-1}^i - \mathbf{R}_{t-1}\mathbf{F}(\mathbf{h}_{t-1}^i) \tag{3.12}$$



(a) Recurrent Neural Network (b) Traditional Regressor

Figure 3.2: Differences in the architecture of the proposed recurrent regressor (a) compared to the traditional regressor (b).

Eq. 3.12 is a recurrent version of Eq. 3.10 except for the term $\mathbf{b}_{t-1}$ which can be implemented by expanding the feature space by several columns set to 1.

As shown in Fig. 3.2, the traditional regressors at different time steps ($\mathbf{R}_1$, $\mathbf{R}_2$, ...) are trained independently, relying only on the input features while totally lacking memory regarding previous states, since as it is shown in Eq. 3.11, every step has a separate loss function. In contrast, for our model the overall loss over all the recurrent steps is defined and learned jointly by summing the steps up (Eq. 3.6). Another way of thinking about our recurrent module, is to treat it not as a regressor, but rather as a way of generating unique regressors at every recurrent step with respect to the memory and the input features.

3.4 Experiments

Datasets We evaluate the performance of our algorithm using the widely adopted 300-W dataset Sagonas et al. [2013]. This dataset is a combination of several in-the-wild datasets, including AFW Zhu and Ramanan [2012], LFPW Belhumeur et al. [2013], HELEN Le et al. [2012] and XM2VTS Messer et al. [1999], that are annotated with 68-point markup in a consistent manner. Similarly to previous works Zhu et al. [2015a]; Ren et al. [2014], for training the model we use the training samples from LFPW, HELEN and the whole AFW dataset, which makes 3148 images in total. Testing is performed on three different sets of images: (i) the *common* set includes the testing images from the LPFW and HELEN, (ii) the *challenging* set includes recently released 135 images also known as the IBUG set, and (iii) the *full* set is a combination of the first two. We do not report the results for the original annotations for HELEN and LFPW, since the accuracy of the state-of-the-art methods has saturated.

Evaluation Metrics To evaluate the performance of our method, we follow the widely adopted evaluation metric Zhu et al. [2015a]; Tulyakov and Sebe [2015]; Ren et al. [2014], which is the **average error** of the point-to-point Euclidean distance, normalized by the distance between the outer corners of the eyes. This metric has been adopted for the 300-W challenge.

3.4.1 Implementation

For the CNN module, we follow the settings of the SRCNN framework in Dong et al. [2014]. This module will extract features for all the image patches. After obtaining the features for each image patch, we concatenate the feature vectors of the 68 landmarks and feed the features to the RNN module. For the RNN module, we set the total number of recurrent iterations T to be 5. The weights in Eq. 3.6 are set to powers of 10: $\mathbf{w} = [10^{-2}, 10^{-1}, ..., 10^{2}]$. Powers of 2 and 5 showed slightly inferior performance to the powers of 10, while equal weights $[1, 1, 1, 1, 1]$ showed the worst performance.

To augment the size of the training data, we duplicate the images by adding the mirrored examples, and we also replicate the training data 3 times by adding noise to the bounding boxes. In the training phase, the batch size is set to 204 images. The learning rate is set to 0.01. The decay rate is set to 0.5, and the learning rate will be decayed after every 10 epochs and the training process is terminated after 200 epochs. After we obtain the model, we generate another three replicates in the same manner and fine tune the network with the new replicates for another 200 epochs.

3.4.2 Understanding when to stop iterating

One of the further advantages of our RCFA is that the model can be easily extended beyond the learned number of recurrent iterations without the need of retraining the whole pipeline. Importantly, there is no upper bound on the number of recurrent iterations one can perform. This, however, requires devising a strategy to stop iterating. During training the RNN performed 5 recurrent iterations. Intuitively, the model should require less iterations for a simple image, while difficult examples may need additional recurrences. In order to define a stopping strategy, we show the relationship between the average error and the recurrent steps as shown in Fig. 3.3.

As it is seen from the left graph in Fig. 3.3 for easy images from the common set additional iterations are redundant and do more harm, while the hard cases from the challenging set benefit from iterating further (see Fig. 3.3, right). Typically, for hard cases the error still continues decreasing when the number of iterations is more than 15, while for the easy ones it remains stable between 5-th and 9-th iterations and then goes up. This suggests a simple and efficient stopping criteria that was used to generate the results for the RCFA adaptive in the Table 3.1. If the difference between the previous landmark positions and the current landmark positions is

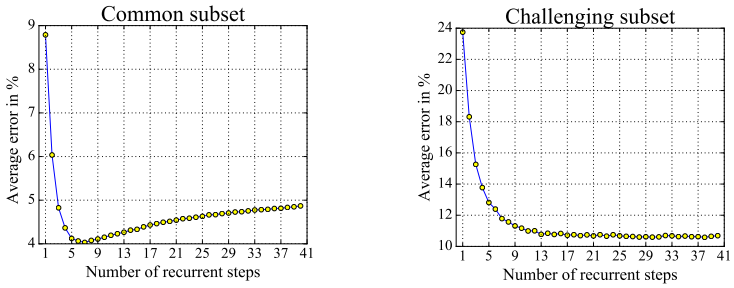


Figure 3.3: Average error vs the number of recurrent iterations

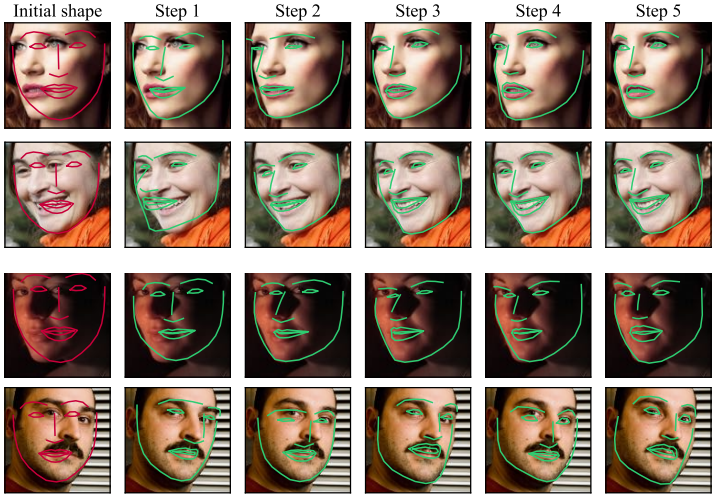


Figure 3.4: Landmark localization for 5 recurrent steps. The top two rows show examples, for which 5 iterations is sufficient, while the examples in the last two rows require additional iterations.

smaller than a threshold, we stop iterating. To set the threshold value, we take the average difference between the 4-th and 5-th recurrence of all the images in the training data. This simple stopping strategy allows our model to automatically decide whether any additional iterations are necessary.

Fig. 3.4 shows several qualitative examples of different number of recurrent iterations required for different testing examples. The first two rows show that 4 steps are not sufficient to localize the landmarks, as one can easily see that the landmarks on the jawline in the first row do not fit perfectly until the fifth recurrence. A similar observation can be made for the subject shown in the second row. The last two rows show cases, when even 5 iterations are not sufficient for the method to converge, due to difficult illumination conditions and extreme head poses. This further supports the importance of the adopted stopping strategy.

Table 3.1: Experimental results obtained on the three subsets of the 300-W dataset.

Method	Common	Challenging	Full
Zhu *et al.* Zhu and Ramanan [2012]	8.22	18.33	10.20
DFMF Asthana et al. [2013]	6.65	19.79	9.22
ESR Cao et al. [2014b]	5.28	17.00	7.58
RCPR Burgos-Artizzu et al. [2013]	6.18	17.26	8.35
SDM Xiong and De La Torre [2013]	5.57	15.40	7.50
Smith *et al.* Smith et al. [2014]	-	13.30	-
Zhao *et al.* Zhao et al. [2014]	-	-	6.31
GN-DPM Tzimiropoulos and Pantic [2014]	5.78	-	-
CFAN Zhang et al. [2014]	5.50	-	-
ERT Kazemi and Josephine [2014]	-	-	6.40
LBF Ren et al. [2014]	4.95	11.98	6.32
LBF fast Ren et al. [2014]	5.38	15.50	7.37
CFSS Zhu et al. [2015a]	4.73	9.98	5.76
CFSS Practical Zhu et al. [2015a]	4.79	10.92	5.99
RCFA 5 iterations	4.08	12.81	5.81
RCFA 10 iterations	4.13	11.14	5.51
RCFA adaptive	**4.03**	**9.85**	**5.32**

3.4.3 Experimental Results

We report evaluation results on the three subsets of the 300-W dataset in Table 3.1. It compares three different result of the same RCFA model against best performing state-of-the-art methods. The reported RCFA results are obtained using 5, 10 recurrent iterations and the proposed stopping strategy. We would like to highlight, that due to the end-to-end structure, our model shows better performance than the up-to-date face alignment methods regardless of the number of iterations for the common set and the full set. Notably, when the proposed stopping strategy is used, the proposed method outperforms other works by a large margin for all three testing sets.

Interestingly, RCFA outperforms CFSS Zhu et al. [2015a] by a margin of 16% on the common subset, while showing a little bit lower performance gain for the challenging set and the full set (10% and 9% correspondingly). There are mainly two reasons for this. Firstly, the commonly accepted evaluation metric is severely affected by a small portion of hard examples.

Secondly, these hard examples are not evenly presented in the 300-W training/testing sets. The dataset is rather biased towards having less extreme head poses, facial expressions and poor illumination conditions.

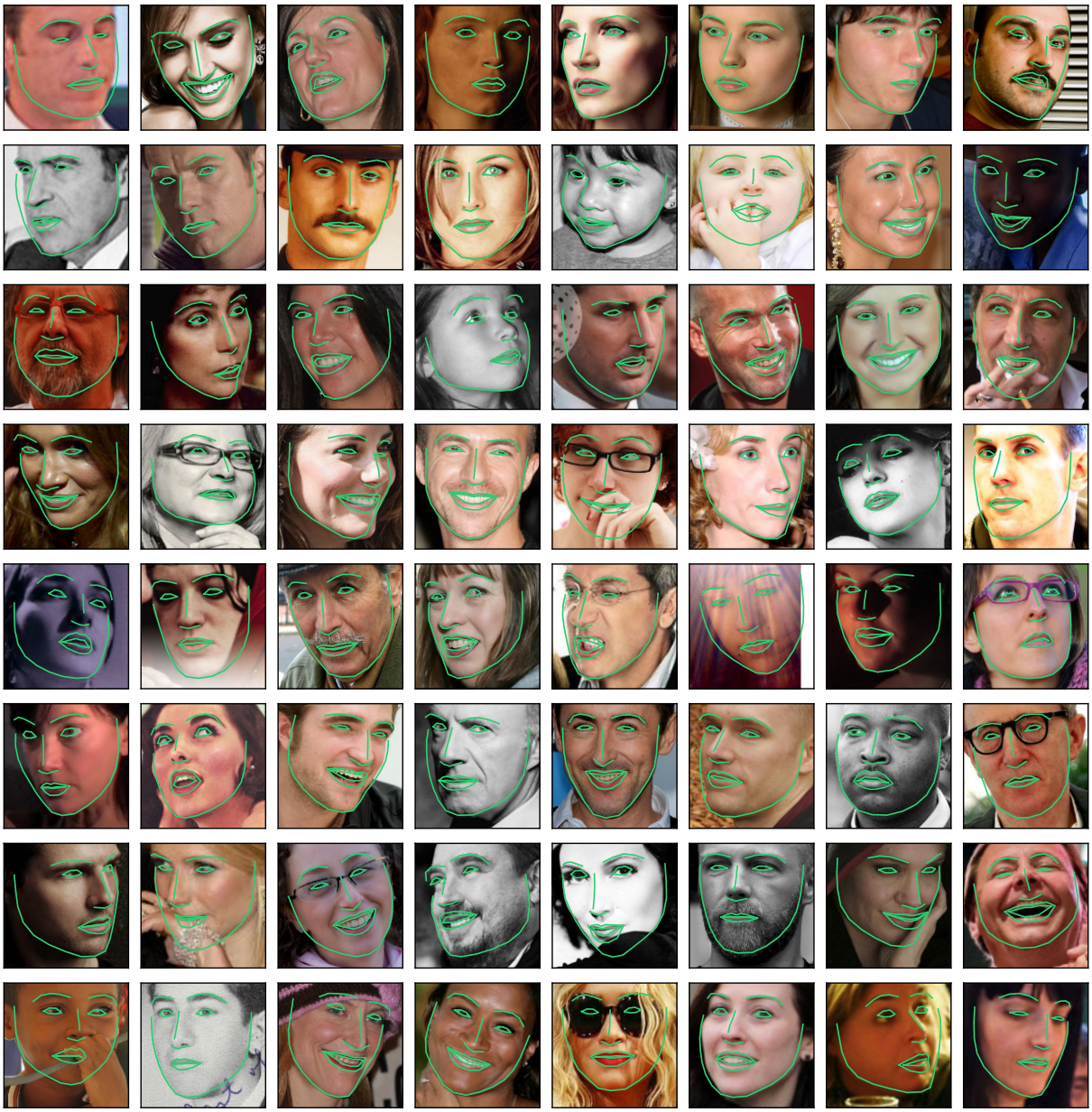


Figure 3.5: Selected qualitative examples taken from the full set of the 300-W dataset.

Fig. 3.5 shows the qualitative results for the images taken from the full set. Clearly, due to end-to-end learning our framework handles even challenging face images, such as facial expressions, extreme head poses, difficult lighting conditions. It is also very interesting to observe that RCFA can handle faces with severe occlusions, including sun-glasses, hands and

hats. The reason why our framework can work well for these images is because our RNN network can not only learn the dependencies between each regressor, it also learns the location dependencies between the landmarks. Thus, even though parts of the face is occluded, our framework can still predict the location of the occluded landmarks based on other landmarks.

In the current implementation, a single forward pass through the pipeline takes around 10ms on average on Tesla K40, making it possible to apply the proposed model for real-time video processing at 100 frames per second. We would like to note, that no specific performance optimizations were used, therefore, we believe the running time can be decreased dramatically.

3.5 Conclusions

In this paper, we reformulate the classical cascaded regression face alignment problem as a recurrent process, alleviating the two major limitations of the CRMs. The proposed recurrent framework features end-to-end learning, starting from the raw pixel data, removing the previously used hand-crafted features. Replacing a standard cascade of independently learned shape regressors by a single recurrent regressor brings further advantage of iterating beyond the learned limit, making it possible to automatically decide when to stop.

The proposed RFCA method has room for further improvements. In our experiments an average shape is used to initialize the pipeline, while it has been shown that selecting a proper starting shape brings extra benefits Zhu et al. [2015a]. Additionally, more rigorous data augmentation can alleviate the bias of the training set and can make the data more uniform. Furthermore, we believe similar recurrent-convolutional shape regression models can be employed to various other tasks such as action recognition Wang et al. [2016e] and human pose estimation.

Chapter 4

Face Aging

4.1 Introduction

Face aging, also known as age progression, is attracting more and more research interest. It has wide applications in various domains including cross-age face verification Park et al. [2010] and finding lost children. In recent years, face aging has witnessed many breakthroughs and a number of face aging models have been proposed Fu et al. [2010]; Suo et al. [2012]; Kemelmacher-Shlizerman et al. [2014]; Tiddeman et al. [2001]; Suo et al. [2010]; Tazoe et al. [2012]. However, it remains a very challenging task in practice for several reasons. First, faces may have various expressions and lighting conditions, which pose great challenges to modeling the aging patterns. Besides, the training data are usually very limited as face images for the same person usually cover a very narrow range of ages. Moreover, the face aging process depends on both the environment and genes which are hard to model.

Generally, face aging follows some common patterns of the human aging process. For kids, the main appearance change is the shape change caused by cranium growth. For adults, the appearance change is mainly reflected in wrinkles Suo et al. [2012]. Various face aging approaches have been proposed to model such dynamic aging patterns, which can be roughly divided into two types Suo et al. [2012], namely, prototype approaches Kemelmacher-Shlizerman et al. [2014]; Tiddeman et al. [2001] and physical model approaches Suo et al. [2010]; Tazoe et al. [2012]. The physical model approaches employ parametric models to simulate face aging by modeling the aging mechanisms of muscles, skins, or cranium. However, these approaches are very complex and computationally expensive, which require a large number of face sequences of the same person covering a wide range of ages. However, only few of the current face aging datasets can provide sufficient data. In contrast, the prototype approaches Kemelmacher-Shlizerman et al. [2014] do not require face sequences of the same person with continuous ages. The prototype approaches model face aging using a non-parametric model. First, all the available faces are divided into discrete age groups, and an average face within each age group is

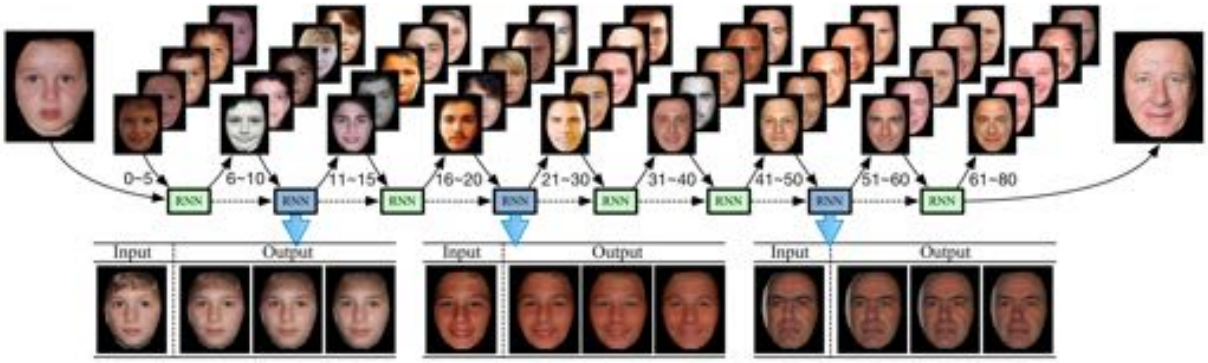


Figure 4.1: The recurrent face aging (RFA) framework exploits a RNN to model the aging pattern. The aged face is synthesized by referring to the autoregressive memory of the previous faces. The intermediate transitional faces can also be synthesized.

computed as a prior. The difference between the average faces is treated as the aging pattern and the pattern is transferred to each individual face to produce an aged face. However, the prototype approaches totally discard the personalized information and all the aged people have the same texture since they share the same rigid aging pattern. Moreover, regardless of the model type, all these methods perform a sharp one-shot transformation from one age group to another by learning a single mapping function. Thus, the one-shot mapping function fails to capture the dynamics of the in-between face sequence between adjacent age groups.

Similar to the prototype approaches, we divide the faces of each gender into 9 age groups, as shown in Fig. 5.2. Therefore, our model only requires the faces of the same person covering two adjacent groups. This setting effectively alleviates data insufficiency of the face sequences which cover long time span. To model the complex yet smooth dynamics of face aging, we propose a recurrent face aging (RFA) framework whose units in the hidden layers are autoregressively connected. As shown in Fig. 5.2, the RNN transforms a face across different age groups by decomposing the complex image generation process into a sequence of intermediate states with smaller and subtle changes. The intermediate states do not necessarily correspond to intermediate age points. One of the many benefits of using RNN is that the autoregressive connections within the layers allow the framework to progressively generate a complex image. To be specific, RNN is applied to construct complex image iteratively where the rough outlines are gradually replaced by precise forms, and lines are sharpened. RNN can solve the difficulty in learning a one-shot transformation from A to B effectively.

Aiming to generate an entire scene at once from A to B would preclude the possibility of iterative self-correction and introduce significant intrinsic error when the transformation is very complex. Even though the deep convolutional network is powerful, it is still very challenging to learn the transformation from A to B in one shot. The one-shot approach is fundamentally

difficult to scale to high resolution images with richer details. Therefore, instead of learning the transformation in one shot, we transform A to B progressively via recursive process which can be understood as a painting process. At first, only the image sketch is painted and the details are added iteratively. The iterative process makes the network more powerful to generate high resolution images.

The purpose of the recurrent process is not to exactly model the incremental aging process. Instead, the recurrent process works as a computational mechanism which decomposes a complex image generation process into a sequence of simpler steps. Besides, the quality of the warped image is sensitive to the quality of optical flow. Large and low-quality optical flow will result in unexpected ghosting effects – some facial parts are very blurry. Take the last face image in Fig. 4.13 as an example, the region centered at the mouth is very blurry, and these blurry effects represent the ghosting artifacts. The recurrent converging process could generate a sequence of low-rank faces. The calculated optical flows between the original image and the low-rank image sequence are a sequence of small high-quality optical flows. The young face can be progressively warped to the target low-rank aged face smoothly with the help of the sequence of the small high-quality optical flows, and a sequence of warped young faces can be obtained. Finally, a sequence of intermediate aged face could be obtained by mixing the textures of the warped young face and the nearest old face of the same age in the training set. As shown in Fig. 4.15, the output aged faces of our recurrent net is free of ghosting artifacts compared with other one-shot methods (such as the coupled dictionary learning method Shu et al. [2015]).

This paper is the extension of our previous conference work Wang et al. [2016a]. In particular, we extend our previous bi-layer model to a triple-layer GRU. The triple-layer GRU is more flexible compared with the single GRU. For the single GRU, its hidden state, which represents the aged face, must have the same dimension as the input face. But this will limit its encoding power. However, in our triple-layer GRU unit, we only need to set the dimension of the hidden state of the top GRU to be the same with the input face, while the hidden variables in the middle and bottom GRUs can be set to different dimensions. We set the hidden variables with a high dimension since using high dimension could boost its capability to encode complex high dimensional signals. The deeper GRU is able not only to represent the face with a higher internal dimension, but it is also more powerful in modelling complex signals with the help of non-linear operations (e.g., tanh layer and sigmoid layer). These nonlinear operations are not just a simple projection to change the dimensions, they are a complex non-linear dynamic system. Each layer of RFA is autoregressive: every GRU receives the input from both the preceding GRU which is nested in the same layer and the GRU which is from its bottom layer. The triple-layer hierarchical GRU has better performance than the bi-layer GRU.

The collected face images, however, usually have various expressions. Mild expressions

can have a dramastic effect in face analysis methods Tulyakov et al. [2016a]; Xiong and De la Torre [2013], especially on the face landmark positions. We employ a two-step face normalization process to normalize the faces. In the first step, we follow the pipeline in Kemelmacher-Shlizerman et al. [2014] to learn a robust eigenface space. The high dimensional eigenface captures the texture information, (*e.g.*, the expressions). In the second step, we progressively remove the high dimensional eigenfaces and reconstruct the image with the left eigenfaces. Then the original image is warped to the reconstructed one using optical flow. In this way, the expressions will be progressively alleviated. In the training process, the noise (*e.g.*, the detailed wrinkles) should be filtered out while the regular shading (*e.g.*, the shading around the mouth) and texture information should be kept in the training phase. The eigenfaces Turk and Pentland [1991] are very robust to noise as they capture the global structure information. Thus, after obtaining the normalized images, we project the images to the eigenface space and take their low rank coefficients as the image representation input to the RFA framework. After obtaining the synthesized low rank aged face, we synthesize the textures by transferring the textures from its nearest neighbor in the eigenface space. As shown in Fig. 5.2, we can generate realistic old faces with detailed textures.

To summarize, our paper makes these contributions: (1) We propose a face aging process between every two neighbouring groups with an RNN network. (2) Our method generates smooth intermediate faces and it handles the ghosting artifacts properly. (3) To the best of our knowledge, we are the first to do face aging based on RNN with hierarchical autoregressive memories. (4) We present a progressive face normalization process. (5) We collect a large face dataset with age labels for the purpose of research.

4.2 Related Work

4.2.1 Face Aging

Face aging has been a hot research topic recently. Face aging models can be roughly divided into prototype approaches and physical model approaches Fu et al. [2010]; Ramanathan et al. [2009]. The prototype approaches Tiddeman et al. [2001] aim at constructing an average face as prototypes for the young and old groups, and transferring the texture difference between the prototypes to the test image. The state-of-the-art prototype method Kemelmacher-Shlizerman et al. [2014] improves the result by replacing the average face with a relighted average face whose lighting condition can be tuned to be the same with the input. The relighted average faces are calculated in the eigenface domain. Eigenfaces for different groups are different. For example, the eigenfaces in the older age group contain a lot of general aged information, such as the shadings. Then the texture can be synthesized by the eigenfaces through reconstruction. Apart from the lighting considerations, the geometry/shape transformation is implemented us-

ing optical flow. The texture transformation is calculated by deducting the reconstructed face in the target group by the reconstructed face from the source group Kemelmacher-Shlizerman et al. [2014]. However, the limitation of the prototype models still exists: face texture changes are the same for different inputs. Therefore, these methods lack individuality. Besides, the detailed texture information (*e.g.*, wrinkles) is averaged out. Recently, a coupled dictionary learning (CDL) model Shu et al. [2015] was proposed. Similar to Wang et al. [2016f], the CDL model learns a dictionary for each age group. The face is reconstructed by the dictionary basis Shu et al. [2015] instead of the eigenfaces Kemelmacher-Shlizerman et al. [2014]. This model assumes that the sparse coefficients of the same person remain the same across the dictionaries. Thus, the aging patterns are encoded by the dictionary bases. Every two neighbouring dictionaries are learned jointly. In addition, the reconstruction error is regarded as the personalized information and it is added into the synthesized aged face directly. However, this method has ghost artifacts as the reconstruction residual does not evolve over time. Recently, deep generative models such as generative adversarial networks (GANs) Goodfellow et al. [2014] and variational auto-encoders (VAEs) Kingma and Welling [2014] exhibited spectacular results in image generation. Both GANs and VAEs have been applied to the problem of image segmentation Xiao et al. [2017] and synthesis Hou et al. [2017a]. For instance, in Hou et al. [2017a] Hou *et al.* proposed a variation of traditional VAEs which considers a perceptual loss and demonstrated that this model is effective for capturing the information that encodes facial expressions.

The physical model approaches try to model the aging mechanisms of the geometry (*e.g.*, such as the width of the face and the change of the aspect ratio Tazoe et al. [2012]) and muscles Ramanathan and Chellappa [2008]. The detailed facial geometry, such as the textures can also be synthesizd by statistic models Golovinskiy et al. [2006]. However, these physical models usually rely on a 3-D face model which requires a large amount of detailed 3-D face data which cover a long time range. However, the available data are limited in practice and they are not sufficient for training a complex model. Moreover, the synthesized images using these models often look not realistic. Different from Tazoe et al. [2012]; Ramanathan and Chellappa [2008]; Golovinskiy et al. [2006], our method only requires the face images to cover two neighbouring age groups (i.e., our method is more applicable in practice) and the generated images look more realistic.

4.2.2 Face Normalization

Face normalization is a very important preprocessing step for face aging. To normalize the faces, many works rely on the facial landmarks, such as Zhu et al. [2015b]; Learned-Miller et al. [2016]. To align the faces, many methods are proposed, such as Congealing method Huang et al. [2007a] and RASL Peng et al. [2012] method. Congealing Huang et al. [2007a] is an unsupervised method which tries to find the similarity transformation for the pixels. RASL Peng

et al. [2012] uses sparse representation of the faces to align a batch of linearly correlated faces. Both Congealing and RASL align faces directly without using facial landmarks. However, with the help of facial landmarks, we could better align the faces. The recent years have witnessed a lot of breakthroughs on face alignment, such as the classical supervised descent method Xiong and De la Torre [2013] which uses the hand-crafted features (*i.e., SIFT and HOG*) and the most recent recurrent face alignment method Wang et al. [2016b] which uses the deep features extracted using convolutional neural networks. The facial landmarks have been successfully applied to face normalization problems. For example, in the task of pose-invariant expression recognition, Rudovic *et al.* Rudovic et al. [2010] employs regression functions to map the 2D landmark locations to the corresponding locations in the frontal pose. In this paper, we also rely on face landmarks to align faces.

To normalize the faces, the most direct way is to warp the faces directly by aligning the detected face landmarks to the frontal face landmark template Shu et al. [2015], but this will cause unexpected face distortions. Some works rely on 3D models. For example, Zhu et al. [2015b] first normalizes the 2D landmarks with the help of 3D face model which preserves the identity and expression information. Then the face is warped to the normalized landmarks. However, all these normalization methods suffer from distortion problems. Additionally, Zhu et al. [2015b] fails to normalize facial details (*e.g.*, the gaze direction of the eyeballs cannot be normalized as the eyeballs are not included in the 3D model). To tackle these problems, we propose a progressive face normalization method which can mitigate the distortion problem and can take care of facial details.

4.2.3 Recurrent Neural Network

Traditional RNN can learn complex dynamics by mapping the input sequence to a hidden variable sequence. By passing the hidden variables recursively to the repeating module in the network, RNN is able to memorize the previous information. Thus, RNN performs well in dealing with sequential data which have interdependencies.

As a special type of RNN, Long Short Term Memory (LSTM) networks are explicitly designed to tackle the long-term dependency problem. LSTM was firstly introduced for speech recognition problems Hochreiter and Schmidhuber [1997], where the memory cells enable the LSTM network to process sequential data with interdependencies. The key idea behind LSTM is its memory state and four gates which can control the information flow inside the unit adaptively. Given the success of LSTM, many recurrent network architectures are explored, such as depth RNN Yao et al. [2015], clockwork RNN Koutnik et al. [2014] and the Gated Recurrent Unit (GRU) Cho et al. [2014]. As shown in Fig. 4.6, GRU is a simplified version of LSTM. GRU replaces the forget and input gates in LSTM with one single update gate. It also merges the cell state and the hidden state.

Given the various RNN variants, Greff et al. [2017] and Rafal Zaremba [2015] conducted a thorough search in order to find out the optimal RNN structure, such as the classical LSTM Graves and Schmidhuber [2005], the LSTM with 'peephole connections' between the memory cell state and the gates Gers and Schmidhuber [2000]. The research in Zaremba [2015] revealed that GRU is much easier to train as it has simpler structure than LSTM. Under the situation where the forget gate of LSTM is initialized with a large value close to 1, the LSTM is not significantly different from GRU. They also discovered that the forget gate is the most significant gate, the input gate is relatively important and the output gate is not important. Three architectures which are similar to the GRU were proposed in Cho et al. [2014]. RNN has also been applied for image generation Gregor et al. [2015]; Im et al. [2016] and video analysis Zhu et al. [2017c,b]. These image generation works share a similar idea, *i.e.*, the RNN is applied to refine the generated image recursively. Different from these works whose inputs are image patches, the inputs to the RNN in this paper are the encoded hidden representations of the faces which are obtained by projecting the faces to the eigenface space.

4.3 Recurrent Face Aging

Our model conducts face aging in the following two phases which are face normalization and aging pattern learning. The normalization phase consists of two steps with the first step to learn a robust eigenface space Kemelmacher-Shlizerman and Seitz [2012]. The second step is weakening the expressions and calibrating the poses. More details of face normalization are given in Section 4.3.1. To learn the aging patterns between the neighbouring groups, we exploit RNN with hierarchical autoregressive memory.

4.3.1 Face Normalization

The most important factor to be considered in face normalization is to preserve the intrinsic age information, such that one can normalize the face group-wisely by leveraging the faces within the same age group. In this way, the age-specific information can be maintained. For instance, children's eyes are relative large compared to their overall face size, as shown in Fig. 5.2. These characteristics could be well preserved by the eigenfaces.

Another factor that needs to be considered is the smoothness of age progression between the adjacent age groups. Instead of normalizing each face group independently, we normalize the faces from every two adjacent age groups jointly. A shared eigenface space can be learned for each pair of groups. A smooth transformation can be learned in the shared eigenface space. The faces across different age pairs are all aligned to the same rigid template using the transformation matrix calculated based on the controlling points of the input face and the template face. The controlling points are the center of each eye and the mouth. The transformations include

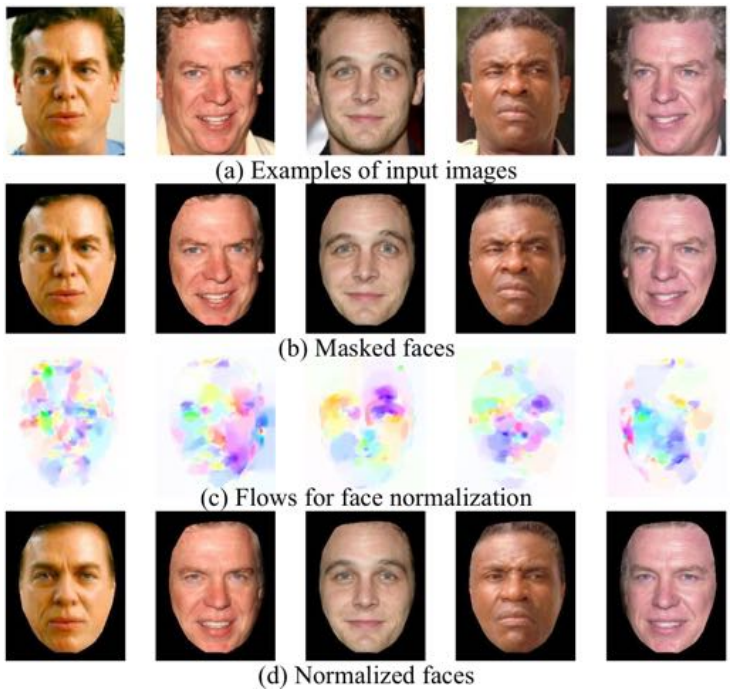


Figure 4.2: Step 1 of face normalization. (a) Examples of input images. (b) Masked images. (c) Estimated flow for face normalization. (d) Normalized faces with the estimated optical flow.

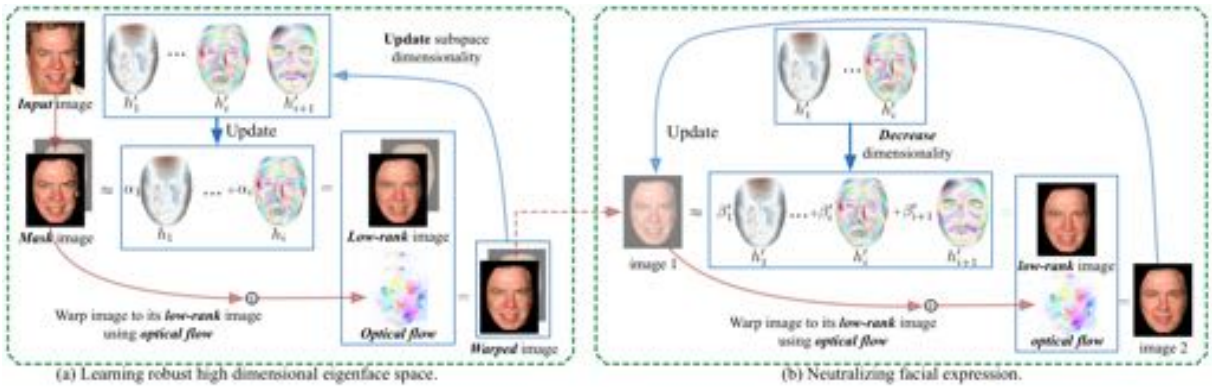


Figure 4.3: Face normalization process consists of two steps. Step 1, shown in (a), is to learn a robust eigenface space incrementally which is insensitive to the errors brought by the optical flow. Step 2, shown in (b), is to neutralize the facial expressions progressively by decreasing the dimensionality of the eigenface space.

rotation, translation and scaling. Therefore, the location of the eyes and the ratio between the inter-pupil distance and the perpendicular distance between the mouth and the center point of the eyes are fixed. After the face alignment, the subsequent facial expression normalization operation would not change the positions of the controlling points. Even though the pair-wise

subspace is different, the controlling points of the faces remain stable across different pairs. Therefore, the face aging pairs can be concatenated together seamlessly. The procedure will not provide additional drift.

As shown in Fig. 4.3, the face normalization process can be divided into **two** steps. The first step is to learn a robust eigenface space *incrementally* which is insensitive to the error brought by optical flow. Then we warp the input face to its reconstructed low-rank face using optical flow without ghosting artifacts. The second step is to neutralize facial expressions of the flowed face obtained in the first step. We warp the face image to its reconstructed low-rank face progressively by decreasing the dimensionality of the eigenface space.

The motivation of learning the subspace incrementally and reducing the dimensionality is to mitigate the ghosting effects for reconstructed images in the expression normalization process. The neutral expressions are preserved in the lower dimensional subspace. By warping the image to the low dimensional subspace using the optical flow, we can mitigate the face expression. However, if we warp the images to the low-dimensional subspace using the optical flow directly, the synthesized image will suffer severe ghosting artifacts. Besides, even if we warp the image to the high dimensional subspace using the optical flow, the reconstructed images still have severe ghosting artifacts. This is because the calculated optical flow has big error when the pixels of the original face image and the corresponding pixels of the low-rank face image have large relative displacement. To tackle this problem, we need to learn a subspace insensitive to errors brought by optical flow.

Therefore, we compute the subspace and the flowed (wrapped) images jointly. In each iteration, the flowed image is first updated using the subspace information and then the subspace dimensionality is updated using the flowed images. Even though the flowed images may still have the ghosting artifacts, they are progressively removed by updating the subspace and flowed images alternatively. In this way, a space of relatively high dimensionally being robust to optical flow errors can be learned and the ghosting artifacts of the flowed images can be mitigated. However, the high-dimensional space could not neutralize the facial expression. To obtain neutral expression, we warp the images gradually from the high-dimensional space to the low-dimensional space. Accordingly, in the second part of the face normalization process, we implement a smooth transformation from high dimensional space to a low dimensional space.

To sum up, by transferring from a low-dimensional subspace to a high-dimensional space, a robust subspace could be learned which can be utilized to reconstruct the face image without ghosting artifacts. By going back to the low-dimensional subspace from the high dimensional space, the expression could be neutralized.

Figure 4.4: (a) The first 4 eigen faces encode the lighting of the faces. (b) The other eigen faces encode face textures.

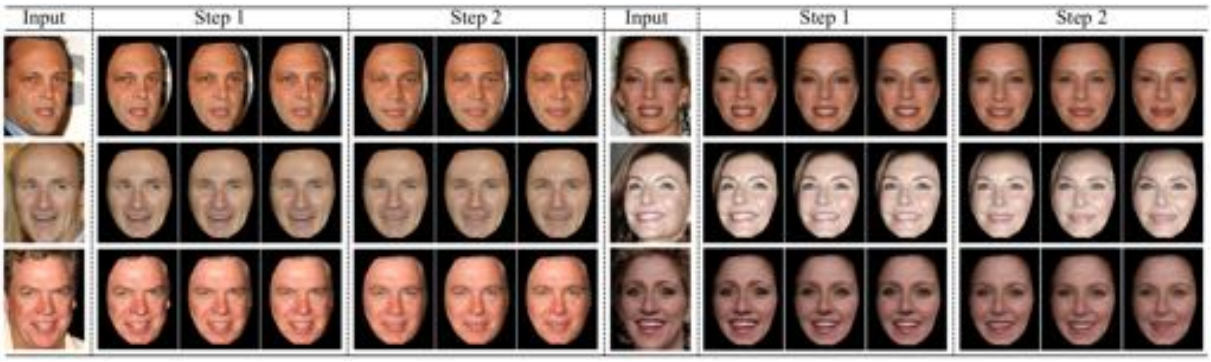


Figure 4.5: Two-step face normalization: (step 1) coarse face normalization; (step 2) progressive face normalization.

Learning Robust Eigenface

We optimize the eigenfaces and optical flow estimation iteratively. First, we stack the images column-wisely into the matrix $\mathbf{M}=[\mathbf{I}_1, ..., \mathbf{I}_n]$. Here $\mathbf{I}$ denotes an image. Then we implement singular value decomposition on $\mathbf{M}$: $\mathbf{M}=\mathbf{USV}^T$. We keep the top k eigenvectors in $\mathbf{U}$ and denote them as $\mathbf{H}=\mathbf{U}(:, 1{:}k)$. We reconstruct image $\mathbf{I}$ in the low rank eigenface space as $\mathbf{I}'=\mathbf{H}(\mathbf{H}^T\mathbf{I})$ where $\mathbf{H}^T\mathbf{I}$ means projecting the image $\mathbf{I}$ to the eigenface space $\mathbf{H}$. Then the optical flow from $\mathbf{I}'$ to $\mathbf{I}$ can be calculated, and we can get $\widehat{\mathbf{I}}'$ by warping $\mathbf{I}$ to $\mathbf{I}'$ reversely using the optical flow. As the optical flow cannot recover the images perfectly, $\widehat{\mathbf{I}}'$ and $\mathbf{I}'$ are not exactly the same and $\widehat{\mathbf{I}}'$ has ghost artifacts. To remove the ghost artifacts, we reset $\mathbf{M}=[\widehat{\mathbf{I}}_1, ..., \widehat{\mathbf{I}}_n]$ and repeat the process above until convergence. In each new face normalization process, we progressively increase the number of eigenvectors.

As shown in Fig. 4.4, the first 4 eigen faces encode the lighting condition of the image while the rest encode face textures (*e.g.*, wrinkles, mustache, *etc.*). We start the process from $k{=}4$ and terminate the process when $k{=}80$. Fig. 4.2 shows the face normalization results of the first step. It is worth noting that the expressions of the 5 people are normalized (*e.g.*, the eyeballs of the first image come to the middle; the mouth of the second image becomes horizontal; the

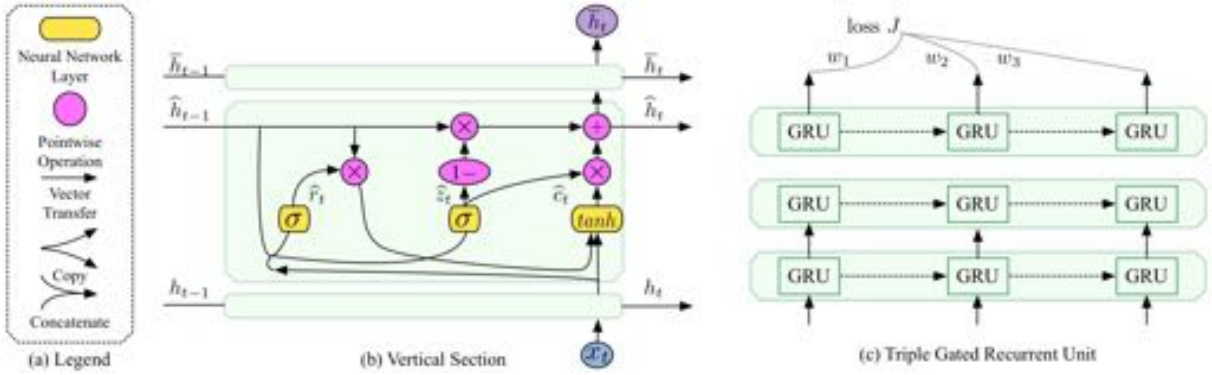


Figure 4.6: Recurrent face aging (RFA) framework with triple-layer GRU. (b) shows the vertical section of the RFA. (c) shows the overall architecture of RFA, where the weighted loss is employed to make the system focus more on the latter recurrences.

wide-open eyes of the third image become smaller; the closed eyes of the forth image are open, and the eyeball gaze direction is neutralized).

Progressive Normalization

After warping the face images to their low-rank face images which are reconstructed by 80 eigenfaces, we can observe that the expressions have been weakened. However, the normalization effect still has room to be improved as we can observe that sometimes the mouths are still open. As these expressions are depicted by the high dimensional eigenfaces, we can neutralize the face if we can weaken the influence brought by the high-dimensional eigenfaces. Inspired by this idea, we warp the images to the low-rank face images which are reconstructed only by the low dimensional eigenfaces while the high dimensional eifenfaces are discarded progressively. In this way, the normalization effect could be further improved.

To achieve this target, we first reconstruct image $\mathbf{I}$ by N eigenfaces $\mathbf{U}(:, 1{:}N)$, and warp $\mathbf{I}$ to the reconstructed face, and then obtain $\widehat{\mathbf{I}}'_N$. Then we reconstruct image $\widehat{\mathbf{I}}'_N$ by $N-1$ eigenfaces $\mathbf{U}(:, 1{:}N-1)$, and warp $\widehat{\mathbf{I}}'_N$ to the reconstructed face, then $\widehat{\mathbf{I}}'_{N-1}$ is obtained. We repeat the process above recurrently. In each recurrence, we progressively decrease the number of eigenvectors. We figure out that there is no huge visual difference between the warped image and the input image when the number of eigenfaces is greater than 10 while the shape changes dramatically when the eigenfaces are less than 10. Then we set the step size smaller when the number of eigenfaces is smaller than 10. Different step sizes, $40, 20, 10, 2$ are employed in the implementation. We start with the step size of 40. We obtain the number of eigenface sequences as $[80, 40]$. Then we further decrease the lower bound 40 with the step size of 20, we have $[40, 20]$. Then we continue deceasing the lower bounds with smaller step sizes. Finally, we obtain the

eigenface sequence of $[80, 40, 20, 10, 8, 6, 4]$, but the quality of the normalized face remains the same, and there is no obvious visual difference. The range $[80, 40, 10, 6, 4]$ is good enough to obtain a high quality normalized face. Therefore, we fix the step range to $[80, 40, 10, 6, 4]$.

The second step can improve the normalization effect to a large extent. Fig. 4.5 shows the results of face normalization process of three men and three women. We can observe that the eyes of the second man (*row 2*) are centered after the first normalization step. However, his mouth is still open. The second step closes his mouth progressively. Similar results can be observed for the other five people, especially for the women in the right column of Fig. 4.5. The second step can also normalize the small poses. The woman in row 2 of Fig. 4.5 has a yaw angle and she is looking towards the ceiling with her mouth wide open. After the second normalization step, her eyes are calibrated to looking towards the front, and the other parts, such as mouth and nose are also frontalized. The men in row 1 & 3 are also frontalized.

4.3.2 RNN-based Face Aging Model

The whole pipeline of face aging can be divided into two phases. In the first phase, we rely on the RNN to generate the low-rank aged faces. Then the final output aged face is obtained by mixing the low-rank aged face, the texture of the input young face and the texture of the most similar old face exemplar in the training set. The illustration of texture mixing process is available in Fig. 4.8.

Problem Formulation

Let $\mathbf{H}^{(k)}$ represent the shared eigenfaces for age group k and $k+1$, where each column in $\mathbf{H}^{(k)}$ denotes one eigenface. $\mathbf{H}^{(k)}$ is a complete and orthogonal matrix. The columns in $\mathbf{H}^{(k)}$ are unit vectors and they are orthogonal to each other. Let $\mathbf{\Lambda}_k=(\lambda_1, \lambda_2, ..., \lambda_n)^T$ denote the eigenvalues of the eigenfaces. Let $\mathbf{I}_y$ be the low rank young face, $\mathbf{I}_o$ be the low rank image of the ground truth of the old face, $\mathbf{I}'_o$ be the synthesized low rank old face, and $\mathbf{x}_y$, $\mathbf{x}_o$, $\mathbf{x}'_o$ be their coefficients in the eigenface space. We expect the synthesized image to be as similar as possible to the ground truth image. Therefore the loss function writes:

$$\|\mathbf{I}_o - \mathbf{I}'_o\|_F^2 = \|\mathbf{H}^{(k)}\mathbf{x}_o - \mathbf{H}^{(k)}\mathbf{x}'_o\|_F^2 = \|\mathbf{x}_o - \mathbf{x}'_o\|_F^2. \tag{4.1}$$

During the face normalization process, we observe that the first 4 eigenvalues occupy more than 60% of the total energy. The previous studies Kemelmacher-Shlizerman and Seitz [2012] revealed that the first 4 eigenfaces correspond to the lighting effect of the face while the others correspond to the texture as shown in Fig. 4.4. Thus, in order to keep the illumination consistency between the source and target images, we transfer the first 4 coefficients directly from the young image to the synthesized aged image. The high rank coefficients mainly preserve the

texture information. We rely on the high rank coefficients to learn the aging patterns. We visualize the high rank eigenfaces and find that these eigenfaces capture different texture information (*e.g.*, beard, open mouth with teeth and closed mouth). Here we normalize the distribution over these eigenfaces by dividing the coefficients by their corresponding eigenvalues. Then we propose the following loss function:

$$J = \|(\mathbf{x}_o - \mathbf{x}'_o) \odot (\frac{1}{\lambda_1}, ..., \frac{1}{\lambda_n})^T\|_F^2. \tag{4.2}$$

To optimize the objective function, we adopt RNN to learn the aging patterns as follows:

$$\mathbf{x}_y \odot (\frac{1}{\lambda_1}, ..., \frac{1}{\lambda_n})^T \xrightarrow{RNN} \mathbf{x}'_o \odot (\frac{1}{\lambda_1}, ..., \frac{1}{\lambda_n})^T. \tag{4.3}$$

The $\odot$ in Eq. (4.2) and Eq. (4.3) is an element-wise multiplication to scale the samples to the interval $[-1, 1]$ by dividing the i-th element of $\mathbf{x}$ by λ_i. Although these eigenvectors which have low energy have a small contribution to the image reconstruction, they represent face textures as shown in Fig. 3. Therefore, these eigenvectors should be emphasized as they have a big contribution to face aging. Thus, we normalize the coefficients to treat all the eigenvectors equally important. To avoid over-emphasizing these eigenvectors and the overfitting problem of the subsequent age progression model, we only select the first 80 eigenvectors to reconstruct the image. In other words, H is incomplete.

Recurrent Age Progression

Our RFA aims at learning a smooth transformation between two neighbouring age groups. Although many models can be used to learn the smooth transformation, such as LSTM, GRU, as well as their variants, we use GRU to learn the aging patterns because of its simple structure and superior performance Greff et al. [2017]. A triple-layer GRU is exploited as the basic recurrent module in our RFA, as shown in Fig. 4.6. The three GRU layers have the same structure except for the dimensions of the hidden states.

$$\begin{aligned}
\overline{z}_t &= \sigma(\overline{W}_{zh}\overline{h}_{t-1} + \overline{W}_{zx}\widehat{h}_t + \overline{b}_z) \\
\overline{r}_t &= \sigma(\overline{W}_{rh}\overline{h}_{t-1} + \overline{W}_{rx}\widehat{h}_t + \overline{b}_r) \\
\overline{c}_t &= tanh(\overline{W}_{ch}\overline{r}_t \odot \overline{h}_{t-1} + \overline{W}_{cx}\widehat{h}_t + \overline{b}_c) \\
\overline{h}_t &= (1 - \overline{z}_t) \odot \overline{h}_{t-1} + \overline{z}_t \odot \overline{c}_t
\end{aligned} \tag{4.4}$$

$$\begin{aligned}
\widehat{z}_t &= \sigma(\widehat{W}_{zh}\widehat{h}_{t-1} + \widehat{W}_{zx}h_t + \widehat{b}_z) \\
\widehat{r}_t &= \sigma(\widehat{W}_{rh}\widehat{h}_{t-1} + \widehat{W}_{rx}h_t + \widehat{b}_r) \\
\widehat{c}_t &= tanh(\widehat{W}_{ch}\widehat{r}_t \odot \widehat{h}_{t-1} + \widehat{W}_{cx}h_t + \widehat{b}_c) \\
\widehat{h}_t &= (1 - \widehat{z}_t) \odot \widehat{h}_{t-1} + \widehat{z}_t \odot \widehat{c}_t
\end{aligned} \tag{4.5}$$

$$
\begin{aligned}
z_t &= \sigma(W_{zh} h_{t-1} + W_{zx} x_t + b_z) \\
r_t &= \sigma(W_{rh} h_{t-1} + W_{rx} x_t + b_r) \\
c_t &= tanh(W_{ch} r_t \odot h_{t-1} + W_{cx} x_t + b_c) \\
h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot c_t
\end{aligned}
\tag{4.6}
$$

This triple-layer GRU works as an autoencoder. As shown in Fig. 4.6(b), the bottom GRU works as an encoder while the top GRU works as a decoder. The bottom GRU first encodes the input face to a hidden high dimensional variable, and then its output works as the input to itself for the next iteration autoregressively and the input to the middle layer.

Then the top GRU decodes the hidden high dimensional state to an aged face. The hidden states are initialized with zeros. The difference between the output and the ground truth aged face is calculated as the loss. Different rewards are assigned for the loss of each recurrence. The reward in the last recurrence is set to the largest as we take the output of the last step as the most important step and it is the final aged face. Smaller reward is set for the loss in the first recurrence since the output in the beginning is an intermediate transitional face and it is not required to be very close to the final aged face. These losses will guide the system to age the face progressively. Eq. (4.4, 4.5, 4.6) define the triple-layer GRU.

In Eq. (4.4, 4.5, 4.6), σ is the logistic sigmoid function. $[\mathbf{W}_z, \mathbf{W}_r, \mathbf{W}]$ are weight matrices and $[\mathbf{b}_z, \mathbf{b}_r, \mathbf{b}_c]$ are bias terms which need to be learned. Each GRU has two gates which are *reset* gate and *update* gate, and one hidden state.

The *reset* gate $\mathbf{r}_t$ decides whether the information of previous faces should be ignored. If $\mathbf{r}_t$ is close to 0, the previous face information is forced to be discarded, and the unit will focus on the current input face only. This gate allows the unit to remember or drop the irrelevant face information.

The *update* gate $\mathbf{z}_t$ controls the amount of face information that could be transferred from the previous state to the current state. This update gate works as the forget and input gates. Instead of calculating the value of forget and input gates separately like LSTM, the update gate in GRU calculates them together with $\mathbf{z}_t$ and $1 - \mathbf{z}_t$. This setting means that the unit only accepts the new input face when it forgets something of the previous faces. The update gate acts similarly as the memory cell in the LSTM. c_t is the new face candidate created by a *tanh* layer that could be added to the current face. Then the face candidate is merged with previous face information to form a new face (*hidden state*) with the weights generated by the update gate:

$$
\mathbf{h}_t = (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \mathbf{c}_t. \tag{4.7}
$$

The system has short-term memory and ignores the previous faces if the reset gate is activated all the time. If the update gate is always inactivated, the system can have long-term memory and all the previous faces will be memorized.

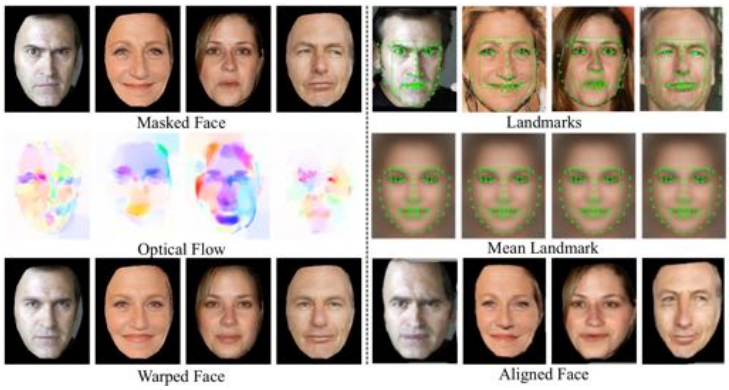


Figure 4.7: Face alignment. (left) Align the face with our two-step face normalization method. (right) Align the face to the mean position of the face landmarks via interpolation.

As we expect to age the face progressively, we add the supervision information for each recurrence. After each recurrence, the aged face is expected to move closer to the target aged face. To achieve this target, a loss is calculated for each recurrence, which is defined as the l_2 loss between the intermediate transitional face and the target aged face. Therefore, in our RFA framework, RNN acts as a refinement process which transforms the young face *progressively* to the aged face. In our settings, each basic unit will iterate for 3 times. The **input series** $[\mathbf{x}_1, ..., \mathbf{x}_n]$ is the replicates of the young face. The **loss** is calculated after each recurrence. We expect to age the face gradually. In other words, the in-between faces should become more similar to the target face after each iteration. Thus, the loss between the in-between faces and the target faces should become smaller gradually in the training process. To meet this requirement, we set a series of rewards $\mathbf{w}$=$[0.1, 1, 10]$ for the loss as shown in Fig. 4.6(c). The rewards increase monotonically as we expect that the faces could be transformed to the target face gradually. We assign the largest reward for the loss in the last recurrence. Therefore, the system will pay more attention to the last recurrence. For the transformation from group k to k+1, the loss function writes:

$$J = \sum_{t=1}^{n} \mathbf{w}_t \|(\mathbf{x}_{k+1} - \widehat{\mathbf{h}}_t) \odot (1/\mathbf{\Lambda}_k)\|_F^2 \ , \tag{4.8}$$

where $\mathbf{x}_{k+1}$ represents the target image in group $k+1$, $\widehat{\mathbf{h}}_t$ is the in-between states during the recurrent training process, and $\mathbf{w}_t$ is the weight for recurrence step t.

The first four eigenfaces correspond to the lighting while the others correspond to the textures. Thus, the projection over these four eigenfaces has no contribution to the aging effects. Thus, the first four projections are neglected during the training process. The values corresponding to the rank-4 eigenfaces are transferred directly from the source images during testing phase.

4.3.3 Face Aging cross Multiple Age Groups

After training the RNNs for every two neighbouring age groups, we chain up RNNs to operate at different age gaps in order to produce a chain of progressively aged faces. The concatenation of the RNNs is described in Section 4.3.3. When we obtain the low-rank aged face, we mix the textures of the young face and most similar face in the training set to synthesize the aged face. The texture mixing is described in Section 4.3.3.

RNN concatenation

Finally, we concatenate the RNN pieces to form a complete RFA architecture. For each pair of neighbouring RNNs, their corresponding low-rank eigenface spaces are different. Thus, the output of the previous RNN cannot be used as input to the following RNN directly. We rely on the following formula to transform the output ($\mathbf{x}_{k+1}$) of the k-th RNN to the input ($\overline{\mathbf{x}}_{k+1}$) of the $(k+1)$-th RNN.

$$\overline{\mathbf{x}}_{k+1}=\mathbf{U}_{k+1}(\mathbf{U}_k\mathbf{x}_{k+1})=(\mathbf{U}_{k+1}\mathbf{U}_k)\mathbf{x}_{k+1}=\overline{\mathbf{U}}\mathbf{x}_{k+1} \tag{4.9}$$

where $\mathbf{x}_{k+1}$ is a column vector and it is the output of the k-th RNN. $\mathbf{U}_k\mathbf{x}_{k+1}$ is its low rank image. Then the operation $\mathbf{U}_{k+1}(\mathbf{U}_k\mathbf{x}_{k+1})$ reprojects the image to the eigenface space of $k+1$-th RNN. This transformation can be integrated into the RNN framework. As shown in Fig. 4.6, the term $\mathbf{W}\mathbf{x}_t$ in the first three equations can be transformed as following:

$$\mathbf{W}\mathbf{x}_t = \mathbf{W}(\overline{\mathbf{U}}\mathbf{x}_k)=(\mathbf{W}\overline{\mathbf{U}})\mathbf{x}_k = \overline{\mathbf{W}}\mathbf{x}_k. \tag{4.10}$$

By replacing $\mathbf{W}$ with $\overline{\mathbf{W}}$ calculated by Eq. (4.10), the transformation can be embedded into the RNN pipeline.

Texture Transfer

After obtaining the synthesized aged low rank face $\mathbf{I}'$, we warp the input face to the coordinates of $\mathbf{I}'$, and obtain $\mathbf{I}$. Let $\widehat{\mathbf{I}}$ be the low rank face of $\mathbf{I}$. Thus, the detailed texture of the young face is preserved in $\mathbf{I}-\widehat{\mathbf{I}}$. Though the high dimensional eigenface captures the texture information, it is still not good enough because the fine grained details are filtered out as noise by PCA. To tackle this problem, we transfer the texture from its nearest neighbour $\mathbf{J}$ in the low rank eigen face space to the synthesized low rank face.

We measure the distance between the synthesized low rank aged face and other aged faces based on the Euclidean distance between their low rank coefficients. Then we select its closest neighbour $\mathbf{J}$. The low rank face of $\mathbf{J}$ is denoted with $\widehat{\mathbf{J}}$. The texture of the nearest neighbour is preserved by $\mathbf{J}-\widehat{\mathbf{J}}$. Thus, the texture of the aged face can be synthesized by the linear combination of the textures $\alpha(\mathbf{I}-\widehat{\mathbf{I}}) + (1-\alpha(\mathbf{J}-\widehat{\mathbf{J}}))$. Then we add the texture to the synthesized aged

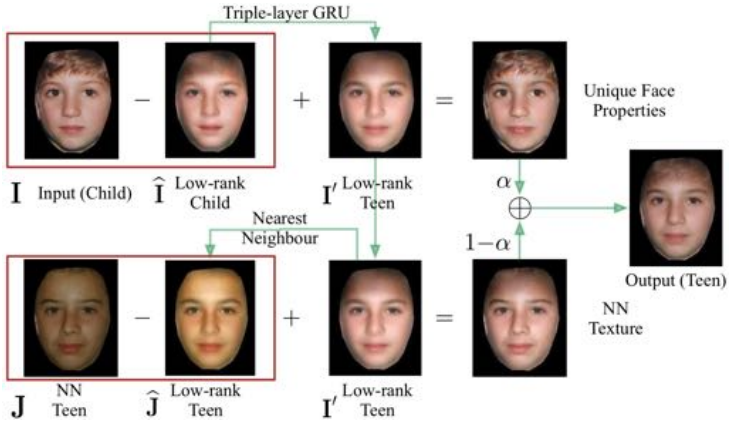


Figure 4.8: Texture transfer from the nearest neighbour.

face, and obtain the aged face. Feature transfer from its nearest neighbour $\mathbf{J}$ to $\mathbf{I}'$ is illustrated in Fig. 4.8.

As shown in Fig. 4.8, this process can merge the detailed texture characteristics from the target age group to the synthesized aged face. Therefore, the age of the synthesized face will look much closer to the target age. The smaller the α is, the more original texture details will be discarded and the more texture details from the nearest neighbour will be fused into the target aged face. From this perspective, the smaller the α is, the better the result will be. However, if α is too small, most of the texture information from the original face will be discarded. This will bring us a new question: will the person lose its identity information if we discard too much original texture information?

Table 4.1: Equal Error Rate VS α

α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
equal error (%)	8.50	8.25	**8.15**	8.47	8.80	10.28	14.42	19.74	26.98	34.07

To find out a suitable value of α, we rely on the results of the face verification system. Given two images, the face verification system needs to predict whether the two images are the same person or not. Thus, the face verification task is a binary classification task. The details of face verification task are available in Section 4.4.3, and the settings for the experiments are similar except for the dataset. In the training phase, 10% of the data are selected for validation and these data are not used in the training process. The most widely used metric for face verification is the equal error rate which is the rate between false acceptance rate and false rejection rate (FAR-FRR). The smaller the rate is, the better result we will have. Table 4.1 shows the result of FAR-FRR with respect to different values of α. We can observe that when α is below a certain threshold, it will benefit and boost the face verification result. However, when it is greater than

a threshold, the result will become really worse. From Table 4.1, we can observe that the best face verification result could be obtained when $\alpha = 0.3$. Therefore, we fix α to 0.3.

4.4 Experiments

In this section, we first describe the details of data collection and image pre-processing, followed by introducing the implementation details of the RFA framework. Then we will show the qualitative experimental results, as well as the quantitative evaluations.

4.4.1 Data Collection

We collect face images following a celebrity list which contains 3,561 celebrities from the dataset of Labeled Faces in the Wild (LFW) Huang et al. [2007b]. We collect 163,810 images from Google and Bing image search engines where 3,240 celebrities have photos which cover different age groups. We also use the images from the Morph Aging Dataset Ricanek Jr and Tesafaye [2006] and Cross-Age Celebrity Dataset (CACD) Chen et al. [2014]. The Morph Aging Dataset contains 13,000 people with 55,134 images. The CACD dataset contains 163,446 photos of 2,000 people. Both datasets contain multiple images for each person which cover different age groups. In order to ensure the high quality of the data, we remove the images which have large poses (greater than 30 degrees in yaw and pitch angles). For each crawled image, the groundtruth of its age is estimated by an off-the-shell age estimator Li et al. [2012]. Then we manually check the accuracy of the estimated age.

Finally, we have 4,371 photos for male and 6,264 photos for female in total. Each person has several photos covering 2 or 3 age groups. After dividing the data into 9 age groups for both male and female: 0-5, 6-10, 11-15, 16-20, 21-30, 31-40, 41-50, 51-60, 61-80, we obtain 2,611 image pairs which cover two neighbour age groups for male and 3,821 pairs for female in total.

4.4.2 Implementation Details

Face Normalization

We follow a similar pipeline as Kemelmacher-Shlizerman and Seitz [2011] for image pre-processing which includes face detection, landmark localization, pose estimation, and masking the images.

First, we detect the faces in the images, and then we detect the 66 face landmarks Taigman et al. [2014] given the detected position (the green rectangles in Fig. 4.7 (right)) of each face. We coarsely align the faces according to the centers of the eyes and mouth. The centers of the eyes are set symmetric to the vertical line of the template. The center of the mouth is set along the vertical line. The distance between the eyes and the distance between the mouth center and

the middle of the two eyes are set to a fixed ratio. Then we implement the rotation, scaling and translation transformation to fit the face to the rigid template. All the faces are coarsely aligned via the same strategy. Then we normalize the coarsely aligned faces with the optical flow progressively following the second step of the normalization pipeline. Finally, we can obtain the well aligned faces as shown in Fig. 4.7 (left). We also try to align the faces according to the mean position of the face landmarks via interpolation. The mean position of the landmarks are calculated by taking the average of all the faces. However, the aligned face will be twisted as shown in Fig. 4.7 (right).

In the face normalization process, we revise the optical flow implementation of Liu [2009] to calculate the flow because of its superior performance Kemelmacher-Shlizerman et al. [2014]; Kemelmacher-Shlizerman and Seitz [2012]. We follow Kemelmacher-Shlizerman and Seitz [2012] to set the parameters in the optical flow algorithm: α=0.01, $ratio$=0.75, $minWidth$=60, $nOuterFPIterations$=5, $iInnerFPIteration$=1, $nCGIterations$=50. The running time of each image is around 0.7s.

Face Aging

After performing face normalization, we calculate 80 low rank eigenfaces for every two neighbouring age groups with respect to each RGB channel. Then for each image, we concatenate its coefficients over the eigenfaces from three channels and get $d = 240$ dimensional representations. Those coefficients are used as the input for the triple-layer GRU. After removing the rank-4 coefficients which correspond to the illumination, the input to the RNN is 228 dimension. The bottom GRU acts as an encoder which encodes the input to a hidden state h_t as shown in Fig. 4.6. The hidden unit dimension can be set to different values from the input dimension d. The output dimension of the hidden state of the bottom GRU is set to be $d\times k$. To strengthen the encoding capability for various faces, we set $k = 15$, which can generate satisfactory in-between faces and also consumes less training time (around half an hour for each RNN). The dimension of the hidden state of the middle GRU is set to be the same with its input dimension which is $d\times k$. The dimension of the hidden state of the top GRU is set to be the same as the input dimension since it needs to decode the hidden representations to an aged face which has the same dimension of the input face.

4.4.3 Evaluation

To evaluate the performance of our RFA method, both qualitative and quantitative comparisons are implemented. For the qualitative evaluation, in order to demonstrate the benefits brought by our smooth progressive RFA method, we mainly compare with other one-shot methods which include the state-of-the-art Coupled Dictionary Learning (CDL) method Shu et al. [2015], the

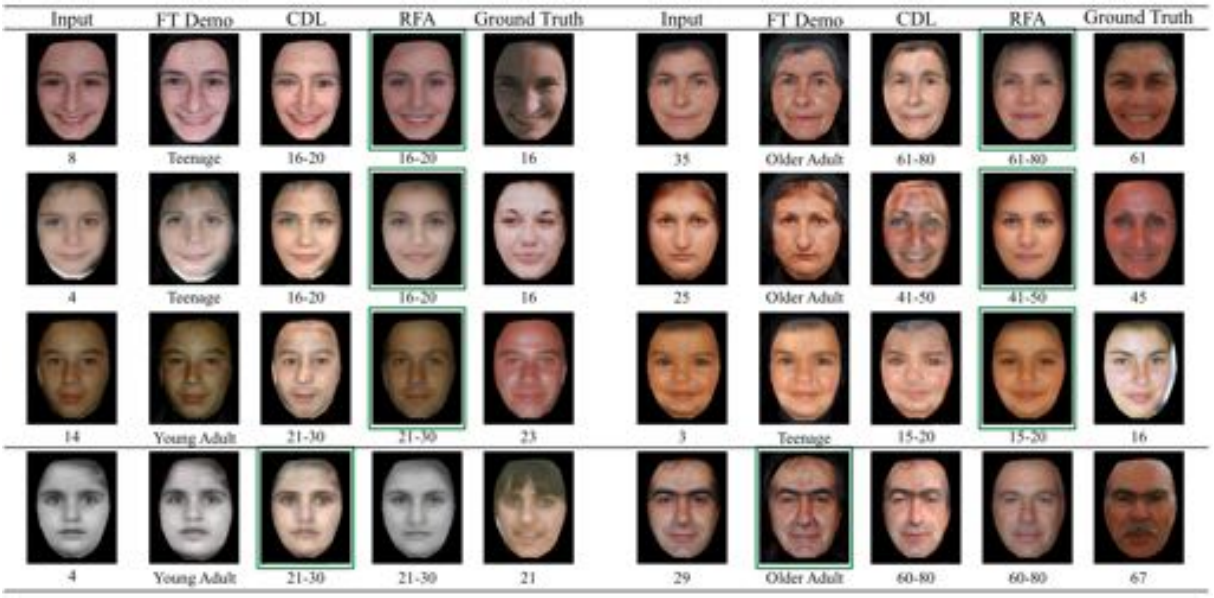


Figure 4.9: Face aging results comparison between FT Demo, Coupled Dictionary Learning (CDL) Shu et al. [2015], RFA, and the ground truth (GT). The images in the green boxes are aged faces which are most similar to the GT. Usually, our RFA method can beat the other methods.

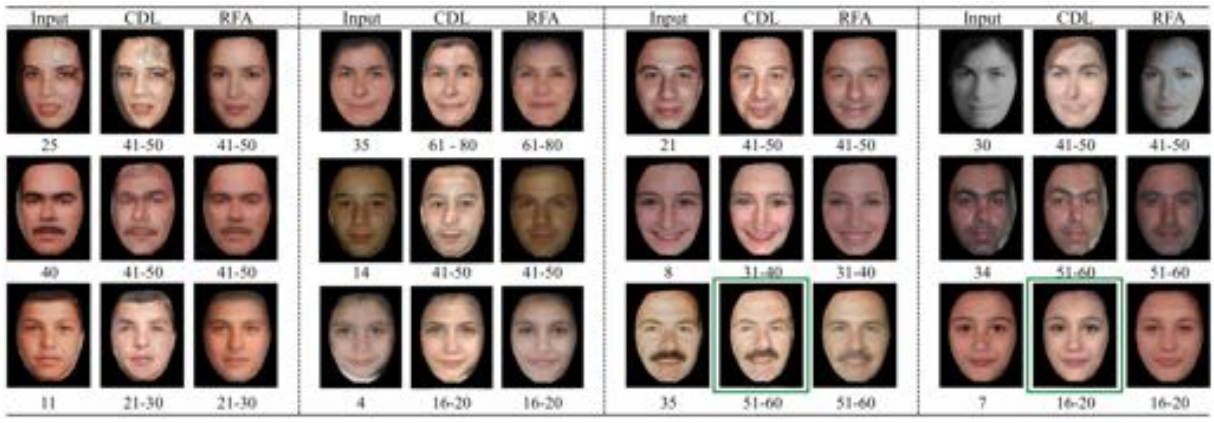


Figure 4.10: Comparison between CDL and RFA. We do not include the ground truth images as some of them are unavailable. We can observe that the aged face generated by our method matches the characteristics of the target age group well (*e.g.*, the aged face in row 2, column 3) gets some wrinkles, and his eyes become smaller during the aging process). But for some cases our aged faces are not so clear as the ones generated by CDL, such as the examples in the green boxes.

Face Transformer (FT) demo (a free online website), as well as the one-shot linear regression method which replaces the recurrent process in RFA system with a one-short linear regression. In the qualitative comparison, we mainly consider two most important factors for face

aging, which are the identity accuracy and age accuracy. For the quantitative comparison, besides the CDL, we also compare with the illumination aware age progression (IAAP) method Kemelmacher-Shlizerman et al. [2014] and two-layer GRU Wang et al. [2016a] method.

Quality of Synthesized Images

We compare the performance of our method with another two face aging models, *i.e.* the coupled dictionary learning (CDL) model Shu et al. [2015] and Face Transformer (FT) demo[1]. The results are shown in Fig. 4.9 and 4.10. CDL defines the same 9 age groups as ours. The FT Demo has fewer age groups: Baby, Child, Teenage, Young Adult and Older Adult. For fair comparison with FT Demo, we select the pairs from our dataset with large age gaps such that the ages of the images can be consistent with that in FT Demo. Some experimental results are shown in Fig. 4.9. We further compare our method with CDL on fine-grained age groups as shown in Fig. 4.10. The images from FG-NET database Lanitis et al. [2002] are used as the test data. As some examples do not have the ground truth aged faces, we do not include the ground truth in Fig. 4.10. In Fig. 4.9 and 4.10, the aged faces in the green boxes are the results which are considered to have the best aging effects. One can observe that the images generated by CDL and FT Demo suffer from the ghost artifacts. However, there are a few cases when our method outputs blurry images, such as the two exemplars in the black box at the bottom of Fig. 4.9. Even though the aged faces are blurry compared with other baselines, our synthesized aged face are still visually more reasonable. First, the illumination condition of the synthesized aged face by our method remains the same with the young face in both exemplars. Second, our synthesized aged face of the first exemplar is free of the ghosting artifacts while the synthesized face by CDL has ghosting artifacts and it looks like there are stains in the synthesized face. Third, the eyebrows of the second exemplar remains black in the synthesized aged face by FT-Demo while our method generates grey eyebrows which is visually similar to the ground truth old face. Generally speaking, our method usually generates images with more realistic appearance.

Comparison with Prior Works

Several prior works released their best face aging results Kemelmacher-Shlizerman et al. [2014]; Park et al. [2008]; Scherbaum et al. [2007]; Shu et al. [2015]; Suo et al. [2010]. Shu et al. [2015] summarized all the posted images, and found that there were 246 aged faces with 72 input images in total. We synthesize the aged face with the same age range of these methods for each input image. Similar to prior works, we evaluate our results through user study.

[1] http://cherry.dcs.aber.ac.uk/Transformer/

In the user study, each subject views three images: the young image C, and the aged images B & A which are generated by other methods and our method respectively. We set two metrics for the evaluation, age accuracy and identity accuracy. Each subject is asked to evaluate the images based on these two metrics simultaneously and the two metrics are weighted equally. The weights for two metrics, namely, the age and identity, are set to $[0.5, 0.5]$. The average score of the two metrics works as the final score. Three types of scores are provided. If A is better, it gets a score of 1. If B is better, A gets a score of 0. If A and B are similar, then both of them get the score of 0.5. We invited 40 people to evaluate our results and got 9840 votes in total. When we collect the vote, we compare the output of our method with the aged faces with other methods together. In order to avoid the bias as much as possible, we compare our method with each baseline separately.

Table 4.2: Comparison between RFA and other baselines.

	Baseline is better	RFA is better	Equivalently good
Kemelmacher-Shlizerman et al. [2014]	24%	49%	27%
Park et al. [2008]	18%	76%	6%
Scherbaum et al. [2007]	15%	71%	14%
Shu et al. [2015]	38%	46%	16%
Suo et al. [2010]	21%	53%	26%

Table 4.2 shows the score statistics (first column: the percentage of vote that agrees that the baseline is better, second column: the percentage of vote that agrees that the RFA is better, third column: the percentage of vote that agrees that the two methods are equivalently good). We can observe that our method RFA always has better performance compared with other baselines. In order to get rid of the bias of the user study, we further test the identity accuracy and age accuracy using a face verification system and an age estimator. The details are in the following two subsections.

Cross-age Face Verification

Another important factor that we need to consider for face aging is the identity. The aging system should guarantee that the synthesized aged face should have the same identity as the ground-truth old face.

During the last decade, many groups have made breakthroughs for face verification Sun et al. [2014], Taigman et al. [2014]. We employ the deep Convolutional Neural Network model introduced in Sun et al. [2014] for face verification. To evaluate the performance of our method for the cross-age face verification, we exploit FG-NET dataset which consists of 1,002 photos

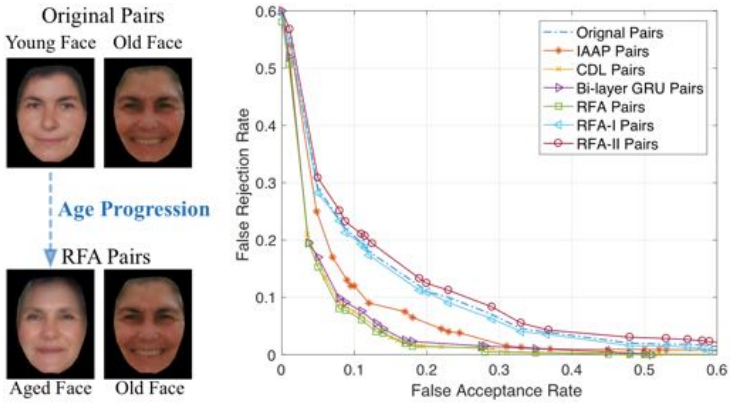


Figure 4.11: FAR-FRR curve of different methods.

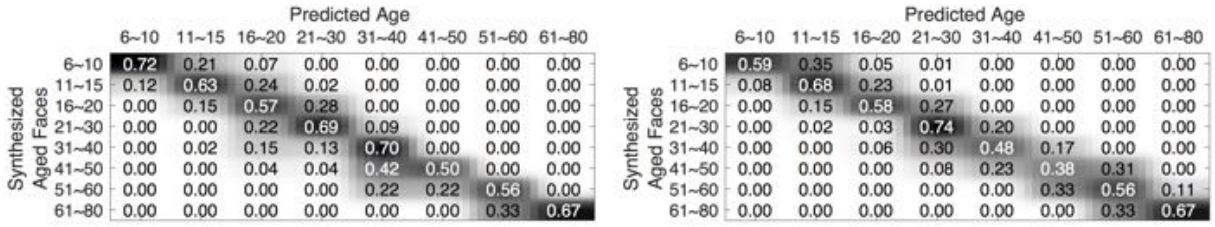


Figure 4.12: Confusion matrix of the estimated ages of (a) the synthesized aged faces with **triple-layer RNN** and their targeted age groups, (b) the synthesized aged faces with **bi-layer RNN** and their targeted age groups.

of 82 people as our input. We select the image pair in which the age gap is larger than 20 years. 916 image pairs are obtained in total. We further select 916 image pairs randomly from different people as negative pairs. We name these pairs as "Original Pairs", which are denoted as (A, B). A denotes the young face, B denotes the old face. B could either be the same person as A if it is a positive pair or a different person if it is a negative pair. For the positive pairs, as there is an age gap between A and B, the two faces are visually different. Sometimes, even though A and B are the same person, the face verification system will mis-classify them as different people. To mitigate the influence of the age gap and improve the performance of the face verification system for cross-age faces, we use our RFA system to age the young face. Then the input pairs are the synthesized aged face $\widehat{A}$ and the old face B. We name these new image pairs $(\widehat{A}, B)$ as "RFA Pairs". We also name the pairs whose aged faces are synthesized by the CDL Shu et al. [2015] method and the illumination aware age progression (IAAP) method Kemelmacher-Shlizerman et al. [2014] as "CDL Pairs" and "IAAP Pairs" respectively. We also compare our RFA (triple-layer GRU) with the bi-layer GRU by removing the middle layer. We have also evaluated the RFA without the face normalization module (denoted as "RFA-I pairs") and the RFA without the RNN module (denoted as "RFA-II Pairs"). Fig. 4.11 shows the input

face pairs for the face verification system. Therefore, no matter how similar the synthesized face is compared with the input young face A, it will not be classified as the same person with B, unless the synthesized face is similar to the B.

Table 4.3: Equal error rate (EER) (%)

Pairs	Original	CDL	Bi-layer GRU	RFA-I	RFA-II	RFA
EER(%)	14.89	8.53	8.69	14.61	15.35	8.21

Fig. 4.11 and Table 4.3 show the face verification results for different image pairs. The x axis in Fig. 4.11 denotes the false acceptance rate. False acceptance means that the image pairs from two different people are classified as from the same person. The y axis denotes the false rejection rate. False rejection means that the image pairs from the same person are classified as from different people. For each input pair, a prediction score could be obtained from face verification system. By setting the score threshold to different values, different error rates could be obtained. A balanced equal error rate could be obtained by comparing the two rates. Usually, the equal error rate is employed as the evaluation metric. Fig. 4.11 shows the false acceptance rate versus false rejection rate (FAR-FRR) curve by setting the threshold to different values.

As shown in Fig. 4.11, our RFA method has the best performance among all these methods. The two layer GRU has comparable performance with CDL and outperforms the rest of the baselines. Table 4.3 shows the equal error rate (EER) of the 6 methods, and RFA has better performance than all the baselines. Our method decreases the equal error rate (EER) by 6.68% compared with the original pairs. This observation validates the fact that our method can effectively mitigate the FRR consequence caused by large age gap. The triple-layer RNN can better preserve the identity information than the bi-layer RNN. From Table 4.3, we can draw a similar conclusion for face aging: the triple-layer RNN has better performance than the bi-layer RNN.

Age Estimation of the Synthesized Images

From Section 4.4.3, we can observe that the aged face can boost the performance of the face verification system. Another very important consideration is that the face aging system can age a person to the target age groups accurately. Assume that we are aging a person I from the age group A(*0-5*) to age group B(*6-10*) where A represents the source age group, and B denotes the target aged group. To verify the performance of our RFA system, we need to judge whether the synthesized face $\widehat{I}$ has the same age as the target age group B. 20 human annotators were asked to estimate which age group the synthesized aged faces belong to. For each aged face image, 20 votes were collected and these votes cover several different age groups. We chose the age group receiving the most votes as the estimated age.

We employ FG-Net to do the experiment. The FG-Net contains 82 people (47 male and

35 male) with 1,002 photos (571 male and 431 female). Among these images, the numbers of image pairs with the same identity which cover each two adjacent groups for male are [97, 65, 41, 46, 23, 12, 5, 3], and for the female, the numbers of image pairs are [53, 53, 52, 42, 31, 14, 4, 0]. The numbers of image pairs for both genders are [150, 118, 93, 88, 54, 26, 9, 3]. We evaluate multiple age groups. Given the images from 8 different age groups (from group 1 to group 8), we synthesize their corresponding aged faces (from group 2 to group 9), and then we estimate the ages of the synthesized faces to see if they have the same age of the target age.

Fig. 4.12 demonstrates the estimated ages of the synthesized aged faces (either from the triple-layer RFA or the bi-layer RNNs). The columns represent the target ages of the synthesized faces, and the rows represent the estimated ages of the synthesized faces which are used as the ground truth. The accuracy of RFA is 65.43% and the accuracy of bi-layer RNN is 61.00% indicating that the RFA performs slightly better than the bi-layer RNN.

4.4.4 Ablation Study

To evaluate the contribution of the face normalization module and the RNN module, we implement the ablation study.

The Contribution of Face Normalization

To evaluate the contribution of face normalization, we use the coarsely aligned faces to do face aging.

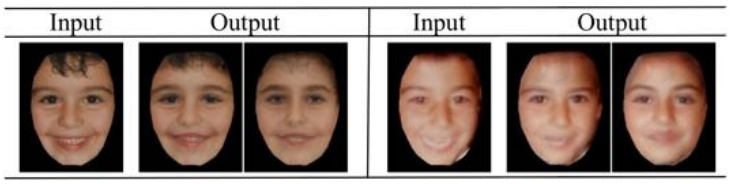


Figure 4.13: Face aging without face normalization.

As shown in Fig. 4.13, if we do face aging without face normalization, the synthesized faces will have some local ghosting artifact. For example, the mouths of the people in Fig. 4.13 are wide open as they are smiling. In the aging process, the mouths are always very blurry as the faces are not well normalized. Therefore, face normalization plays a very important role in face aging.

We have also compared our normalization results with other baselines. As shown in Fig. 4.14, our normalization method can output more natural looking images compared with the baseline HPEN Zhu et al. [2015b]. We can also observe that the eyeballs of the man in Fig. 4.14 can

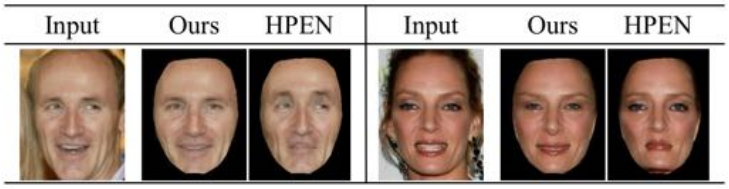


Figure 4.14: Comparison of different face normalization methods.

be normalized to the center of the eyes while HPEN fails aligning the eyeballs to the center. Besides, the lips of the outputs of HPEN are a little bit distorted.

To evaluate the contribution of the face normalization module quantitatively, we first synthesized the aged face without doing face normalization, and then we paired the synthesized aged face with the ground truth aged face (denoted as RFA-I pairs in Fig. 4.11), and implemented face verification. The face verification result is shown in Fig. 4.11. Table 4.3 shows that the EER of RFA-I is similar to the original pairs. Therefore, we can conclude that the RFA without face normalization brings marginal improvement compared with the original pairs.

The Contribution of RNN to Face Aging

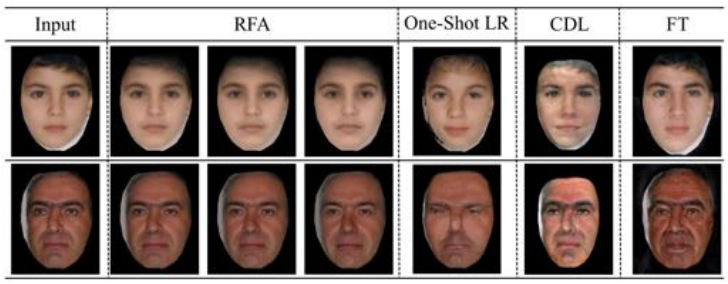


Figure 4.15: Comparison between RFA and other one-shot methods.

To evaluate the contribution of the RNN module, we compare the aging results of RNN and other one-shot method. The results are shown in Fig. 4.15.

For the one-shot method (*e.g.*, linear regression (LR) method), the shape and texture change is too dramatic. This leads to unnatural images with severe artifacts. As shown in Fig. 4.15, we can observe that the output aged face of our RFA method is free of ghosting effects while preserving both the identity information and the characteristics of the aged group. However, if we replace the recurrent process with a one-shot linear regression (LR) model, most of the aged faces have severe artifacts as shown in Fig. 4.15. Similar results can be observed for other one-shot methods. Besides, the FT requires manually aligning the faces while our method can implement this in an automatic way.

To evaluate the contribution of the RNN module quantitatively, we replaced RNN with a simple linear regression model and obtained synthesized aged faces with the linear model. Similarly, we paired the synthesized aged faces with the ground truth aged faces (denoted as RFA-II pairs in Fig. 4.11), and implemented the face verification. From Table 4.3, we observe that the EER of RFA-II is larger than the original pairs. Namely, removing the RNN component of RFA degrades its performance compared with the original pairs. The RNN module has a more significant influence on the performance compared with the face normalization module.

However, we have the constraint that the RNN must be learned pairwise. To adjust the architecture to relax the adjacency constraint of age groups when learning RNNs, we need a single representative encoding method which could reconstruct the images of different ages accurately with the encoded hidden representations. Then we can learn an end-to-end RNN across different age groups based on these hidden variables.

4.5 Conclusion and Future Work

In this paper, we propose a recurrent face aging (RFA) framework which consists of triple-layer GRU. The triple-layer GRU can better preserve the identity information than the bi-layer GRU. With the help of RNNs, the smooth transitional faces between two adjacent groups can also be synthesized. The shape of the face evolves progressively by referring to the autoregressive memory and this leads to a high-quality optical flow. Then the synthesized faces derived from the optical flow are more realistic compared with the one-shot methods. It is worth highlighting that we also propose a progressive face normalization method. By warping the face progressively to the low-rank faces by decreasing the number of eigenfaces, the expressions can be weakened and the profiles can also be calibrated to the frontal faces very well.

To sum up, we exploit a very powerful tripe-layer GRU as our recurrent module. The bottom layer works as an encoder which can encode the image to a high-dimension space and the top layer works as a decoder which decodes the hidden variables to an aged face. The middle layer which has a high dimension is capable of modeling the complex dynamic aging pattern. However, during the testing phase, the system requires the age of the input face which might be unavailable. In the future, we will integrate the age estimation model into our framework.

Chapter 5

Smile Video Generation

5.1 Introduction

Facial expressions are one of the –if not *the*– most prominent non-verbal signals for human communication Vinciarelli et al. [2009]. The automatic recognition of facial expressions has been studied for the last decades Zen et al. [2016]; Park et al. [2015]; Gong et al. [2009]; Zhang et al. [2016]. Indeed, a plethora of discriminative approaches, aiming to learn the boundaries between various categories in different video sequence representation spaces, were proposed to tackle the recognition of facial expressions. Naturally, these approaches focus on recognizing the dynamics of the different signals/expressions. Even if their performance is, specially lately, very impressive, these methods do not posses the ability to reproduce the dynamics of the patterns they accurately classify. How to generate realistic facial expressions is a scientific challenge yet to be soundly addressed.

In particular, we are interested in in enforcing *diversity* when generating facial expressions, for instance, posed *vs*. spontaneous smiles. This is motivated because people never smile twice the same way, and therefore future intelligent agents must be able to automatically generate smiles with differential traits, but still corresponding to the same category (*e.g.* posed). Figure 5.1, displays two image sequences of the same person spontaneously smiling. Importantly, these videos are quite different (*e.g.* closed vs. open eyes and mouth). The underlying research question is, given one single neutral face, *can we generate diverse face expression videos conditioned on a facial expression label?*

Thanks to the proliferation of deep neural architectures, and in particular of generative adversarial networks (GAN) Goodfellow et al. [2014]; Denton et al. [2015] and variational auto-encoders (VAE) Kulkarni et al. [2015], the popularity of image generation techniques has increased in the recent past. Roughly speaking, these methods are able to generate realistic images from encoded representations that are learned in an automatic fashion. Remarkably, the literature on video generation is far less populated and few studies addressing the generation

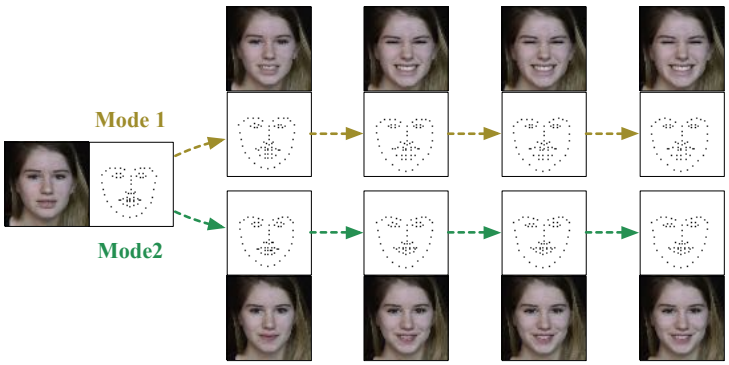


Figure 5.1: Two different sequences of spontaneous smiles and associated landmarks. While there is a common average pattern, the changes from one sequence to another are clearly visible.

of videos Oh et al. [2015b]; Srivastava et al. [2015] or the generation of predicted actions in videos Koppula and Saxena [2016] exist. In this context, it is still unclear how to generate distinct video sequences given a single input image.

The dynamics of facial expressions, and of many other facial (static and dynamic) attributes are encoded in the *facial landmarks*. For instance, it has been shown that landmarks can be used to detect whether a person is smiling spontaneously or in a posed manner Dibeklioğlu et al. [2012]. Action units (e.g. check raiser, upper lip raiser) are also closely related to both facial expressions and facial landmarks King-Smith and Carden [1976]. Therefore, we adopt facial landmarks as a compact representation of the facial dynamics and a good starting point towards our aim. Figure 5.1 shows an example to further motivate the use of landmarks and to illustrate the difficulty of the targeted problem. Indeed, in this figure we can see two examples of spontaneous smiles and their associated landmarks. The differences are small but clear (*e.g.* closed vs. open eyes). Therefore, it is insufficient to learn an "average" spontaneous smiling sequence. We are challenged with the task of learning distinct landmark patterns belonging to the same class. Thus, given a neutral face, the generation of diverse facial expression sequences of a certain class is a *one-to-many* problem.

A technology able to generate different facial expressions of the same class would have a positive impact in different fields. For instance, the face verification and facial expression recognition systems would be more robust to noise and outliers, since there would be more data available for training. In addition, systems based on artificial agents, impersonated by an avatar, would clearly benefit from an expression generation framework able to synthesize distinct image sequences of the same class. Such agents would be able to smile in different ways, as humans do.

In this paper, we propose a novel approach for generating videos of smiling people given an initial image of a neutral face. Specifically, we introduce a methodological framework which

generates various image sequences (i) that correspond to the desired class of expressions (*i.e.* posed or spontaneous smile), (ii) that look realistic and implicitly preserve the identity of the input image and (iii) that have clearly visible differences between them. As previously explained, we exploit facial landmarks since they encode the dynamics of facial expressions in an effective manner. First, a compact representation of the landmark manifold is learned by means of a variational auto-encoder. This representation is further used to learn a conditional recurrent network (LSTM) which takes as input the landmarks automatically extracted from the initial neutral face and generates a sequence of landmark embeddings conditioned on a given facial expression. This sequence is then fed to a multi-mode recurrent landmark generator, which consists of multiple LSTMs and is able to output a set of clearly distinct landmark embedding sequences. Remarkably, the second generating layer does not require additional ground truth to be trained. The input face image is then used for translating the generated landmark embedding sequences into distinct face videos. The joint architecture is named Conditional Multi-Mode (CMM) recurrent network. We evaluate the proposed method on three public datasets: the UvA-NEMO Smile Dibeklioğlu et al. [2012], the DISFA Mavadati et al. [2013] and DISFA+ Mavadati et al. [2016].

Part of this work was previously published at Wei et al. [2018], and this paper contributes with the following items. First, we present a in-depth description of the proposed approach, providing all the architectural and implementation details of the method, with special emphasis on guaranteeing the reproducibility of the experiments. Second, the related work has been significantly expanded, including more recent publications on face image and video generation. Finally, we extend the experimental evaluation provided in several directions: (i) we report the performance of our method on challenging cases, *i.e.* faces wearing glasses, (ii) we provide insights of the effect of the proposed pull-push loss, and (iii) with the help of an external face recognition software, we also evaluate the ability of the proposed model to preserve the identity of the input face, and show that our method successfully preserves the identity of the person in the input image.

The rest of the paper is organized as follows. Section 5.2 briefly reviews the related works on image and video generation. Section 5.3 introduces the architecture of the proposed method. Section 5.4 shows the results of our extensive evaluation on various publicly available datasets. Finally, in Section 5.5 conclusions are drawn.

5.2 Related Work

In this section we review previous works on image and video generation with deep generative models, with special emphasis on recent methods focusing on faces.

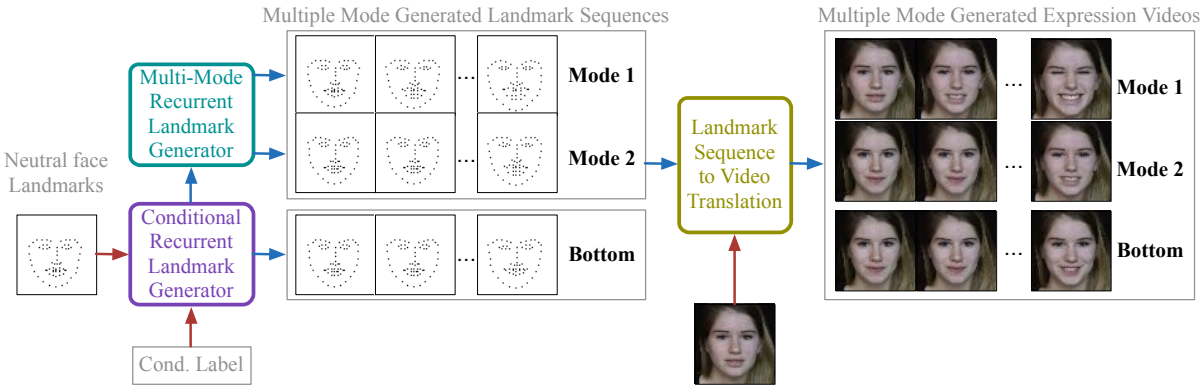


Figure 5.2: Overview of the proposed framework. The input image is used together with the conditioning label to generate a set of K distinct landmark sequences. These landmark sequences guide the neutral face image to translate into face videos.

Image Generation. In the last few years Generative Adversarial Networks (GANs) Goodfellow et al. [2014] and Variational Auto-Encoders (VAE) Kulkarni et al. [2015] have become extremely popular due to their effectiveness in generating visual contents. These models have been exploited in several applications. For instance, GANs, and in particular conditional GANs Mirza and Osindero [2014], have been used to generate images of fashion products Yoo et al. [2016], to modify synthetic images turning them into realistic photos Bousmalis et al. [2017] and for image colorization Isola et al. [2016]. Similarly, VAEs have been adopted for generating images of handwritten digits Salimans et al. [2015], pictures of house numbers Gregor et al. [2015] and for future frame prediction Walker et al. [2016]. Besides works focusing on applications, recent studies have also attempted to improve the original models. For instance, GANs have been extended modifying the basic network architecture and/or the original loss, as in CycleGAN Zhu et al. [2017a], DiscoGAN Kim et al. [2017], and Wasserstein GAN (W-GAN) Arjovsky et al. [2017]. Similarly, many VAE-like models have been introduced, such as Gaussian Mixture VAE Dilokthanakul et al. [2016], Hierarchical VAE Goyal et al. [2017] and VAE-GAN Larsen et al. [2016].

Recent works have considered both GANs and VAEs models for image face generation. For instance, Hou *et al.* Hou et al. [2017b] proposed a VAE equipped with a perceptual loss for synthesizing faces with specific facial expressions. Similarly, Yan *et al.* Yan et al. [2016] addressed the problem of generating face images given some specific attributes (*e.g.* age, gender, expressions, etc). A GAN-based model is proposed in Li et al. [2016] for transferring facial attributes while preserving as much as possible information about identity. Karras *et al.* Karras et al. [2017] introduced a new training methodology for GANs and generated high-resolution face images.

Among previous research studies, the most related to ours are Bulat and Tzimiropoulos

[2017] and Di et al. [2017], as both these works exploit face landmarks. In Bulat and Tzimiropoulos [2017] landmarks are employed for a different task to ours, *i.e.* face super-resolution. In Di et al. [2017] a Gender Preserving Generative Adversarial Network (GP-GAN) is introduced and facial landmarks are exploited for face generation.

However, while all these previous works have considered the problem of generating images, in this paper we explicitly aim to synthesize *face videos* (*e.g.* of smiling people). More importantly, none of these models can solve the one-to-many problem. Given one input signal, these models can only generate one corresponding image.

Video Generation. In the wake of the success achieved in image generation using GANs and VAEs models, researchers have also started to explore deep networks to generate videos Zhou and Berg [2016]; Vondrick et al. [2016]; Saito et al. [2017]; Oh et al. [2015a]. Recent approaches for video generation can be roughly divided in two categories. The first category comprises methods which generate video sequences using spatio-temporal networks, such as to synthesize all the frames simultaneously. For instance, Vondrick *et al.* Vondrick et al. [2016] proposed a 3D deep convolutional generative adversarial networks and exploited this architecture for future frame prediction tasks. Similarly, Saito *et al.* Saito et al. [2017] introduced a Temporal Generative Adversarial Network (TGAN) to generate multiple video frames at the same time. Differently from our approach, these methods are not conceived in order to integrate conditional labels or image priors as input. For instance, TGAN can only generate random videos which looks realistic by sampling random vectors from a Gaussian distribution. Furthermore, these approaches are typically associated to a poor image quality.

The second category of methods models temporal dependencies by taking advantage of recurrent neural networks (RNNs) such as to generate images sequentially. For instance, in Oh et al. [2015a] a convolutional long-short term memory (LSTM) network is used to predict the future frames in Atari games conditioned on an action label. Tulyakov *et al.* Tulyakov et al. [2017] proposed an approach which employs a gated recurrent (GRU) neural network within an adversarial learning framework to generate videos. In these works, the images are first embedded to a manifold space, and the videos can be represented by a trajectory in the manifold space where each embedding point in the trajectory corresponds to one image. However, it is usually very difficult to learn the compact manifold space as the real images are usually very complex. Srivastava *et al.* Srivastava et al. [2015] successfully applied LSTMs for future frame prediction tasks. However, this approach in Srivastava et al. [2015] requires a sufficient amount of input frames (*e.g.*, 10 images) to learn the video dynamics. In Kalchbrenner et al. [2016] Video Pixel Network (VPN), a deep generative model encoding the four-dimensional structure of video tensors, is introduced. However, the methods in Srivastava et al. [2015]; Kalchbrenner et al. [2016] do not integrate any conditioning label. Therefore, these models could not be

directly compared with ours.

Our work belongs to the second category. However, to the best of our knowledge, this is the first study proposing a method able to generate multiple sequences given an input image and a conditioning class label. Current video generation approaches only focus on creating a single sequence and the problem of synthesizing visual contents in a one-to-many setting has only recently been addressed in case of images Ghosh et al. [2017]. Furthermore, different from previous studies (*e.g.* Tulyakov et al. [2017]), we investigate the use of landmark images for face generation. We demonstrate that by operating on landmarks it is possible to better capture the dynamics of facial expressions. This is intuitive, as the landmark manifold space is relatively easier to learn with respect to that associated to the original face images. The benefits of exploiting landmark information to generate smile sequences are demonstrate in the experimental section.

5.3 Cond. Multi-Mode Generation

5.3.1 Overview

The proposed framework to generate facial expression sequences in diversity consists of three blocks (see Fig. 5.2). The input consists of (i) a neutral face image and (ii) a given facial expression class (*e.g.* spontaneous vs. posed smile). The output is a set of K face videos each one containing a different facial expression sequence corresponding to the specified class. First, the conditional recurrent landmark generator (purple box) receives a landmark image computed from the input neutral face, encodes it into a compact representation and generates a landmark sequence corresponding to the desired facial expression class. Second the multi-mode recurrent landmark generator (turquoise box) receives this sequence and generates K sequences of the same class with clearly distinct features. Finally, the landmark sequence to video translation module (ocher box) receives these landmark sequences and the initial neutral face image to produce the output facial expression videos. The entire architecture is named Conditional Multi-Mode recurrent network. In the following we detail the three main blocks.

5.3.2 Conditional Recurrent Landmark Generator

The conditional recurrent landmark generator (magenta box in Fig. 5.4) receives a face image and a conditioning facial expression label as inputs. We automatically extract the landmark image from the face image and encode it using a standard VAE Kingma and Welling [2014] into a compact embedding, denoted as h_0 (details are in Section 5.3.5).

The structure of the conditional recurrent landmark generator is illustrated in Fig. 5.3. The conditional recurrent landmark generator adopts a single layer LSTM which is connected with

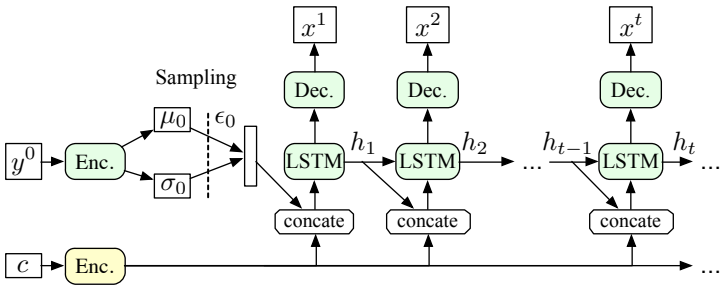


Figure 5.3: Internal structure of the conditional recurrent landmark generator. y^0 denotes the initial input face landmark image with neutral expression. $x^i, (i{=}1, 2, \cdots)$ represents the generated face landmark images. The LSTM is the recurrent unit. At each time step the recurrent unit receives as input the concatenation of h_{t-1} and the embedding of conditioning label c.

an encoder with a re-parameterization layer at the bottom and a decoder on the top (Fig.5.3). The encoder and the decoder blocks are derived from the VAE (see Table 5.1). The input landmark image is encoded to deterministic variables (μ_0, σ_0) which represent a Gaussian distribution. A sampled latent representation is obtained as $\mu_0{+}\sigma_0{*}\epsilon_0$, by sampling ϵ_0 from a standard normal distribution.

The conditional LSTM generates a sequence of T facial landmark embeddings $\mathbf{h} = (h_1, \ldots, h_T)$. The conditional label c is encoded and provided as input at all time steps. The labels are represented by hot vectors in which only one element corresponding to the label is set to 1, while the others are set to 0.

As shown in Fig.5.3, at each step t the input to the recurrent module is the concatenation of h_{t-1} with the embedding of the conditional label. The embedding sequence $\mathbf{h}$ is decoded into a landmark image sequence, $\mathbf{x} = (x_1, \ldots, x_T)$, which is encouraged to be close to the training landmark image sequence $\mathbf{y}$ by minimizing a loss. The loss of the conditional recurrent landmark generator is the sum of two terms, *i.e.* $\mathcal{L} = \mathcal{L}_{\text{BCE}} + \mathcal{L}_{\text{KL}}$. The first loss $\mathcal{L}_{\text{BCE}}$ is defined as the binary cross entropy loss and encourages the generated landmark images x_t^n to be similar to the original ones y_t^n. y_t^n is a binary image, where the pixels corresponding to the landmark positions are set to 1 and the others are set to 0. Given a training set of N sequences of length T, $\{\mathbf{y}^n = (y_1^n, \ldots, y_T^n)\}_{n=1}^N$, the loss $\mathcal{L}_{\text{BCE}}$ writes:

$$\mathcal{L}_{\text{BCE}}{=}\frac{1}{N \cdot T} \sum_{n,t=1}^{N,T} y_t^n \odot \log x_t^n {+}(1{-}y_t^n) \odot \log(1{-}x_t^n), \tag{5.1}$$

where $\odot$ and $\log$ denote the element-wise product and natural logarithm operations respectively.[1] Second, the KL divergence loss $\mathcal{L}_{\text{KL}}$ forces the latent representations to follow a Gaus-

[1]To keep the notation simple, the addition over the pixels in the image is not explicit. In addition, the upper index denotes correspondence to the n-th training sample.

sian distribution. In order to optimize the KL divergence, a re-parameterization layer is implemented on the top of the encoder. Therefore, instead of generating a vector of real values, the encoder will generate a vector of means and a vector of standard deviations. The KL-divergence loss writes:

$$\mathcal{L}_{\text{KL}} = \frac{1}{2} \sum_{j=1}^{J} (\mu_j^2 + \sigma_j^2 - \log((\sigma_j)^2) - 1), \tag{5.2}$$

where J represents the dimension of the latent vector and μ_j and σ_j are the means and standard deviations learned by the encoder.

If one needs to generate face videos of a given class the presented deep network would suffice. However, how could we generate diverse sequences of the same class given one single input image? First, this would require recording several times the "same" facial expression of a person with different patterns, which is particularly difficult for spontaneous facial expressions. Second, a single conditional LSTM does not suffice for generating several distinct sequences: a straightforward training would do nothing else but learn the average landmark sequence. The module described in the next section is specifically designed to overcome these limitations.

5.3.3 Multi-Mode Recurrent Landmark Generator

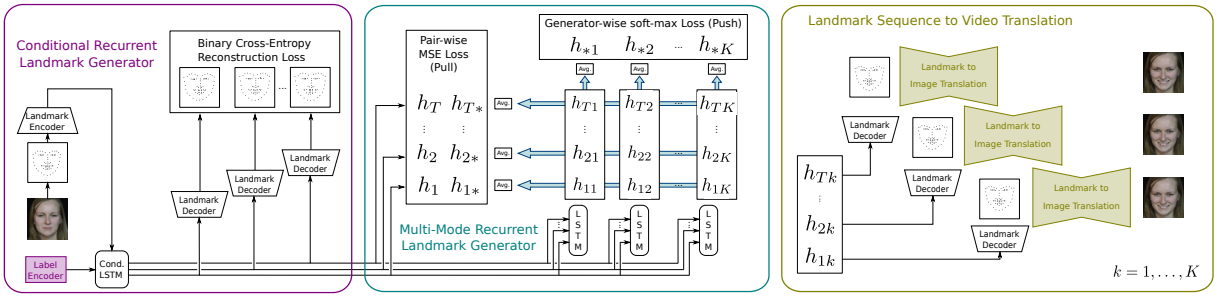


Figure 5.4: Detail of the conditional multi-mode recurrent network. The left block (magenta) encodes the landmark image and generates a sequence of landmark embeddings according to the conditioning label. The second block (turquoise) generates K different landmark embedding sequences. Finally, the third block (ocher) translates each of the sequences into a face video.

As briefly discussed in the previous section, we would like to avoid recording several sequences of the same person, since it may be a tedious process and, more importantly, spontaneous facial expressions are scarce and hard to capture. Ideally, the network module used to generate multiple modes should not require more supervision than the one already needed by the previous module.

We designed the multi-mode recurrent landmark generator (turquoise box of Fig. 5.4) on these grounds. It consists of K LSTMs, whose input is the sequence of embeddings gener-

ated by the conditional LSTM: $h_1, \ldots, h_T$ and the output is a set of K generated sequences $\{\mathbf{h}_k{=}(h_{1k}, \ldots, h_{Tk})\}_{k=1}^{K}$. In a nutshell, this is a one-to-many sequence mapping that has to be learned in an unsupervised fashion. On the one side, we would like the sequences to exhibit clearly distinct features. On the other side, the sequences must encode the desired facial expression. Intuitively, the method finds an optimal trade-off between *pushing* the sequences to be distinct and *pulling* them towards a common pattern. While the differentiating characteristics can happen at various instants in time, the common pattern must respect the dynamics of the smile. This is why, as formalized in the following, the pushing happens over the temporally-averaged sequences while the pulling is used on the mode/generator-wise averages.

Formally, we define $(h_{1*}, \ldots, h_{T*})$ as the sequence of mode-wise averaged generated landmark encodings (horizontal turquoise arrows) and $\{h_{*k}\}_{k=1}^{K}$ as the set of temporally-averaged landmark embedding sequences. With this notation, and following the intuition described in the previous paragraph the push-pull loss is defined as follows. First, we impose a mean squared error loss between the generator-wise average $(h_{1*}, \ldots, h_{T*})$ and the sequence generated by conditional LSTM $(h_1, \ldots, h_T)$:

$$\mathcal{L}_{\text{Pull}} = \sum_{n,t=1}^{N,T} \|h_t^n - h_{t*}^n\|_2. \tag{5.3}$$

Second, inspired by the multi-agent diverse GAN Ghosh et al. [2017], we use the cross-entropy loss so as to discriminate between the sequences obtained from the K generators:

$$\mathcal{L}_{\text{Push}} = -\sum_{n,k=1}^{N,K} \log \phi_k(h_{*k}^n), \tag{5.4}$$

where ϕ_k represents the k-th output of the discriminator (a fully connected layer followed by a soft-max layer). Therefore, the overall architecture is GAN-flavored in the sense that the hierarchical LSTMs are topped with a discriminator to differentiate between the various generators. Importantly, this discriminative loss is complementary to the BCE. The entire loss pushes the multiple sequences far away from each other while encouraging the overall system to behave accordingly to the training data. In GAN, the generator and discriminator compete with each other. Conversely, they work cooperatively in our module.

The K LSTM networks of the multi-mode recurrent landmark generator are distinct and do not share any parameters. In addition to the loss from the conditional recurrent landmark generator, a push $\mathcal{L}_{\text{push}}$ and a pull $\mathcal{L}_{\text{pull}}$ losses are combined in a weighted sum:

$$\mathcal{L}_{\text{push}} + \gamma \mathcal{L}_{\text{pull}}, \tag{5.5}$$

where γ is a user-defined parameter. Here γ controls the trade-off between *pushing* the sequences to be distinct and *pulling* them towards a common pattern. If γ is too big, the differences between the sequences will be hardly noticeable and all the sequences will look alike. On

the other side, if γ is too small, the sequences will have too much flexibility and do not follow the common pattern. In our experiments (Section 5.4) we observe that γ=1 is a good trade-off between generating diversity and keeping a common pattern. We also set K equal to 3, which means we employ three LSTM branches on the top of the shared conditional LSTM. As discussed, the K LSTMs are distinct from each other, *i.e.* they have different parameters, but they share the same input, *i.e.* the landmark embeddings generated by the conditional LSTM. For these LSTMs, both the input and the output vectors in each recurrence have the same length.

We remark that the combination of the conditional and multi-mode landmark recurrent generators has several advantages. First, as already discussed, the multi-mode generator does not require more ground truth than the conditional one. Second, thanks to the push-pull loss, the generated sequences are pushed to be diverse while pulled to stay around a common pattern. Third, while the conditional block is, by definition, conditioned by the label, the second block is transparent to the input label. This is important on one hand because we do not have a specific multi-mode recurrent landmark generator per conditional label, thus reducing the number of network parameters and the amount of data needed for training. On the other hand, because by training the multi-mode generator with data associated to different class labels, it will focus on facial attributes that are not closely correlated with the conditioning labels, and one can expect a certain generalization ability when a new facial expression is added in the system.

5.3.4 Landmark Sequence to Video Translation

The last module of the architecture is responsible for generating the face videos, *i.e.* for translating the facial landmark embeddings generated by the two first modules into image sequences. In this module we translate the landmark embeddings generated by the previous block using the landmark decoder described in Section 5.3.2. Next, after the facial landmark image decoder, we employ the U-Net like structure Isola et al. [2016] (see Fig.5.5 and Table 5.2) trained in an adversarial setting to convert the landmark images to real face images given the initial conditioning neutral face.

Let z_0^n denote the input neutral face image associated to the n-th training sequence. Together with the facial landmark images $\{\mathbf{y}^n = (y_1^n, \ldots, y_T^n)\}_{n=1}^N$ already used to train the previous modules, the dataset contains the face images (from which the facial landmarks are annotated) denoted by $\{\mathbf{z}^n = (z_1^n, \ldots, z_T^n)\}_{n=1}^N$.

In order to train the translation module we employ a combination of a reconstruction loss and an adversarial loss, since we want the generated images to be *locally* close to the ground-truth and to be *globally* realistic. Let $w_t^n(\theta_\mathcal{G}) = \mathcal{G}(y_t^n, z_0^n; \theta_\mathcal{G})$ denote the face image generated with the facial landmark image y_t^n and the neutral face image z_0^n, with parameters $\theta_\mathcal{G}$. The

reconstruction loss writes:

$$\mathcal{L}_{\text{Rec}} = \sum_{n,t=1}^{N,T} \|z_t^n - w_t^n(\theta_{\mathcal{G}})\|_1. \tag{5.6}$$

The adversarial loss is defined over real $[z_0^n, z_t^n]$ and generated $[z_0^n, w_t^n]$ image pairs:

$$\begin{aligned}\mathcal{L}_{\text{Adv}} = &\sum_{n,t=1}^{N,T} \log \mathcal{D}([z_0^n, z_t^n]; \theta_{\mathcal{D}}) \\ &+ \sum_{n,t=1}^{N,T} \log(1 - \mathcal{D}([z_0^n, w_t^n(\theta_{\mathcal{G}})]; \theta_{\mathcal{D}})),\end{aligned} \tag{5.7}$$

where $\theta_{\mathcal{D}}$ are the parameters of the discriminator $\mathcal{D}$. When the generator is fixed, the discriminator is trained to maximize (5.7). When the discriminator is fixed, the generator is trained to jointly minimize the adversarial and reconstruction losses with respect to $\theta_{\mathcal{G}}$:

$$\sum_{n,t=1}^{N,T} \|z_t^n - w_t^n(\theta_{\mathcal{G}})\|_1 + \log(1 - \mathcal{D}([z_0^n, w_t^n(\theta_{\mathcal{G}})]; \theta_{\mathcal{D}})) \tag{5.8}$$

Table 5.1: Structure of the encoder and decoder networks of the conditional recurrent landmark generator

	ENCODER						DECODER				
Layer	Conv1	Conv2	Conv3	Conv4	Conv μ	Conv σ	ConvT5	ConvT4	ConvT3	ConvT2	ConvT1
Kernel size	4x4	4x4	4x4	4x4	4x4	4x4	4x4	4x4	4x4	4x4	4x4
Stride, padding	2,1	2,1	2,1	2,1	1,0	1,0	1,0	2,1	2,1	2,1	2,1
Channels (input,output)	1,64	64,128	128,256	256,512	512,100	512,100	100,512	512,256	256,128	128,64	64,1
Map size ratio	1/2	1/4	1/8	1/16	1/64	1/64	1/16	1/8	1/4	1/2	1
Appending layers	LReLU	BN LReLU	BN LReLU	BN LReLU	-	-	BN LRelu	BN LReLU	BN LReLU	BN LReLU	Sigmoid

Furthermore, inspired by Yoo et al. [2016], we use the adversarial loss at the pixel-level of the feature map. In other words, there is one label per pixel of the coarsest feature map, instead of one label per image.

5.3.5 Implementation Details

In this section we discuss all the implementation and network architecture details of the proposed model.

Table 5.2: Structure of the generator in the landmark sequence to video translation network.

	ENCODER						DECODER					
Layer	Conv1	Conv2	Conv3	Conv4	Conv5	Conv6	ConvT6	ConvT5	ConvT4	ConvT3	ConvT2	ConvT1
Kernel size	4x4	4x4	4x4	4x4	4x4	4x4	4x4	4x4	4x4	4x4	4x4	4x4
Stride, padding	2,1	2,1	2,1	2,1	2,1	2,1	2,1	2,1	2,1	2,1	2,1	2,1
Channels (input,output)	4,64	64,128	128,256	256,512	512,512	512,512	512,512	1024,512	1024,256	512,128	256,64	128,3
Map size ratio	1/2	1/4	1/8	1/16	1/32	1/64	1/32	1/16	1/8	1/4	1/2	1
Appending layers	LReLU	BN LReLU	BN LReLU	BN LReLU	BN LReLU	Relu	BN ReLU	BN Relu	BN ReLU	BN ReLU	BN ReLU	Tanh

Encoder-Decoder of Face Landmark Images As shown in Fig. 5.4, in the first module, *i.e.* the conditional landmark generator, the landmark-image is first encoded by an encoder and then the embeddings are decoded by a decoder to reconstruct the landmark images. The Encoder-Decoder for face landmark images consists of a symmetric convolutional structure with five layers. As shown in Table 5.1, the first four layers of the encoder are Conv(4,2,1) (kernel size, stride and padding) with 64, 128, 256 and 512 output channels respectively. All of them have a Leaky ReLU layer (with slope set to 0.2) and, except for the first one, they use batch normalization. The final layer models the mean and standard deviation of the VAE and are two Conv(4,1,0) layers with 100 output channels each. After the sampling layer, there are the symmetric five convolutional layers with the same parameters as the encoder and 512, 256, 128, 64, and 1 output channels. While the first four layers have a Leaky ReLU layer (with slope set to 0.2) and use batch normalization, the last layer's output is a sigmoid.

Recurrent Landmark Generator As shown in Fig. 5.4, a conditional LSTM is employed as the conditional landmark generator. The input at each recurrence is the concatenation of the encoded label and the hidden embeddings from the previous recurrence as shown in Fig. 5.3. The labels are then encoded using two fully connected layers to obtain the corresponding embeddings. The first fully connected layer has an input and output dimensions of 2 and 20 and it is followed by a ReLU layer. The input and output dimensions of the second fully connected layer are 20 and 100. The dimension of the hidden embeddings is set to 100. The Multi-Mode Recurrent Landmark Generator *i.e.*, the second module in Fig. 5.4 is on the top of the conditional LSTM. The top layer LSTM consists of K branches, each of which correspond to one mode. Both the input and output dimensions of the LSTM of the multi-mode recurrent landmark generator are set to 100.

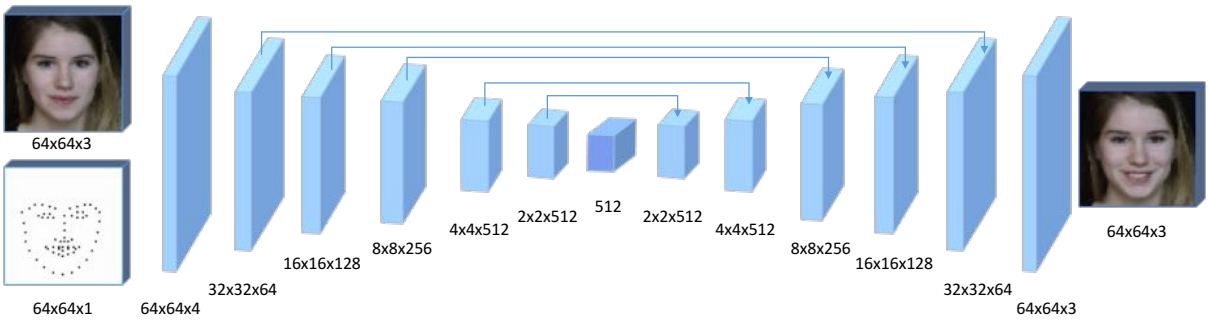


Figure 5.5: Illustration of the generator network used in the landmark sequence to video translation module. The figure reports the feature map size in each layer including the input and output layer. The architecture is symmetric except for the input and output layer. Skip connections are used between the encoder and decoder part.

Table 5.3: Structure of the discriminator in the landmark sequence to video translation network.

Layer	Conv1	Conv2	Conv3	Conv4	Conv5
Kernel size	4x4	4x4	4x4	4x4	4x4
Stride, padding	2,1	2,1	2,1	1,1	1,1
Channels (input,output)	6,64	64,128	128,256	256,512	512,1
Map size ratio	1/2	1/4	1/8	1/8	1/8
Map size	32	16	8	7	6
Appending layers	LRelu	BN LReLU	BN LReLU	BN LReLU	-

Landmark Sequence to Video Translator The translator has an adversarial structure with one generator and one discriminator. The generator of the adversarial translation structure is shown in Table 5.2. The landmark guided image generator has a U-Net structure. U-Net consists of an encoder and a decoder. All ReLus in the encoder is leaky and the slope is set to 0.2 for each of them. The skip connection is built after the batch normalization layers from both the encoder and decoder. For Conv1, the skip connection is added after the convolutional layer directly since it does not have appending batch normalization layer right after it. The pixel value of the images are processed to the range [-1,1]. Therefore, a hyperbolic tangent layer is added at the top of the network to predict the target image.

The generator is a fully convolutional auto-encoder network with 6 Conv(4,2,1) layers with 64, 128, 256, 512, 512 and 512 output channels. The first five convolutional layers use a

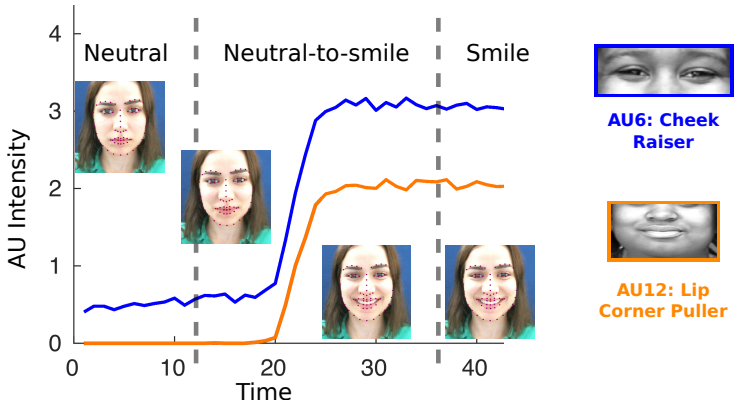


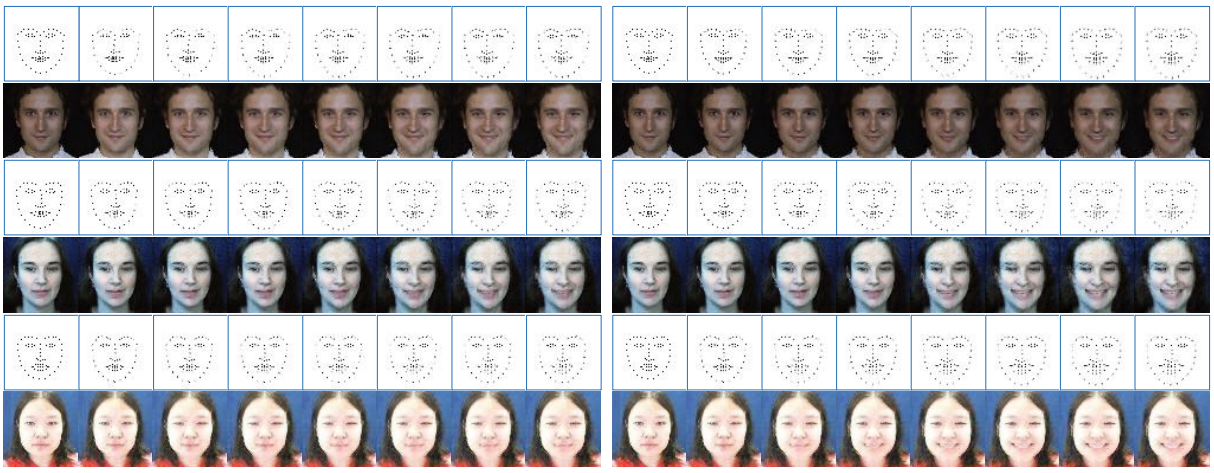
Figure 5.6: Action unit dynamics in neutral-to-smile transitions: *cheek raiser* and *lip corner puller*.

Leaky ReLU, and except for the first, batch normalization. The last layer uses plain ReLU. The decoder has the same structure as the encoder. All layers except the last one use ReLU and batch normalization, and the last one uses a hyperbolic tangent. Notice that the number of input channels is four (neutral face image plus facial landmark image) and the number of output channels is three. The feature maps of the generator network is shown in Fig. 5.5.

As shown in Table 5.3, the discriminator of the adversarial translation structure has three Conv(4,2,1) and two Conv(4,1,1) with 64, 128, 256, 512 and 1 output channels respectively. While all except the last one are followed by a Leaky ReLU, only the three in the middle use batch normalization. Recall that, since the input of the discriminator are image pairs, the input number of channels is six.

5.3.6 Training Strategy

The training of the CMM architecture is done in three phases. First, we train the landmark embedding VAE so as to reconstruct a set of landmark images $\{y_t^n\}_{n,t=1}^{N,T}$. This VAE is trained for 50 epochs before the conditional LSTM is added. The second phase consists on fine-tuning the VAE and training the first layer LSTM on the dataset of sequences of landmark images $\{\mathbf{y}^n\}_{n=1}^{N}$ for 20 epochs. The third stage consists on adding the multi-model recurrent landmark generator. Therefore the VAE and LSTM are fine tuned at the same time the K different LSTMs are learned from scratch. This phase includes the reconstruction, pull and push loss functions previously defined and lasts 10 epochs. Finally, the landmark sequence to video translation module is trained apart from the rest for 20 epochs.



(a) Spontaneous Smile (b) Posed Smile

Figure 5.7: Landmark sequences generated with the first block of our CMM-Net. The associated face images are obtained using the landmark sequence to video translation block. The left block corresponds to generated spontaneous smiles, while the right block to posed smiles. The three row pairs correspond to the UvA-NEMO, DISFA & DISFA+ datasets respectively. Images better seen at magnification.

5.4 Experimental Validation

In this section we first describe the experimental setup and the datasets used in our evaluation and then we provide several qualitative and quantitative results, comparing our approach with previous video generation methods.

5.4.1 Experimental Setup

Datasets

To demonstrate the effectiveness of the proposed approach we perform experiments on three publicly available datasets, namely the UvA-NEMO Smile Dibeklioğlu et al. [2012], the DISFA Mavadati et al. [2013] and the DISFA+ Mavadati et al. [2016] databases.

The UvA-NEMO dataset Dibeklioğlu et al. [2012] contains 1240 videos, 643 corresponding to posed smiles and 597 to spontaneous ones. The dataset comprises 400 subjects (215 male and 185 female) with different ages ranging form 8 to 76 (50 subjects wear glasses). The videos are sampled at 50 FPS and frames have a resolution of 1920×1080 pixels, with an average duration of 3.9 s. The beginning and the end of each video corresponds to a neutral expression.

The DISFA dataset Mavadati et al. [2013] contains videos with spontaneous facial expressions. In the dataset there are 27 adult subjects (12 females and 15 males) with different ethnicities. The videos are recorded at 20 FPS and the image resolution is 1024×768 pixels.

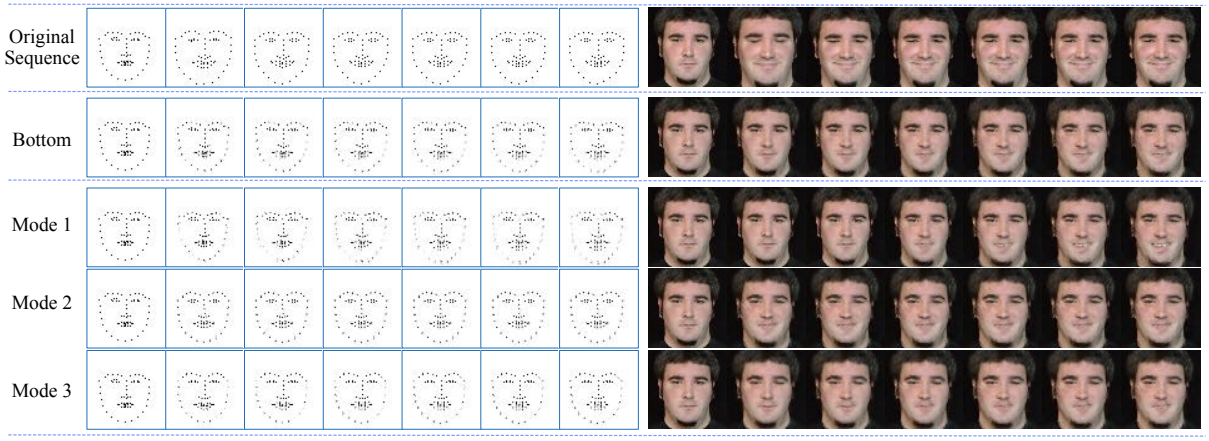


Figure 5.8: Multi-mode generation example with a sequence of the UvA-NEMO dataset: landmarks (left) and associated face images (right). The rows correspond to the original sequence, output of the Conditional LSTM, and output of the Multi-Mode LSTM (last three rows).

While video sequences in the dataset depict people with several facial expressions, in this work we only consider smile sequences and manually segmented the videos to isolate spontaneous smiles, obtaining 17 videos in total. To gather the associated posed smiles, we also consider the DISFA+ dataset Mavadati et al. [2016] which contains posed smile expression sequences for nine individuals present in the DISFA dataset.

Preprocessing

The proposed CMM-Net framework requires training sequences of both posed and spontaneous smiles, as well as the associated landmarks. To collect the training data we process the video sequences from the original datasets and extract the subsequences associated to smile patterns. To automatically segment videos, we rely on Action Units (AUs) Tian et al. [2001] and we extract the intensity variations of two AUs (specifically on the *cheek raiser* and *lip corner puller* AUs) using the method in Baltrušaitis et al. [2015]. We choose the *cheek raiser* and *lip corner puller* AUs since their intensity variations are very characteristic of the neutral-to-smile transition (see Fig. 5.6).

We also perform face alignment on the extracted face sequences using OpenFace Baltrušaitis et al. [2016], considering the center of the two eyes horizontally and the vertical line passing through the center of the two eyes and the mouth. Our approach requires training sequences with fixed length T. Therefore, to create our training videos we perform a preliminary analysis on the original data and we observe that in the considered datasets, the average length of the neutral-to-smile transition is $T{=}32$ frames, with tiny variations. Therefore, to obtain our training data we sample T frames from the first phase of each video. If the number of video frames is less than

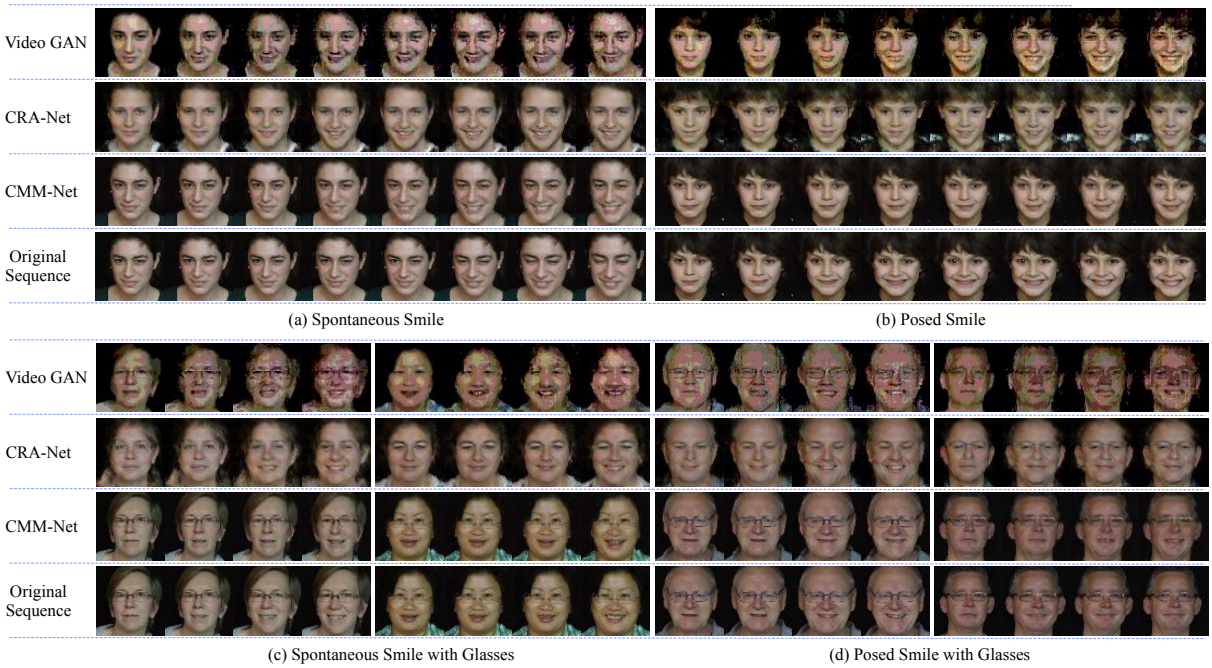


Figure 5.9: Qualitative comparison. From top to bottom: original sequence, Video-GAN, CRA-Net and CMM-Net. Video-GAN introduces many artifacts compared to the other two. CRA-Net learn the smile dynamics, but fail to preserve the identity, as opposed to CMM-Net which produces realistic smiling image sequences.

T, we pad the sequence with subsequent frames. To reduce the computational cost required for training the proposed model we resize images to 64×64 pixels. We also process face images and compute facial landmarks using Baltrusaitis et al. [2013]. Then we convert facial landmarks into 64×64 binary images and consider only those images corresponding to the extracted $T = 32$ frames of the neutral-to-smile transition.

For our evaluation we split the datasets into training and test set. In case of the UvA-NEMO dataset we follow the splitting protocol of Dibeklioğlu et al. [2012] and use nine splits for training and the 10-th for the test. For the paired DISFA-DISFA+ sequences, we randomly select two thirds of the sequences for training and we use the rest for testing.

Baselines

Despite recent efforts, the literature on data-driven automatic video generation is still on its infancy and no previous works have considered the problem of smile generation. Therefore, we do not have direct methods to compare with. However, in order to evaluate the proposed approach we compare with the Video-GAN model Vondrick et al. [2016], even if it has not been specifically designed for face videos. Importantly, since one of the motivations of the present

study is to demonstrate the importance of using facial landmarks, we also compare to a variant of the proposed approach that learns an embedding from the face images directly, instead from landmark images, and we call it conditional recurrent adversarial network (CRA-Net). The CRA-Net has the same structure as the bottom layer conditional recurrent landmark generator. The difference is that a discriminator is added on the top of the generated images to improve the image quality. Other approaches for video generation as discussed in Section 5.2 cannot be compared as they do not allow to provide as input both an image and a conditioning label.

5.4.2 Qualitative Evaluation

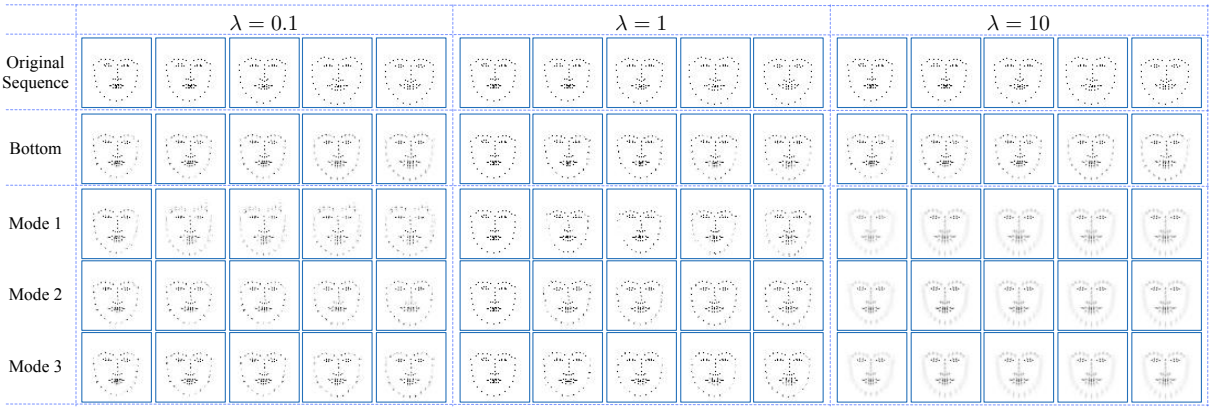


Figure 5.10: The generated landmark sequences *w.r.t.* different λ which is in charge of balancing the trade-off between the push-pull loss.

In a first series of experiments we conduct a qualitative analysis of the video sequences obtained with the proposed method. In particular, we first demonstrate that our Conditional Recurrent Landmark Generator is able to synthesize landmark sequences corresponding to different conditioning labels. Figure 5.7 shows the landmark images obtained for the same neutral face and different conditioning labels (*i.e.* spontaneous/posed). From these results, it is clear that the generated landmarks (and associated face images) follow different dynamics depending on the conditioning label.

To demonstrate the effectiveness of the proposed Multi-Mode Recurrent Landmark Generator block, we also show the results associated to generating multiple landmark sequences with different styles. In this experiment we set $K = 3$. Given a neutral face, the associated landmark image and the conditioning label, we can obtain 4 landmark sequences: the first is obtained from the Conditional LSTM, while the others are generated through the K LSTMs corresponding to different styles. An example of the generated landmark sequences for a posed smile is shown in Fig. 5.8, together with the associated images recovered using the translation block. Our results show that the landmark sequence generated by the Conditional LSTM is very sim-

Table 5.4: Quantitative Analysis. The SSIM and Inception Score.

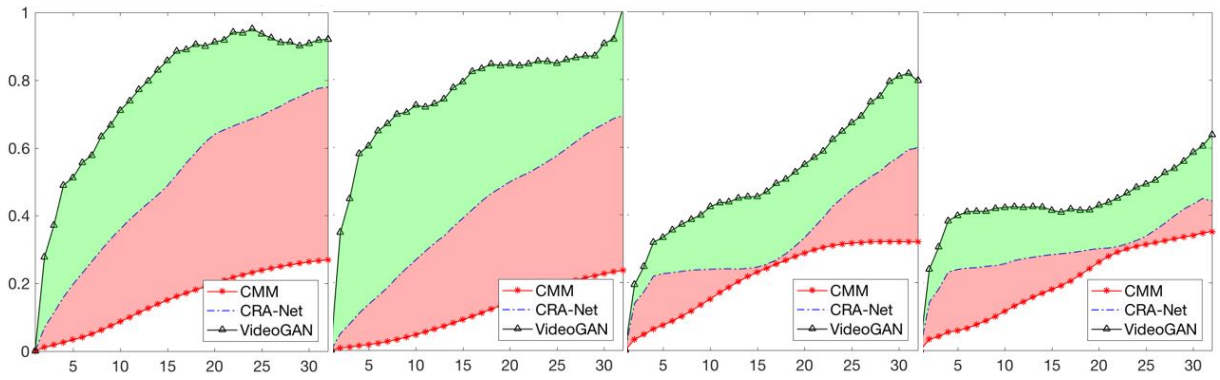
	UvA-NEMO Spont.			UvA-NEMO Posed			DISFA Spont.			DISFA+ Posed		
Model	IS	ΔIS	SSIM	IS	ΔIS	SSIM	IS	ΔIS	SSIM	IS	ΔIS	SSIM
Original	1.419	-	-	1.437	-	-	1.426	-	-	1.595	-	-
Video GAN	1.576	0.157	0.466	1.499	0.062	0.450	1.777	0.351	0.243	1.547	0.048	0.434
CRA-Net	1.311	0.108	0.553	1.310	0.127	0.471	1.833	0.407	0.749	1.534	0.061	0.839
CMM-Net	1.354	0.065	0.854	1.435	0.002	0.827	1.447	0.021	0.747	1.533	0.062	0.810

ilar to the original sequence. Moreover, the landmark images corresponding to multiple styles exhibit clearly distinct patterns, *e.g.* the subject smiles with a wide open mouth (3^{rd} row), with mouth closed (4^{th} row) and with closed eyes (5^{th} row).

Figure 5.9 reports generated sequences of different methods, to benchmark them with the proposed CMM-Net. The first row shows results obtained with Video-GAN Vondrick et al. [2016], the second row corresponds to CRA-Net, the third row is obtained with the proposed CMM-Net, and the fourth row is the original image sequence. From the results, we can observe that the images generated by Video-GAN contain much more artifacts than the other two methods. The images of CRA-Net are quite realistic, meaning that even without learning the landmark manifold space the dynamics of the smile is somehow captured. However, we can clearly see that the identity of the person is not well preserved, and therefore the sequences look unrealistic. The CMM-Net decouples the person identity from the smile dynamics (thanks to the translation and the recurrent blocks, respectively), and thus is able to generate smooth smiling sequences that preserve the identity of the original face. Indeed, as shown in Fig. 5.5, U-Net generates the target face by referring to the initial input face and the target landmark image. Conversely, CRA-Net does not exploit the input neutral image, and therefore it is more prone to identity losses.

To investigate the performance of our approach in more challenging situations, *e.g.* when the person in the input face image is wearing glasses, we also report additional results. Figure 5.9 (c) and (d) show some examples. From the figure it is evident that the glasses are well preserved when using the proposed CMM-Net, while they are missing and/or artifacts appear in the sequences generated by the other methods. The ability to preserve details as the glasses is attributed to the design of the proposed approach which benefit from an accurate video translator module. Indeed, thanks to the skip connections between the encoder and decoder (see Figure 5.5), the overall method exploits features of the input face in the generation of the target face. In this way, the textures and basic level features of the input face are preserved in the target face.

Before analyzing the results of a quantitative evaluation of the proposed method, we report



(a) UvA-NEMO Spontaneous Smile (b) UvA-NEMO Posed Smile (c) DISFA Spontaneous Smile (d) DISFA+ Posed Smile

Figure 5.11: The distance between the first frame and all the other frames (including the first frame) in the video. x axis denotes the index of the frames and y axis represents the action unit score.

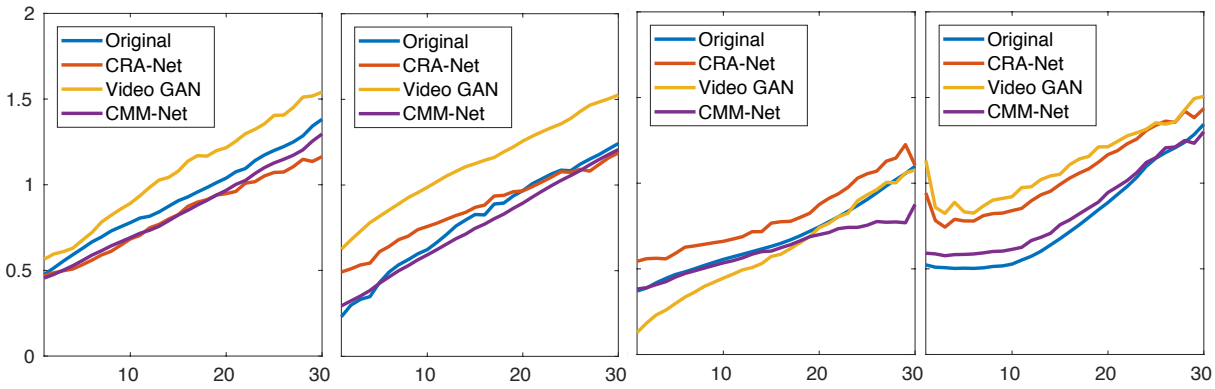
some results to further understand the importance of the proposed push-pull loss. Indeed, while the reconstruction, KL divergence and adversarial losses are well studied in the literature, we are interested in understanding the impact of the novel push-pull loss for multi-mode generation, and in particular to perform a sensitivity analysis of the parameter γ. To this aim, Figure 5.10 shows the generated landmark sequences when γ is set to different values. We can observe that when γ is set to 0.1, the generated landmark sequences with different modes look distinct from each other. However, we can also observe that sometimes these landmark sequences do not follow the common smile patterns. For instance, when $\gamma = 0.1$, the generated landmark images in mode 1 have smile faces in the right beginning with raising brows and wide-open mouths. However, the smile faces should emerge in the very last of the sequence, and the in-between translating landmark images from neutral to smile are missing. When $\gamma = 10$, the pull loss plays a dominant role. Consequently, the generated landmark sequences look very similar to each other. Moreover, the generated landmark images look a little bit blurry.

5.4.3 Quantitative Analysis

To complement the qualitative analysis of the proposed CMM-Nets, we conduct a quantitative evaluation computing different objective measures of the reconstruction quality, performing a user study and measuring the dynamics of the action units in the generated sequences.

Structure Similarity and Inception Scores

Structure similarity (SSIM) Wang et al. [2004] and inception score (IS) Salimans et al. [2016] have been widely employed to measure the quality of the images synthesized with deep generative models. Therefore we also consider these metrics in our evaluation. Table 5.4 reports



(a) UvA-NEMO Spontaneous Smile (b) UvA-NEMO Posed Smile (c) DISFA Spontaneous Smile (d) DISFA+ Posed Smile

Figure 5.12: Dynamics of the action units in neutral-to-smile sequences. x axis denotes the index of the frames. y axis represents the intensity of the action unit.

the two scores for the benchmarked methods. The interpretation of the results in the table must be done with care. Usually, and specially for SSIM, larger image similarity score corresponds to more realistic images. However, high quality images do not always correspond to large IS scores, as observed in Ma et al. [2017]; Shi et al. [2016]. Indeed, a generative model could collapse towards low-quality images with large inception score. This effect is also observed in our experiments if we put Table 5.4 and Figure 5.9 side to side. This is why we also report the score difference between the generated sequence and the original sequence as ΔIS. Intuitively, the smaller this difference is, the more similar is the quality of the generated images to the quality of the original images. Overall, CMM-Net have the higher SSIM score and the smallest difference in IS score. Indeed, while the difference with respect to CRA-Net is hardly noticeable in SSIM for DISFA and DISFA+ as well as in ΔIS for DISFA+, the advantage of using CMM-Net is evident in ΔIS for DISFA and for both measures over UvA-NEMO. As a general observation, we would like to remark that, while SSIM and IS have been reported since they are standard measures for assessing the performance of deep generative models, they do not reflect the identity preservation through the sequence. Therefore we believe that the results of the user study provided in Subsection 5.4.3 are more meaningful for comparing different methods, as users are very sensitive to changes in identity.

Identity Preservation Verification

To evaluate whether the identity is preserved over the video frames, we relied on a pre-trained face recognition model *i.e.*, faceNet Schroff et al. [2015]. This model is trained using a triplet loss, encouraging the face embeddings corresponding to the same identity to lie close by while imposing the face embeddings associated to different identities to be far away from each other.

Table 5.5: CMM-Net vs Video-GAN and CMM-Net vs CRA-Net: percentage (%) of the preferences of the generated videos.

Models		Spontaneous Smile	Posed Smile
Video-GAN Vondrick et al. [2016]		10.14	7.24
CMM-Net		85.14	83.68
∼		4.72	9.08
CRA-Net		17.76	11.94
CMM-Net		54.87	59.72
∼		27.37	28.33

The distance between the face embeddings is employed for face recognition. In our experiments we utilize the distance of the generated video frames with respect to the input image to verify whether the face identity is preserved, and to which extent. Small distances mean that the method preserves the identity.

For each generated spontaneous smile video of the UvA-NEMO dataset, we first computed the distance between the faceNet embeddings of the first frame and other frames. Next, we calculate the average distance and plotted the curve in Fig. 5.11 (a). Similarly, we can obtain the average distance curve of all the posed smile videos, as shown in Fig. 5.11 (b). We also did the same experiments for the videos in the DISFA and DISFA+ datasets and the results are shown in Fig. 5.11 (c) and (d). As shown in Fig. 5.11, the distance between first frame and generated frames grows almost monotonically with respect to the index of the generated frames, meaning that the generated faces drive apart from its identity with time (for all the methods). We also observe that the Video-GAN loses the identity much faster than CRA-Net, and that the proposed CMM-Net model exhibits the least drift between the original image and the generated images. This confirms the fact that our landmark to video translator helps preserving the identity information. The underlying reason is that when we generate the smile face, we always refer to the initial input neutral face. Conversely, the CRA-Net does not have such mechanism. Besides, CRA-Net learns smile patterns using the face embeddings directly, therefore propagating the error from the first frames to the rest of the frames.

User Study

To further demonstrate the validity of the proposed framework, we perform a user-study and compare the videos generated by CMM-Net to the ones generated by Video GAN and CRA-Net. The Video-GAN approach in Vondrick et al. [2016] can only generate videos given an input frame but does not employ conditioning labels. In order to perform a comparison we train two different models corresponding to the two different smiling labels. To compare with

Table 5.6: Distance between the AU curves of different methods and those of the original sequences.

Model	UvA-NEMO Spont.	UvA-NEMO Posed	DISFA	DISFA+
Video GAN	2.976	2.618	3.775	7.979
CRA-Net	4.452	9.783	2.400	9.931
CMM-Net	2.234	1.472	2.035	1.812

each of the baseline, we show a subject a pair of videos (one generated by our CMM-Net and the other by the baseline method) and ask *Which video looks more realistic?*. We prepared 37 video pairs and invited 40 subjects to do the evaluation. We collected 1480 ratings for each of the experiments. Table 5.5 shows the preferences expressed by the annotators (%) both for spontaneous and posed smiles. The symbol $\sim$ indicates that the two videos are rated as similar. When we compare the CMM-Net with the Video GAN baseline (Table 5.5), most annotators prefer the videos generated by our CMM-Net. This is not surprising: by visually inspecting the frames we observe that several artifacts are present in the sequences generated with Video-GAN (see Fig.5.9). Furthermore, comparing our approach with CRA-Net (Table 5.5), we still observe that most annotators prefer images obtained with CMM-Net, confirming the benefit of adopting landmark for face video generation.

Analyzing the Dynamics of AUs

In a final series of experiments we evaluate whether the AUs of the generated data have the same dynamics as the original sequences. In detail, we measure the intensity of the *cheek raiser* AU over the generated sequences using the videos from the testing set, smooth it with a 5-frames long window and plot the average over the test set in Fig. 5.12. We clearly observe that the curves closest to the original data are the ones associated to CMM-Net. This demonstrates the advantage of using a landmark image embedding and proves that the multi-mode image sequences have dynamics that are very similar to the real data. For Video-GAN, the generated videos usually have poor quality making it hard to automatically compute the AU score. Thus, the curves of Video-GAN always significantly deviate from the curve corresponding to the original sequence. Table 5.6 shows the cumulative distance between the AU curves of different models and those corresponding to the original sequences. The values reported in the table further confirm the previous observations.

5.5 Conclusions

We proposed a novel approach for generating multiple video sequences of smiling people. Specifically, we introduced a novel deep architecture which is able to synthesize distinct face

videos of one person given a facial expression (*e.g.* posed vs. spontaneous smile). Differently from previous works on face generation our framework decouples the information about the facial expression dynamics, encoded into landmarks, and the face appearance. For generating landmark sequences we proposed a two layer conditional recurrent network. The first layer net generates a sequence of facial landmark embeddings conditioned on a given facial expression label and an initial face landmark. The second layer is responsible for generating multiple landmark sequences starting from the output of the first layer. The landmark sequences are then translated into face videos adopting a U-Net like architecture. The reported experiments on two publicly available datasets demonstrate the effectiveness of our CMM-Net for generating multiple smiling sequences. Future works will include the design of alternative solutions for the landmark-to-image translation module, as well as the application of the proposed one-to-many generation framework to other problems, such as human pose generation.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, we address human behavior and face analysis problems, which are multi-view action recognition, face alignment, face aging and smile video generation respectively.

- In Chapter 2, we tackle the multi-view action recognition problem by proposing a sparse code filtering (SCF) framework which can mine the action patterns. First, a class-wise sparse coding method is proposed to make the sparse codes of the between-class data lie close by. Then we integrate the classifiers and the class-wise sparse coding process into a collaborative filtering (CF) framework to mine the discriminative sparse codes and classifiers jointly.

- In Chapter 3, we propose a novel Recurrent Convolutional Face Alignment method for face alignment. We frame the standard cascaded alignment problem as a recurrent process and learn all shape increments jointly, by using a recurrent neural network with the gated recurrent unit. Importantly, by combining a convolutional neural network with a recurrent one we alleviate hand-crafted features, widely adopted in the literature and thus allowing the model to learn task-specific features. Moreover, both the convolutional and the recurrent neural networks are learned jointly.

- In Chapter 4, we propose a Recurrent Face Aging (RFA) framework which takes as input a single image and automatically outputs a series of aged faces. The hidden units in the RFA are connected autoregressively allowing the framework to age the person by referring to the previous aged faces. Since human face aging is a smooth progression, it is more appropriate to age the face by going through smooth transitional states.

- In Chapter 5, we introduce a deep neural architecture named conditional multi-mode network (CMM-Net) to generate diverse smile videos of the same person. The smile dynam-

ics are captured thanks to an embedded landmark representations. A conditional recurrent network is used to generate a sequence conditioned to a class label (*e.g.* posed smile). This first sequence is then fed to the multi-mode recurrent landmark generator trained to induce diversity and generate K different sequences of landmark images. Finally, the landmark sequences are translated into video sequences.

To summarize, the contributions of this thesis are as follows:

- We explore several challenge problems, such as multi-view human action recognition, human face alignment, aging and smile video generation.
- We develop several novel architectures, such as the collaborative dictionary learning, recurrent convolutional neural networks, and conditional multi-mode network.
- Our algorithms outperform other state-of-the-art algorithms in multi-view action recognition, face alignment and face aging problems.
- All the proposed algorithms are general framework, potentially applicable to other computer vision and pattern recognition problems.

6.2 Future Works

In the future, we will continue our research with the following possible directions:

- Our models can be extended to other applications. For example, the face alignment method can be also applied for human pose detection, and the smile video generation can be also applied for human action video generation.
- Other methods, *e.g.* Reinforcement Learning (RL) framework can be explored to address these applications, such as face alignment and the smile video generation.

Bibliography

Arjovsky, Martin; Chintala, Soumith, and Bottou, Léon. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223, 2017.

Ashraf, Nazim; Sun, Chuan, and Foroosh, Hassan. View invariant action recognition using projective depth. *Computer Vision and Image Understanding*, 2014.

Asthana, Akshay; Zafeiriou, Stefanos; Cheng, Shiyang, and Pantic, Maja. Robust discriminative response map fitting with constrained local models. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2013.

Auli, Michael; Galley, Michel; Quirk, Chris, and Zweig, Geoffrey. Joint language and translation modeling with recurrent neural networks. In *Conference on Empirical Methods in Natural Language Processing*, 2013.

Baltrusaitis, Tadas; Robinson, Peter, and Morency, Louis Philippe. 3d constrained local model for rigid and non-rigid facial tracking. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2012.

Baltrusaitis, Tadas; Robinson, Peter, and Morency, Louis-Philippe. Constrained local neural fields for robust facial landmark detection in the wild. In *IEEE Int. Conf. on Computer Vision Workshops*, 2013.

Baltrušaitis, Tadas; Mahmoud, Marwa, and Robinson, Peter. Cross-dataset learning and person-specific normalisation for automatic action unit detection. In *IEEE Int. Conf. on Face and Gesture Recognition*, 2015.

Baltrušaitis, Tadas; Robinson, Peter, and Morency, Louis-Philippe. Openface: an open source facial behavior analysis toolkit. In *IEEE Int. Conf. on Winter App. of Computer Vision*, 2016.

Baumann, Florian; Ehlers, Arne; Rosenhahn, Bodo, and Liao, Jie. Recognizing human actions using novel space-time volume binary patterns. *Neurocomputing*, 2016.

Beck, Amir and Teboulle, Marc. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2009.

Belhumeur, Peter N; Jacobs, David W; Kriegman, David J, and Kumar, Narendra. Localizing parts of faces using a consensus of exemplars. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 35(12):2930–2940, 2013.

Bousmalis, Konstantinos; Silberman, Nathan; Dohan, David; Erhan, Dumitru, and Krishnan, Dilip. Unsupervised pixel-level domain adaptation with generative adversarial networks. *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2017.

Bulat, Adrian and Tzimiropoulos, Georgios. Super-fan: Integrated facial landmark localization and super-resolution of real-world low resolution faces in arbitrary poses with gans. *arXiv preprint arXiv:1712.02765*, 2017.

Burgos-Artizzu, Xavier; Perona, Pietro, and Dollár, Piotr. Robust face landmark estimation under occlusion. In *IEEE Int. Conf. on Computer Vision*, 2013.

Cai, Zhuowei; Wang, Limin; Peng, Xiaojiang, and Qiao, Yu. Multi-view super vector for action recognition. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2014.

Cao, Chen; Weng, Yanlin; Lin, Stephen, and Zhou, Kun. 3d shape regression for real-time facial animation. In *SIGGRAPH*, 2013.

Cao, Chen; Hou, Qiming, and Zhou, Kun. Displaced dynamic expression regression for real-time facial tracking and animation. In *SIGGRAPH*, 2014a.

Cao, Xudong. Face alignment by explicit shape regression. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2012.

Cao, Xudong; Wei, Yichen; Wen, Fang, and Sun, Jian. Face alignment by explicit shape regression. *International Journal of Computer Vision*, 107(2):177–190, 2014b.

Chatfield, K.; Simonyan, K.; Vedaldi, A., and Zisserman, A. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, 2014.

Chen, Bor-Chun; Chen, Chu-Song, and Hsu, Winston H. Cross-age reference coding for age-invariant face recognition and retrieval. In *Eur. Conf. on Computer Vision*. 2014.

Cho, Kyunghyun; Van Merriënboer, Bart; Gülçehre, Çağlar; Bahdanau, Dzmitry; Bougares, Fethi; Schwenk, Holger, and Bengio, Yoshua. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing*, 2014.

Cootes, T F and Taylor, C J. Active shape models - 'smart snakes'. In *British Machine Vision Conference*, 1992.

Cootes, T.F.; Edwards, G.J., and Taylor, C.J. Active appearance models. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2001a.

Cootes, Timothy F and Taylor, Chris J. Active shape model search using local grey-level models: A quantitative evaluation. In *British Machine Vision Conference*, 1993.

Cootes, Timothy F; Edwards, Gareth J, and Taylor, Christopher J. Active appearance models. *IEEE Trans. Pattern Analysis and Machine Intelligence*, (6):681–685, 2001b.

Dalal, Navneet and Triggs, Bill. Histograms of oriented gradients for human detection. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2005.

Denton, Emily L; Chintala, Soumith; Fergus, Rob, and others, . Deep generative image models using a laplacian pyramid of adversarial networks. In *Int. Conf. on Neural Information Processing Systems*, 2015.

Di, Xing; Sindagi, Vishwanath A, and Patel, Vishal M. Gp-gan: Gender preserving gan for synthesizing faces from landmarks. *arXiv preprint arXiv:1710.00962*, 2017.

Dibeklioğlu, Hamdi; Salah, Albert, and Gevers, Theo. Are you really smiling at me? spontaneous versus posed enjoyment smiles. *Eur. Conf. on Computer Vision*, 2012.

Dilokthanakul, Nat; Mediano, Pedro AM; Garnelo, Marta; Lee, Matthew CH; Salimbeni, Hugh; Arulkumaran, Kai, and Shanahan, Murray. Deep unsupervised clustering with gaussian mixture variational autoencoders. arXiv preprint arXiv:1611.02648, 2016.

Doll, Piotr; Pietro, Welinder, and Perona, Pietro. Cascaded pose regression. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2010.

Dong, Chao; Loy, Chen Change; He, Kaiming, and Tang, Xiaoou. Learning a deep convolutional network for image super-resolution. In *Eur. Conf. on Computer Vision*. 2014.

Du, Yong; Wang, Wei, and Wang, Liang. Hierarchical recurrent neural network for skeleton based action recognition. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2015.

Fanelli, Gabriele; Dantone, Matthias, and Van Gool, Luc. Real time 3d face alignment with random forests-based active appearance models. In *IEEE Int. Conf. on Face and Gesture Recognition*, 2013.

Farhadi, Ali and Tabrizi, Mostafa Kamali. Learning to recognize activities from the wrong view point. In *Eur. Conf. on Computer Vision*. 2008.

Fu, Yun; Guo, Guodong, and Huang, Thomas S. Age synthesis and estimation via faces: A survey. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 32(11):1955–1976, 2010.

Gers, Felix A and Schmidhuber, Jürgen. Recurrent nets that time and count. In *International Joint Conference on Neural Networks*, 2000.

Ghosh, Arnab; Kulharia, Viveka; Namboodiri, Vinay; Torr, Philip HS, and Dokania, Puneet K. Multi-agent diverse generative adversarial networks. *arXiv preprint arXiv:1704.02906*, 2017.

Goldberg, David; Nichols, David; Oki, Brian M, and Terry, Douglas. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 1992.

Goldberg, Ken; Roeder, Theresa; Gupta, Dhruv, and Perkins, Chris. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 2001.

Golovinskiy, Aleksey; Matusik, Wojciech; Pfister, Hanspeter; Rusinkiewicz, Szymon, and Funkhouser, Thomas. A statistical model for synthesis of detailed facial geometry. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 1025–1034, 2006.

Gong, Boqing; Wang, Yueming; Liu, Jianzhuang, and Tang, Xiaoou. Automatic facial expression recognition on a single 3d face by exploring shape deformation. In *ACM Int. Conf. on Multimedia*, 2009.

Goodfellow, Ian; Pouget-Abadie, Jean; Mirza, Mehdi; Xu, Bing; Warde-Farley, David; Ozair, Sherjil; Courville, Aaron, and Bengio, Yoshua. Generative adversarial nets. In *Int. Conf. on Neural Information Processing Systems*, 2014.

Goyal, Prasoon; Hu, Zhiting; Liang, Xiaodan; Wang, Chenyu, and Xing, Eric. Nonparametric variational auto-encoders for hierarchical representation learning. In *IEEE Int. Conf. on Computer Vision*, 2017.

Graves, Alan; Mohamed, Abdel-rahman, and Hinton, Geoffrey. Speech recognition with deep recurrent neural networks. In *International Conference on Acoustics, Speech, and Signal Processing*, 2013.

Graves, Alex and Schmidhuber, Jürgen. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610, 2005.

Greff, Klaus; Srivastava, Rupesh K; Koutník, Jan; Steunebrink, Bas R, and Schmidhuber, Jürgen. Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10):2222–2232, 2017.

Gregor, Karol; Danihelka, Ivo; Graves, Alex; Rezende, Danilo, and Wierstra, Daan. Draw: A recurrent neural network for image generation. In *International Conference on Machine Learning*, 2015.

Gross, Ralph; Matthews, Iain, and Baker, Simon. Generic vs. person specific active appearance models. *Image and Vision Computing*, 23(12):1080–1093, 2005.

Guha, Tanaya and Ward, Rabab K. Learning sparse representations for human action recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2012.

Hochreiter, Sepp and Schmidhuber, Jürgen. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Hou, Xianxu; Shen, Linlin; Sun, Ke, and Qiu, Guoping. Deep feature consistent variational autoencoder. In *IEEE Int. Conf. on Winter App. of Computer Vision*, 2017a.

Hou, Xianxu; Shen, Linlin; Sun, Ke, and Qiu, Guoping. Deep feature consistent variational autoencoder. In *IEEE Int. Conf. on Winter App. of Computer Vision*, 2017b.

Huang, Chun-Hao; Yeh, Yi-Ren, and Wang, Yu-Chiang Frank. Recognizing actions across cameras by exploring the correlated subspace. In *Eur. Conf. on Computer Vision*, 2012.

Huang, Gary B; Jain, Vidit, and Learned-Miller, Erik. Unsupervised joint alignment of complex images. In *IEEE Int. Conf. on Computer Vision*, 2007a.

Huang, Gary B.; Ramesh, Manu; Berg, Tamara, and Learned-Miller, Erik. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, University of Massachusetts, Amherst, 2007b.

Im, Daniel Jiwoong; Kim, Chris Dongjoo; Jiang, Hui, and Memisevic, Roland. Generating images with recurrent adversarial networks. *arXiv preprint arXiv:1602.05110*, 2016.

Isola, Phillip; Zhu, Jun-Yan; Zhou, Tinghui, and Efros, Alexei A. Image-to-image translation with conditional adversarial networks. arXiv preprint arXiv:1611.07004, 2016.

Jeni, Laszlo A; Cohn, Jeffrey F, and Kanade, Takeo. Dense 3d face alignment from 2d videos in real-time. In *IEEE Int. Conf. on Face and Gesture Recognition*, 2015.

Jourabloo, Amin and Liu, Xiaoming. Pose-invariant 3d face alignment. In *IEEE Int. Conf. on Computer Vision*, 2015.

Jozefowicz, Rafal; Zaremba, Wojciech, and Sutskever, Ilya. An empirical exploration of recurrent network architectures. In *International Conference on Machine Learning*, 2015.

Junejo, Imran N; Dexter, Emilie; Laptev, Ivan, and Pérez, Patrick. *Cross-view action recognition from temporal self-similarities*. Springer, 2008.

Junejo, Imran N; Dexter, Emilie; Laptev, Ivan, and Perez, Patrick. View-independent action recognition from temporal self-similarities. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2011.

Kalchbrenner, Nal; Oord, Aaron van den; Simonyan, Karen; Danihelka, Ivo; Vinyals, Oriol; Graves, Alex, and Kavukcuoglu, Koray. Video pixel networks. *arXiv preprint arXiv:1610.00527*, 2016.

Karpathy, Andrej and Fei-Fei, Li. Deep visual-semantic alignments for generating image descriptions. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2015.

Karras, Tero; Aila, Timo; Laine, Samuli, and Lehtinen, Jaakko. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

Kazemi, Vahid and Josephine, S. One millisecond face alignment with an ensemble of regression trees. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2014.

Kemelmacher-Shlizerman, Ira and Seitz, Steven M. Face reconstruction in the wild. In *IEEE Int. Conf. on Computer Vision*, 2011.

Kemelmacher-Shlizerman, Ira and Seitz, Steven M. Collection flow. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2012.

Kemelmacher-Shlizerman, Ira; Suwajanakorn, Supasorn, and Seitz, Steven M. Illumination-aware age progression. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2014.

Kim, Taeksoo; Cha, Moonsu; Kim, Hyunsoo; Lee, Jungkwon, and Kim, Jiwon. Learning to discover cross-domain relations with generative adversarial networks. arXiv preprint arXiv:1703.05192, 2017.

King-Smith, Peter Ewen and Carden, D. Luminance and opponent-color contributions to visual detection and adaptation and to temporal and spatial integration. 66(7):709–717, 1976.

Kingma, Diederik P and Welling, Max. Auto-encoding variational bayes. *Int. Conf. on Learning Representations*, 2014.

Koppula, Hema S and Saxena, Ashutosh. Anticipating human activities using object affordances for reactive robotic response. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 38(1):14–29, 2016.

Koutnik, Jan; Greff, Klaus; Gomez, Faustino, and Schmidhuber, Juergen. A clockwork rnn. In *International Conference on Machine Learning*, 2014.

Krizhevsky, Alex; Sutskever, Ilya, and Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. In *Int. Conf. on Neural Information Processing Systems*, 2012.

Kulkarni, Tejas D; Whitney, William F; Kohli, Pushmeet, and Tenenbaum, Josh. Deep convolutional inverse graphics network. In *Int. Conf. on Neural Information Processing Systems*, 2015.

Lai, Siwei; Xu, Liheng; Liu, Kang, and Zhao, Jun. Recurrent convolutional neural networks for text classification. In *Association for the Advancement of Artificial Intelligence*, 2015.

Lanitis, Andreas; Taylor, Chris J, and Cootes, Timothy F. Toward automatic simulation of aging effects on face images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(4):442–455, 2002.

Larsen, Anders Boesen Lindbo; Sønderby, Søren Kaae; Larochelle, Hugo, and Winther, Ole. Autoencoding beyond pixels using a learned similarity metric. In *International Conference on Machine Learning*, 2016.

Le, Vuong; Brandt, Jonathan; Lin, Zhe; Bourdev, Lubomir, and Huang, Thomas S. Interactive facial feature localization. In *Eur. Conf. on Computer Vision*. 2012.

Learned-Miller, Erik; Huang, Gary B; RoyChowdhury, Aruni; Li, Haoxiang, and Hua, Gang. Labeled faces in the wild: A survey. In *Advances in Face Detection and Facial Image Analysis*, pages 189–248. Springer, 2016.

Li, Changsheng; Liu, Qingshan; Liu, Jing, and Lu, Hanqing. Learning ordinal discriminative features for age estimation. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2012.

Li, Mu; Zuo, Wangmeng, and Zhang, David. Deep identity-aware transfer of facial attributes. arXiv preprint:1610.05586, 2016.

Li, Ruonan and Zickler, Todd. Discriminative virtual views for cross-view action recognition. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2012.

Liang, Ming and Hu, Xiaolin. Recurrent convolutional neural network for object recognition. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2015.

Liang, Xiaodan; Liu, Si; Shen, Xiaohui; Yang, Jianchao; Liu, Luoqi; Dong, Jian; Lin, Liang, and Yan, Shuicheng. Deep human parsing with active template regression. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 37(12):2402–2414, 2015.

Liu, Ce. *Beyond pixels: exploring new representations and applications for motion analysis*. PhD thesis, Citeseer, 2009.

Liu, Jingen and Shah, Mubarak. Learning human actions via information maximization. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2008.

Lowe, David G. Object recognition from local scale-invariant features. In *IEEE Int. Conf. on Computer Vision*, 1999.

Lowe, David G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2): 91–110, 2004.

Luo, Jiajia; Wang, Wei, and Qi, Hairong. Group sparsity and geometry constrained dictionary learning for action recognition from depth maps. In *IEEE Int. Conf. on Computer Vision*, 2013.

Lv, Fengjun and Nevatia, Ramakant. Single view human action recognition using key pose matching and viterbi path searching. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2007.

Ma, Liqian; Jia, Xu; Sun, Qianru; Schiele, Bernt; Tuytelaars, Tinne, and Van Gool, Luc. Pose guided person image generation. In *Int. Conf. on Neural Information Processing Systems*, 2017.

Mahasseni, Behrooz and Todorovic, Sinisa. Latent multitask learning for view-invariant action recognition. In *IEEE Int. Conf. on Computer Vision*, 2013.

Matikainen, Pyry; Sukthankar, Rahul, and Hebert, Martial. Model recommendation for action recognition. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2012.

Mavadati, Mohammad; Sanger, Peyten, and Mahoor, Mohammad H. Extended disfa dataset: Investigating posed and spontaneous facial expressions. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition Workshops*, 2016.

Mavadati, S Mohammad; Mahoor, Mohammad H; Bartlett, Kevin; Trinh, Philip, and Cohn, Jeffrey F. Disfa: A spontaneous facial action intensity database. *IEEE Trans. on Affective Computing*, 4(2):151–160, 2013.

Messer, Kieron; Matas, Jiri; Kittler, Josef; Luettin, Juergen, and Maitre, Gilbert. Xm2vtsdb: The extended m2vts database. In *Second international conference on audio and video-based biometric person authentication*, 1999.

Mirza, Mehdi and Osindero, Simon. Conditional generative adversarial nets. arXiv preprint:1411.1784, 2014.

Natarajan, Pradeep and Nevatia, Ram. View and scale invariant action recognition using multiview shape-flow models. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2008.

Oh, Junhyuk; Guo, Xiaoxiao; Lee, Honglak; Lewis, Richard L, and Singh, Satinder. Action-Conditional Video Prediction using Deep Networks in Atari Games. In *Int. Conf. on Neural Information Processing Systems*, 2015a.

Oh, Junhyuk; Guo, Xiaoxiao; Lee, Honglak; Lewis, Richard L, and Singh, Satinder. Action-conditional video prediction using deep networks in atari games. In *Int. Conf. on Neural Information Processing Systems*, 2015b.

Park, Sung Yeong; Lee, Seung Ho, and Ro, Yong Man. Subtle facial expression recognition using adaptive magnification of discriminative facial motion. In *ACM Int. Conf. on Multimedia*, 2015.

Park, Unsang; Tong, Yiying, and Jain, Anil K. Face recognition with temporal invariance: A 3d aging model. In *IEEE Int. Conf. on Face and Gesture Recognition*, 2008.

Park, Unsang; Tong, Yiying, and Jain, Anil K. Age-invariant face recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 32(5):947–954, 2010.

Peng, Yigang; Ganesh, Arvind; Wright, John; Xu, Wenli, and Ma, Yi. Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 34(11):2233–2246, 2012.

Peursum, Patrick; Venkatesh, Svetha, and West, Geoff. Tracking-as-recognition for articulated full-body human motion analysis. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2007.

Pinheiro, Pedro H. O. and Collobert, Ronan. Recurrent convolutional neural networks for scene parsing. *arXiv*, 2013.

Qiu, Qiang; Jiang, Zhuolin, and Chellappa, Rama. Sparse dictionary-based representation and recognition of action attributes. In *IEEE Int. Conf. on Computer Vision*, 2011.

Raina, Rajat; Battle, Alexis; Lee, Honglak; Packer, Benjamin, and Ng, Andrew Y. Self-taught learning: transfer learning from unlabeled data. In *International Conference on Machine Learning*, 2007.

Ramanathan, Narayanan and Chellappa, Rama. Modeling shape and textural variations in aging faces. In *IEEE Int. Conf. on Face and Gesture Recognition*, 2008.

Ramanathan, Narayanan; Chellappa, Rama; Biswas, Soma, and others, . Age progression in human faces: A survey. *Journal of Visual Languages and Computing*, 15:3349–3361, 2009.

Reddy, Kishore K; Liu, Jingen, and Shah, Mubarak. Incremental action recognition using feature-tree. In *IEEE Int. Conf. on Computer Vision*, 2009.

Ren, Shaoqing; Cao, Xudong; Wei, Yichen, and Sun, Jian. Face alignment at 3000 fps via regressing local binary features. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2014.

Ricanek Jr, Karl and Tesafaye, Tamirat. Morph: A longitudinal image database of normal adult age-progression. In *IEEE Int. Conf. on Face and Gesture Recognition*, 2006.

Rudovic, Ognjen; Patras, Ioannis, and Pantic, Maja. Coupled gaussian process regression for pose-invariant facial expression recognition. *Eur. Conf. on Computer Vision*, 2010.

Sagonas, Christos; Tzimiropoulos, Georgios; Zafeiriou, Stefanos, and Pantic, Maja. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *IEEE Int. Conf. on Computer Vision Workshops*, 2013.

Saito, Masaki; Matsumoto, Eiichi, and Saito, Shunta. Temporal generative adversarial nets with singular value clipping. In *IEEE Int. Conf. on Computer Vision*, 2017.

Salimans, Tim; Kingma, Diederik P; Welling, Max, and others, . Markov chain monte carlo and variational inference: Bridging the gap. In *International Conference on Machine Learning*, 2015.

Salimans, Tim; Goodfellow, Ian; Zaremba, Wojciech; Cheung, Vicki; Radford, Alec, and Chen, Xi. Improved techniques for training gans. In *Int. Conf. on Neural Information Processing Systems*, pages 2234–2242, 2016.

Saragih, Jason M; Lucey, Simon, and Cohn, Jeffrey F. Deformable model fitting by regularized landmark mean-shift. *International Journal of Computer Vision*, 91(2):200–215, 2011.

Scherbaum, Kristina; Sunkel, Martin; Seidel, H-P, and Blanz, Volker. Prediction of individual non-linear aging trajectories of faces. In *Computer Graphics Forum*, volume 26, pages 285–294. Wiley Online Library, 2007.

Schroff, Florian; Kalenichenko, Dmitry, and Philbin, James. Facenet: A unified embedding for face recognition and clustering. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2015.

Schuster, Mike and Paliwal, Kuldip K. Bidirectional recurrent neural networks. *IEEE Trans. Signal Processing*, 45(11): 2673–2681, 1997.

Shi, Wenzhe; Caballero, Jose; Huszár, Ferenc; Totz, Johannes; Aitken, Andrew P; Bishop, Rob; Rueckert, Daniel, and Wang, Zehan. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 1874–1883, 2016.

Shu, Xiangbo; Tang, Jinhui; Lai, Hanjiang; Liu, Luoqi, and Yan, Shuicheng. Personalized age progression with aging dictionary. In *IEEE Int. Conf. on Computer Vision*, 2015.

Simonyan, Karen and Zisserman, Andrew. Very deep convolutional networks for large-scale image recognition. *arXiv*, 2014.

Smith, Brandon; Brandt, Jonathan; Lin, Zhe, and Zhang, Li. Nonparametric context modeling of local appearance for pose-and expression-robust facial landmark localization. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2014.

Srivastava, Nitish; Mansimov, Elman, and Salakhutdinov, Ruslan. Unsupervised learning of video representations using LSTMs. In *International Conference on Machine Learning*, 2015.

Sun, Chuan; Junejo, Imran, and Foroosh, Hassan. Action recognition using rank-1 approximation of joint self-similarity volume. In *IEEE Int. Conf. on Computer Vision*, 2011.

Sun, Yi; Chen, Yuheng; Wang, Xiaogang, and Tang, Xiaoou. Deep learning face representation by joint identification-verification. In *Int. Conf. on Neural Information Processing Systems*, 2014.

Suo, Jinli; Zhu, Song-Chun; Shan, Shiguang, and Chen, Xilin. A compositional and dynamic model for face aging. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 32(3):385–401, 2010.

Suo, Jinli; Chen, Xilin; Shan, Shiguang; Gao, Wen, and Dai, Qionghai. A concatenational graph evolution aging model. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 34(11):2083–2096, 2012.

Szegedy, Christian; Liu, Wei; Jia, Yangqing; Sermanet, Pierre; Reed, Scott; Anguelov, Dragomir; Erhan, Dumitru; Vanhoucke, Vincent, and Rabinovich, Andrew. Going deeper with convolutions. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2015.

Taigman, Yaniv; Yang, Ming; Ranzato, Marc'Aurelio, and Wolf, Lars. Deepface: Closing the gap to human-level performance in face verification. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2014.

Tazoe, Yusuke; Gohara, Hiroaki; Maejima, Akinobu, and Morishima, Shigeo. Facial aging simulator considering geometry and patch-tiled texture. In *SIGGRAPH*. ACM, 2012.

Tian, Y-I; Kanade, Takeo, and Cohn, Jeffrey F. Recognizing action units for facial expression analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(2):97–115, 2001.

Tiddeman, Bernard; Burt, Michael, and Perrett, David. Prototyping and transforming facial textures for perception research. *Computer Graphics and Applications*, 21(5):42–50, 2001.

Tulyakov, S. and Sebe, N. Regressing a 3d face shape from a single image. In *IEEE Int. Conf. on Computer Vision*, 2015.

Tulyakov, S.; Vieriu, R. L.; Semeniuta, S., and Sebe, N. Robust Real-Time Extreme Head Pose Estimation. In *International Conference on Pattern Recognition*, 2014.

Tulyakov, Sergey; Alameda-Pineda, Xavier; Ricci, Elisa; Yin, Lijun; Cohn, Jeffrey F., and Sebe, Nicu. Self-Adaptive Matrix Completion for Heart Rate Estimation from Face Videos under Realistic Condition. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2016a.

Tulyakov, Sergey; Alameda-Pineda, Xavier; Ricci, Elisa; Yin, Lijun; Cohn, Jeffrey F., and Sebe, Nicu. Self-adaptive matrix completion for heart rate estimation from face videos under realistic conditions. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, June 2016b.

Tulyakov, Sergey; Liu, Ming-Yu; Yang, Xiaodong, and Kautz, Jan. Mocogan: Decomposing motion and content for video generation. *arXiv preprint arXiv:1707.04993*, 2017.

Turk, Matthew A and Pentland, Alex P. Face recognition using eigenfaces. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 1991.

Tzimiropoulos, Georgios. Project-out cascaded regression with an application to face alignment. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2015.

Tzimiropoulos, Georgios and Pantic, Maja. Optimization problems for fast aam fitting in-the-wild. In *IEEE Int. Conf. on Computer Vision*, 2013.

Tzimiropoulos, Georgios and Pantic, Maja. Gauss-newton deformable part models for face alignment in-the-wild. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2014.

Vemulapalli, Raviteja; Arrate, Felipe, and Chellappa, Rama. Human action recognition by representing 3d skeletons as points in a lie group. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2014.

Venugopalan, Subhashini; Rohrbach, Marcus; Donahue, Jeffrey; Mooney, Raymond; Darrell, Trevor, and Saenko, Kate. Sequence to sequence-video to text. In *IEEE Int. Conf. on Computer Vision*, 2015.

Vinciarelli, Alessandro; Pantic, Maja, and Bourlard, Hervé. Social signal processing: Survey of an emerging domain. *Image and Vision Computing*, 27(12):1743–1759, 2009.

Vondrick, Carl; Pirsiavash, Hamed, and Torralba, Antonio. Generating videos with scene dynamics. In *Int. Conf. on Neural Information Processing Systems*, 2016.

Walker, Jacob; Doersch, Carl; Gupta, Abhinav, and Hebert, Martial. An uncertain future: Forecasting from static images using variational autoencoders. In *Eur. Conf. on Computer Vision*, 2016.

Wang, Naiyan and Yeung, Dit-Yan. Learning a deep compact image representation for visual tracking. In *Int. Conf. on Neural Information Processing Systems*, 2013.

Wang, Nannan; Gao, Xinbo; Tao, Dacheng, and Li, Xuelong. Facial feature point detection: A comprehensive survey. *arXiv*, 2014.

Wang, Wei; Yan, Yan, and Sebe, Nicu. Attribute guided dictionary learning. In *ACM International Conference on Multimedia Retrieval*, 2015.

Wang, Wei; Cui, Zhen; Yan, Yan; Feng, Jiashi; Yan, Shuicheng; Shu, Xiangbo, and Sebe, Nicu. Recurrent face aging. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2016a.

Wang, Wei; Cui, Zhen; Yan, Yan; Feng, Jiashi; Yan, Shuicheng; Shu, Xiangbo, and Sebe, Nicu. Recurrent face aging. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2016b.

Wang, Wei; Cui, Zhen; Yan, Yan; Feng, Jiashi; Yan, Shuicheng; Shu, Xiangbo, and Sebe, Nicu. Recurrent face aging. 2016c.

Wang, Wei; Tulyakov, Sergey, and Sebe, Nicu. Recurrent convolutional face alignment. In *Asian Conference on Computer Vision*, 2016d.

Wang, Wei; Yan, Yan; Nie, Liqiang; Zhang, Luming; Winkler, Stefan, and Sebe, Nicu. Sparse code filtering for action pattern mining. In *Asian Conference on Computer Vision*, 2016e.

Wang, Wei; Yan, Yan; Winkler, Stefan, and Sebe, Nicu. Category specific dictionary learning for attribute specific feature selection. *IEEE Trans. Image Processing*, 25(3):1465–1478, 2016f.

Wang, Zhou; Bovik, Alan C; Sheikh, Hamid R, and Simoncelli, Eero P. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Processing*, 13(4):600–612, 2004.

Wei, Wang; Alameda-Pineda, Xavier; Xu, Dan; Fua, Pascal; Ricci, Elisa, and Sebe, Nicu. Every smile is unique: Landmark-guided diverse smile generation. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2018.

Weinland, Daniel; Ronfard, Remi, and Boyer, Edmond. Free viewpoint action recognition using motion history volumes. *Computer Vision and Image Understanding*, 2006.

Weinland, Daniel; Boyer, Edmond, and Ronfard, Remi. Action recognition from arbitrary views using 3d exemplars. In *IEEE Int. Conf. on Computer Vision*, 2007.

Weinland, Daniel; Özuysal, Mustafa, and Fua, Pascal. Making action recognition robust to occlusions and viewpoint changes. In *Eur. Conf. on Computer Vision*, 2010.

Xiao, Huaxin; Wei, Yunchao; Liu, Yu; Zhang, Maojun, and Feng, Jiashi. Transferable semi-supervised semantic segmentation. In *Association for the Advancement of Artificial Intelligence*, 2017.

Xiong, Xuehan and De La Torre, Fernando. Supervised descent method and its applications to face alignment. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2013.

Xiong, Xuehan and De la Torre, Fernando. Supervised descent method and its applications to face alignment. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2013.

Xiong, Xuehan and Torre, Fernando De. Global supervised descent method. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2015.

Yan, Xinchen; Yang, Jimei; Sohn, Kihyuk, and Lee, Honglak. Attribute2image: Conditional image generation from visual attributes. In *Eur. Conf. on Computer Vision*, 2016.

Yan, Yan; Ricci, Elisa; Subramanian, Ramanathan; Liu, Gaowen, and Sebe, Nicu. Multitask linear discriminant analysis for view invariant action recognition. *IEEE Trans. Image Processing*, 2014.

Yang, Heng and Patras, Ioannis. Sieving regression forest votes for facial feature detection in the wild. In *IEEE Int. Conf. on Computer Vision*, 2013.

Yao, Kaisheng; Cohn, Trevor; Vylomova, Katerina; Duh, Kevin, and Dyer, Chris. Depth-gated recurrent neural networks. *arXiv preprint arXiv:1508.03790*, 2015.

Yoo, Donggeun; Kim, Namil; Park, Sunggyun; Paek, Anthony S, and Kweon, In So. Pixel-level domain transfer. In *Eur. Conf. on Computer Vision*, 2016.

Yu, Xiang; Huang, Junzhou; Zhang, Shaoting; Yan, Wang, and Metaxas, Dimitris N. Pose-free facial landmark fitting via optimized part mixtures and cascaded deformable shape model. In *IEEE Int. Conf. on Computer Vision*, 2013.

Zaremba, Wojciech. An empirical exploration of recurrent network architectures. In *International Conference on Machine Learning*, 2015.

Zen, Gloria; Porzi, Lorenzo; Sangineto, Enver; Ricci, Elisa, and Sebe, Nicu. Learning personalized models for facial expression analysis and gesture recognition. *IEEE Trans. Multimedia*, 18(4):775–788, 2016.

Zhang, Feifei; Mao, Qirong; Dong, Ming, and Zhan, Yongzhao. Multi-pose facial expression recognition using transformed dirichlet process. In *ACM Int. Conf. on Multimedia*, 2016.

Zhang, Jie; Shan, Shiguang; Kan, Meina, and Chen, Xilin. Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment. In *Eur. Conf. on Computer Vision*. 2014.

Zhao, Xiaowei; Kim, Tae-Kyun, and Luo, Wenhan. Unified face analysis by iterative multi-output random forests. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2014.

Zheng, Jingjing and Jiang, Zhuolin. Learning view-invariant sparse representations for cross-view action recognition. In *IEEE Int. Conf. on Computer Vision*, 2013.

Zhou, Shaohua Kevin and Comaniciu, Dorin. Shape regression machine. In *Medical Imaging*, 2007.

Zhou, Yipin and Berg, Tamara L. Learning temporal transformations from time-lapse videos. In *Eur. Conf. on Computer Vision*, 2016.

Zhu, Jun-Yan; Park, Taesung; Isola, Phillip, and Efros, Alexei A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE Int. Conf. on Computer Vision*, 2017a.

Zhu, Linchao; Xu, Zhongwen, and Yang, Yi. Bidirectional multirate reconstruction for temporal modeling in videos. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2017b.

Zhu, Linchao; Xu, Zhongwen; Yang, Yi, and Hauptmann, Alexander G. Uncovering the temporal context for video question answering. *International Journal of Computer Vision*, 124(3):409–421, 2017c.

Zhu, Shizhan; Li, Cheng; Change, Chen, and Tang, Xiaoou. Face alignment by coarse-to-fine shape searching. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2015a.

Zhu, Xiangxin and Ramanan, Deva. Face detection, pose estimation, and landmark localization in the wild. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2012.

Zhu, Xiangyu; Lei, Zhen; Yan, Junjie; Yi, Dong, and Li, Stan Z. High-fidelity pose and expression normalization for face recognition in the wild. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2015b.