# LEARNING DISENTANGLED REPRESENTATIONS IN DEEP VISUAL ANALYSIS

by

XI PENG

A Dissertation submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Computer Science

Written under the direction of

Dimitris N. Metaxas

And approved by

_____

_____

_____

_____

New Brunswick, New Jersey

JANUARY, 2018

# ABSTRACT OF THE DISSERTATION

## Learning Disentangled Representations in Deep Visual Analysis

By XI PENG

Dissertation Director:
Dimitris N. Metaxas

Learning reliable and interpretable representations is one of the fundamental challenges in machine learning and computer vision. Over the last decade, deep neural networks have achieved remarkable success by learning conditional distributions on the data for the purposes of solving different tasks. However, representations learned by deep models do not always manifest consistent meaning along variations: many latent factors are highly entangled. This may significantly impair the representative power, even though large-scale labeled data and sophisticated learning skills have been applied in training.

In this work, we are interested in learning disentangled representations that encode distinct aspects of the data separately. The objective is to decouple the latent factors in a representation space, where factorizable structures are obtained and consistent semantics are associated with different variables. The disentanglement can be learned in an either supervised or self-supervised manner. Especially, we investigate three different visual analysis tasks: viewpoint estimation, landmark localization, and large-pose recognition. We show that, by learning disentangled representations, deep models are efficient to train and robust to variation, achieving state-of-the-art performance in the wild.

# Acknowledgements

First and foremost, thank you to my advisor Prof. Dimitris Metaxas for the continuous support of my Ph.D. study in the past five years. His guidance helped me in all the time of research, all the effort of publications, and writing of this thesis. His visionary suggestions motivated me to move forward not only in research but also independent thought.

Thank you to all my co-authors whose work is featured in this thesis (in alphabetical order): Manmohan Chandraker, Ahmed Elgammal, Rogerio Feris, Junzhou Huang, Qiong Hu, Kang Li, Dimitris Metaxas, Sharath Pankanti, Nalini Ratha, Kihyuk Sohn, Christian Vogler, Xiaoyu Wang, Fei Yang, Xiang Yu, Yang Yu, Shaoting Zhang.

Thank you to my defense committee members: Prof. Dimitris Metaxas (Chair), Prof. Jingjin Yu, Prof. Konstantinos Michmizos, and especially thanks Prof. Xiaoming Liu for serving as the outside committee member in Michigan State University.

Thank you to the other professors who have served as my qualify exam members: Prof. Apostolos Gerasoulis, Prof. Konstantinos Michmizos and Prof. Amelie Marian.

Thank you to IBM T.J. Watson Research Center and NEC Labs America for supporting me research internships. Thank you to my internship mentors and colleagues for all the valuable suggestions and discussions, which significantly exploit and extend my research.

Thank you to other Ph.D. students in the vision group (in alphabetical order): Rahil Mehrizi, Zhiqiang Tang, Yu Tian, and Long Zhao. During the past year, we have been working together to push the ongoing research forward in several different areas.

Last but not the least, I would like to express my deepest gratitude to my family. Thank you, Qiong, you are not just my wife, but my companion, my love, and my life. As a Ph.D. student and a young mother, you sacrifice a lot to support me and the family. Thank you, Vincent, you are not just my little boy, but my delight, my inspiration, and my world. Thank you, my parents and parents-in-law, your endless love and support makes me strong in moving forward.

To my parents, my wife, and my son.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

Building machines that possess human intelligence is a dream of scientists and engineers lasting for centuries. It comes even close to fulfillment since artificial intelligence (AI) was founded as an academic discipline in 1950s and many research fields remain hot spots for decades, which include perception and representation, reasoning and planning, motion and manipulation, creativity and generation [112].

Perception and representation, which are the first step to built up intelligent agents, attract intensive research interest in recent years [32]. Generally speaking, perception is the ability to use input from sensors, such as camera, microphone, sonar and others, to deduce aspects of the world and generate knowledge representation. The knowledge representation is then served as observations for following higher-level intelligent activities, such as reasoning and planning.

It is well known that vision plays an important role in human perception as more than 80% information are obtained by our eyes. From the engineering perspective, making machines to gain high-level understanding from visual inputs, such as digital images or videos, is the foundation of many artificial intelligence applications. Therefore, computer vision, which is an interdisciplinary field of mathematics, artificial intelligence, machine learning, signal processing, becomes an important research area with broad applications.

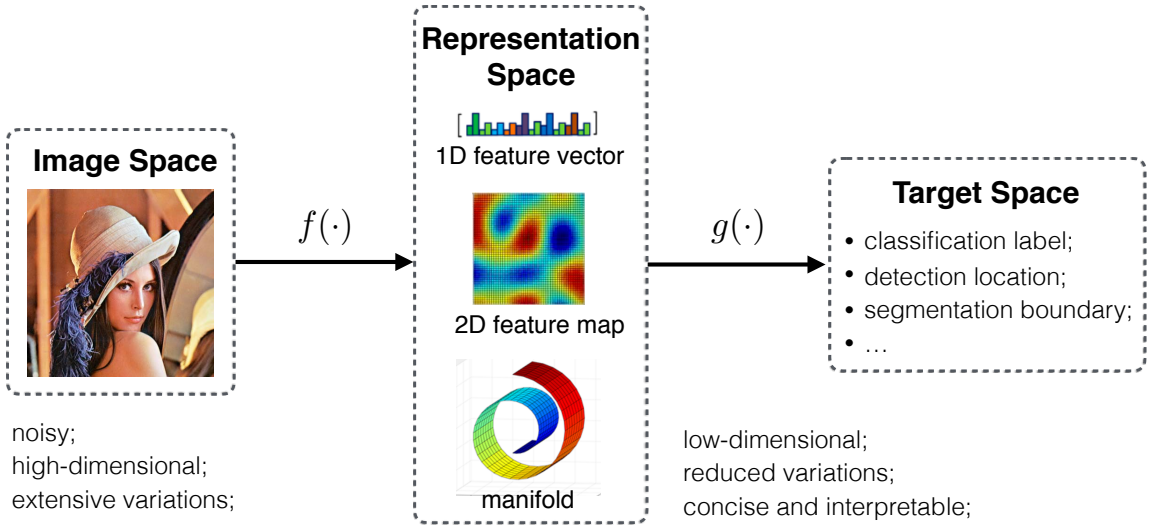



Figure 1.1: Illustration of the two mappings in many computer vision systems. $f(\cdot)$ concentrates information from the image space to achieve abstract embeddings in the representation space; while $g(\cdot)$ aims to project the learned embeddings into the target space for designed objectives. The embedding modality could be 1D vectors, 2D maps, or multi-dimensional manifolds.

## 1.1 Dilemmas in Representation Learning

Generally speaking, many computer vision systems can be summarized as using two mappings to bridge three different domains. Figure 1.1 illustrates this idea. The first mapping $f(\cdot)$ projects pixels in the image space into a representation space to achieve low-dimensional embeddings, which could be 1D vectors, 2D maps, or multi-dimensional manifolds. Then, the second mapping $g(\cdot)$ remaps the embedded representation into a target space to accomplish target objectives, such as classification labels, detection locations, segmentation boundaries, and etc.

In image classification [61], the first mapping extracts image features which are then projected into a one-hot vector to represent the image class label. In object detection [31], the first mapping learns spatial-dependent feature maps which are followed by a second mapping to generate region proposals for object localization. In semantic segmentation [72], a encoder is applied on the image to achieve low-dimensional embeddings, which are then fed into a decoder to recover a set of heatmaps that present pixel-wise class information.

Table 1.1: Configurations of training datasets used to train three recently proposed face recognition networks. A large amount of labeled subjects and images are used, which is not only very expensive but also time-consuming.

| Method | DeepFace [131] | FaceNet [118] | VggFace [90] |
|---|---|---|---|
| Training datasets | SFC | WebFace | VggFace |
| # of images | 4.4M | 200M | 2.6M |
| # of subjects | 4K | 8M | 2.6K |

Table 1.2: Rank-1 face recognition accuracy with respect to different head pose on MultiPIE dataset [34]. The recognition accuracy drops significantly when the head pose changes from frontal to profile.

| Head Pose | 15° | 30° | 45° | 60° | 75° | 90° |
|---|---|---|---|---|---|---|
| VggFace [90] | 97.2% | 96.1% | 92.6% | 84.7% | 62.8% | 34.2% |

There are extensive variations engaged in the image space. First, a single pixel may have a degree of freedom of $256^3$ in a typical RGB image, which results in a extremely high-dimensional space. Second, the imaging conditions can hardly be perfect due to the sensor noise or hardware limitations. Third and most importantly, the imaged contents usually involve extensive intra-class variations and may varies a lot due to many underlining factors. For example, the appearance of a car may look very different depending on if it is imaged in the morning sunshine or a midnight street light; the shape of a chair may look very dissimilar if it is imaged from different viewpoints.

The representation space is usually designed as low-dimensional with restricted variations on purpose. The embeddings in this space are preferred to be informative, concise and interpretable to finial targets. To this end, the first mapping that projects from the image space, which is high-dimensional and filled up with variations, to the representation space, which is relatively low-dimensional and purified for interpretation, is extremely important in vision based perception. In other words, learning reliable and interpretable representations is a crucial and fundamental task in computer vision.

However, learning reliable representations is not only very expensive but also very hard. Table 1.1 lists training datasets used in three recently proposed face recognition systems: DeepFace [131] developed by Google, FaceNet [118] developed by Facebook, and VggFace

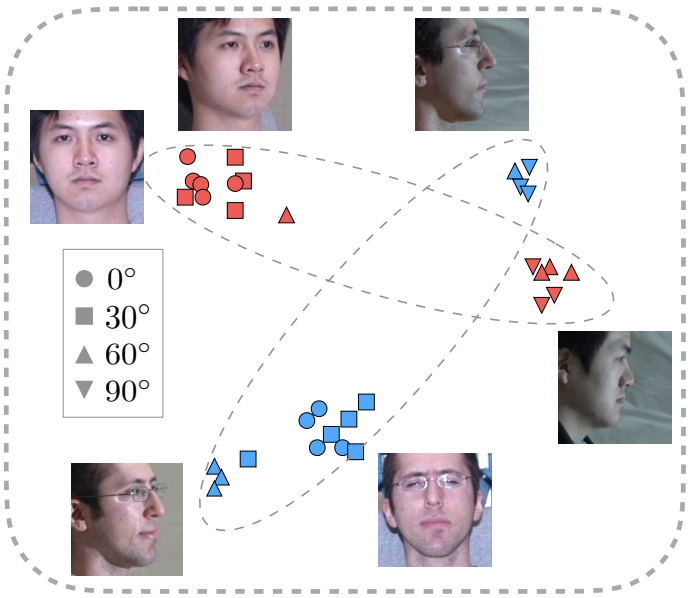



Figure 1.2: Illustration of the feature entanglement. Two subjects (in different colors) from MultiPIE dataset [34] are mapped into the learned representation space of VGGFace [90]. Images in a similar head pose are embedded closer to each other even they belong to different subjects. In other words, generic data-driven features for face recognition might confound images of the same identity with others in large pose conditions.

.

[90] developed by the VGG group. We can see that 2.6 million and 4.4 million labeled images are used to train VggFace and DeepFace, respectively. An astonishing large dataset of 200 million labeled images of 8 million different subjects are used to train FaceNet. Undoubtedly, annotating such a large amount of images is an extremely expensive, time consuming and tedious work, no matter manually or automatically.

The cost is just one aspect of the challenging. An even more frustrating fact is reveled in Table 1.2. As we can see, even after feeding the VGGFace network with 2.6 million labeled images, the performance is still hardly satisfying. For instance, the verification accuracy drops drastically to as low as 34.2% on profile faces (yaw $> 75°$), although it achieves a high accuracy of 97.2% on frontal faces (yaw $< 15°$).

It has been demonstrated in many works that the aforementioned challenges are mainly caused by the difficulty of learning reliable representations [61, 109, 72]. If we simply categorize underlining factors as target-related and target-unrelated, we would find that in many applications, the latter may dominate over the formal due to the factor entanglement

in the representation space. Figure 1.2 illustrates the dilemma in representation learning. Two subjects (in different colors) from MultiPIE dataset [34] are mapped into the learned representation space of VGGFace [90]. We can see that generic data-driven features for face recognition might confound images of the same identity with others in large poses settings: images of the same subjects may be far away from each other in the representation space because of different head poses; while images of different subjects may be close to each other in the representation space because of similar head poses. In other words, target-unrelated factors such as head pose may dominate over target-related factors such as identity in representation learning of face recognition.

## 1.2 Learning Disentangled Representations

Generally speaking, a favorable representation learned from the image space should be compact, interpretable, and most importantly, exclusively related to the final target. As we mentioned, the image space is usually high-dimensional and noisy. Target-unrelated factors may confound target-related ones due to the nature of massive entanglement in the representation space. To achieve reliable representations, the first mapping, which maps from the image space to the representation space, should be capable to factorize and further decouple the latent factors to split target-related and target-unrelated representations. To this end, how to learn a reliable mapping from the image space to the representation space becomes a fundamental goal of many computer vision problems.

The factorization and disentanglement can be achieved either explicitly or implicitly. For instance, in the MNIST handwritten digit recognition [67], the latent variables can be explicitly categorized as *style factors* (who wrote the digit) and *content factors* (which number it is). By decoupling style and content, we can identify not only which number the written digit is but also who wrote the digit. In the multi-view perceptron problem [162], we can explicitly split the latent factors into *object factors* (what the object is) and *viewpoint*

*factors* (from which angle the image is taken). The disentangled representations can be used to recognize the subject consistently no matter how the viewpoint changes. In the face generation task [71], multiple latent factors can be implicitly learned and decoupled from each other, such as wearing eyeglasses, having mustache, black hair, and etc. Then these decoupled factors can be reorganized together to guid the generation of a novel face image that satisfies all required descriptions but never presents in the training dataset.

In this study, we investigate learning disentangled representations and its applications in deep visual analysis. More specifically, we explicitly model three major latent factors in face representations: head pose (P) [98], subject identity (I) [93], and facial expression (E) [43]. We carefully design strategies to factorize and disentangle the aforementioned PIE factors, in order to satisfying different facial analysis tasks. For instance, the variant factors in head pose estimation is undoubtedly the pose (P), while the invariant factors could be both the identity (I) and expression (E) [98, 99, 97]. Similarly, the variant factors are pose (P) and expression (E) while the invariant factors are identity (I) in facial landmark tracking, since a 3D face shape of the same person is tracked [96, 100, 94, 103, 95]. In large-pose face recognition, we are trying to achieve reliable representations that are variant to identity (I) but totally invariant to pose (P) and expression (E) [93, 101, 89]. We show that by learning disentangled representations, reliable and robust performance can be achieved in visual analysis, such as view-point estimation, key point tracking, and large-pose recognition.

The essential idea in learning disentangled representation is to distinguish target-related and target-unrelated factors in the representation space. In other words, we intend to factorize the latent factors into variant and invariant categories according to the objective, and then decouple factors in the two categories to eventually purify embeddings in the representation space. The distilled representations is exclusively correlated to the target, while unrelated factors are significantly suppressed at the same time.

In Section 2, we decouple pose-related and pose-unrelated factors in a latent instance parametric space [97], where 3-dimensional head pose can be estimated accurately no matter

the changes of pose-unrelated factors such as identity, expression, illumination, and etc. In Section 3, we split the identity embedding, which is invariant to facial landmark locations as the same subject are tracked, from the pose and expression embedding, which determines the displacements of facial landmark locations frame to frame, in the bottleneck of a recurrent encoder-decoder network [94]. The additional identity supervision significantly improves the tracking accuracy and robustness especially in large pose and partial occlusions. In Section 4, we leverage the disentanglement of identity and pose factors by designing a novel feature reconstruction task [101]. The disentangled identity representation guarantees accurate face recognition in large head pose up to 90°. In Section 5, we conclude the whole work and discuss plans of future work to further extend the ongoing research.

# Chapter 2

# Head Pose Estimation

Three-dimensional head pose estimation from a single 2D image is a challenging task with extensive applications. Numerous of approaches have been proposed using various machine learning methods such as classification, regression, deformable model, manifold, and deep neural network. In this work, we leverage manifold embedding to decouple pose-related factors and pose-unrelated factors, such as identity, expression, and illumination, in the representation space. Specifically, we propose a coarse-to-fine pose estimation framework, where a circle and a three-sphere are used to model the manifold topology on the coarse and fine layer, respectively. Out-of-sample instances are first approximated in the latent space, which is then used to refer 3D head pose by alternating search. Our approach can uniformly factorize multiple factors in the proposed instance parametric subspace, where novel inputs can be synthesized under a generative framework. Moreover, our approach can effectively avoid the manifold degradation problem when 3D pose estimation is performed. The results on both experimental and in-the-wild databases demonstrate the validity and superior performance of our approach compared with the state-of-the-arts. [1]

[1]The project page: https://sites.google.com/site/xipengcshomepage/cviu2015

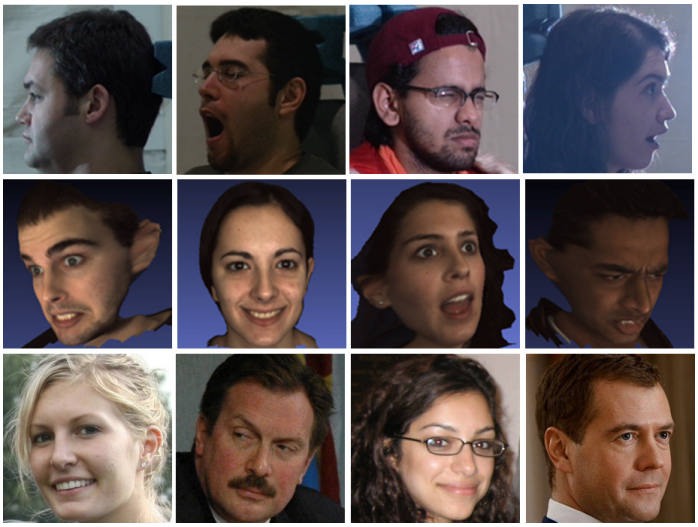


Figure 2.1: Examples from different databases. The first two rows show examples of experimental databases (CMU-MultiPIE [35] and BU-4DFE [149]). The third row show examples of Faces in-the-wild database (AFW [158]). There exist extensive variations of pose-unrelated factors such as identity, facial expression, illumination and etc.

## 2.1 Introduction

The task of inferring the orientation of human head is defined as head pose estimation. In the computer vision context, head pose estimation is specified as processing steps to transform a pixel-based digital image representation of a head into a high-level concept of direction [83]. Many tasks related to face analysis rely on accurate head pose estimation. For instance, a multi-pose face recognition system can carry out head pose estimation first and then select face images with similar poses for matching; a 3D face tracker can use head pose information to render the face model for the optimal fitting. Other applications of head pose estimation include inferring human gaze direction in the human-computer interaction (HCI) system, monitoring driver awareness for safety driving [84] and inferring the intentions of people in both verbal, and nonverbal communication environments [78].

It is often assumed that the human head is a rigid object and three Euler angles, pitch, yaw and roll, can be used to depict the head orientation [27]. Estimating the three angles from a single 2D image is a challenging task, since there exists extensive variations among pose-unrelated factors such as identity, facial expression, illumination condition and other

latent variables. Figure 2.1 shows the examples of the variations. In many cases, these pose-unrelated factors play a more significant role on the appearance variations than pose changes do [5, 7, 44]. Therefore, extracting information that ensures pose changes can dominate over pose-unrelated factors is a crucial point in designing the head pose estimator.

Numerous of approaches have been proposed over the past decades for head pose estimation. We arrange existing methods into four categories: *classification-based approaches* [46, 143], *regression-based approaches* [106, 75, 107, 44, 36], *deformable-model-based approaches* [155, 117, 158] and *manifold-embedding based approaches* [42, 29, 5, 7]. *Classification-based approaches* are limited to estimate 1-dimensional (only yaw angle) discrete head pose. *Regression-based approaches* can predict 3-dimensional continuous pose efficiently, but they are extremely sensitive to noise and pose-unrelated factors. *Deformable-model-based approaches* rely on the localization of facial landmarks, which limits their capability to handle extensive instance variations especially in low resolution images. *Manifold-embedding-based approaches* assume that facial images with consecutive head poses can be viewed as nearby points lying on a low-dimensional manifold embedded in the high-dimensional feature space. Head pose angles can be recovered by measuring the points' distribution in the manifold embedding space.

Although manifold-embedding-based approaches have achieved great success in the former research, they still suffer from multiple limitations. First, there is no guarantee that pose-related factors can dominate over pose-unrelated factors in the manifold embedding process, since pose-unrelated factors will distort the manifold building process and result in geometry deformation across instance manifolds (different combinations of identity, expression and illumination) [83]. Though various approaches [42, 5, 7] have been proposed to partially solve this problem, they either focus on single pose-unrelated factor like identity while ignoring the others [42], or cannot handle multiple pose-unrelated factors in a uniform way [7]. Second, former methods tries to learn the mapping from the high-dimensional feature space to the low-dimensional manifold embedded representation. This mapping

direction would cause manifold degradation (highly folded or self-intersection) [25] when the manifold topology is complicate (in the case of 3-dimensional pose estimation). Hence, most manifold-embedding-based estimators are limited to provide only 1-dimensional yaw estimation while ignoring the pitch and roll variations. Third, the projections from the image feature space to the low-dimensional manifold are defined only on the training space [1]. The entire embedding procedure has to be repeated since they lack of the ability to depict the out-of-sample inputs in an efficient way [5].

To address the limitations of existing methods, we propose a manifold-embedding-based coarse-to-fine framework for 3-dimensional head pose estimation. This approach employs the unit circle and 3-sphere to model the uniform manifold topology on the coarse and fine layer respectively. By learning the instance-dependent nonlinear mappings from the unit circle or 3-sphere to every instance manifold (certain person with certain expression under certain illumination condition), the pose-related and -unrelated factors can be decoupled in a latent instance parametric subspace. The basic idea is that pose-unrelated factors dominate the geometry deformations across different instance manifolds. Hence, we can factorize the instance variations, which are encoded in the geometry deformations, in the instance parametric subspace.

There are several merits of our approach. First, the coarse-to-fine framework guarantees the efficient and accurate 3-dimensional continuous head pose estimation. Second, it can uniformly parameterize multiple pose-related and -unrelated factors under a uniform framework in the latent space. Third, the designed mapping direction of the manifold embedding, which is completely different from the existing methods, can effectively avoid the manifold degradation problem when 3-dimensional pose estimation is performed. Last but not least, the out-of-sample data can be effectively synthesized in the instance parametric subspace, which guarantees the generative ability of our approach.

## 2.2 Related Work

Head pose estimation using visual perception has been a broad and diverse research field for decades. We summarize existing methods and briefly review the most representative and related works to our approach.

### 2.2.1 Taxonomy of Head Pose Estimation

There exists a large amount of literatures on the topic of head pose estimation in the scope of computer vision area. Based on the type of data relied on, existing methods can be coarsely classified as 2D image-based verse 2.5D image-based. 2D image-based approaches carry out head pose estimation based on features extracted from 2D image. While 2.5D image-based approaches take advantage of the rapid development of depth/range sensing technology [33] to design head pose estimators. They rely on both standard 2D image information and reliable depth/range information.

It has been demonstrated that the additional depth/range information can effectively overcome limitations which are inherent with methods relied on only 2D images [12, 26], such as illumination changes and extracting feature from texture-less face regions. However, approaches solely rely on standard 2D image attract much more attention than ones requiring additional depth/range sensing, when considering the extensive application and affordability of the 2D cameras. In the rest part of this section, we focus on the taxonomy of the 2D image-based methods. Broadly speaking, the evolutionary taxonomy of head pose estimation approaches can be outlined as following categories

- **Classification based approaches** split the pose variation space into discrete intervals and apply classifiers learned in a unsupervised or supervised way to assign the pose label to the query image.

- **Regression based approaches** try to learn a linear or nonlinear mapping from the feature space to the pose variation space and use this mapping to predict pose parameters given the testing sample.

- **Deformable model based approaches** fit a flexible face shape model composed by sparse landmarks onto the object facial image and estimate head pose from the configuration of landmarks.

- **Manifold embedding based approaches** seek low-dimensional manifolds of pose variation embedded in the high-dimensional feature space and pose can be estimated by embedding a new image into the learned manifolds.

### 2.2.2 Review of Existing Methods

We follow the taxonomy and briefly review representative methods in each category. We focus on more recent approaches to provide an overview of the up-to-date status on the head pose estimation topics.

**Classification based approaches** in the simplest form are template matching methods. These methods compare testing image with all exemplars in the training set under defined distance metrics, and assigned pose label to the required image with the label of the most similar template. Other supervised learning based tools such as Support Vector Machines (SVMs) [69] and Kernel Linear Discriminant Analysis (KLDA) [9] are demonstrated to estimate head pose from single image. These methods split training images into several groups by specifying discrete pose labels, and train multiple binary classifiers for all groups under one-to-all framework [46, 143]. The pose label of testing image can be acquired by the classifiers with the most confidence.

There are many disadvantages with classification based approaches: first, they can only provide coarse head pose estimation results due to the discrete labeled groups in the training set; second, they can only handle 1-dimensional yaw variations and extensions to

3-dimensional pose changes including pitch, yaw and roll would encounter great obstacle; third, they suffer from over-fitting problems which result in sensitiveness to pose-unrelated variations like facial expression and illumination changes.

**Regression based approaches** use the mapping learned from the image feature space to the pose space to recover the pose in the testing image. Regressors such as Support Vector Regressors (SVRs) [75] and Gaussian Processes Regression (GPR) [107] can be employed to construct the pose estimator. Framework like Artificial Neural Networks (ANN) [9] is also shown the capability to recognize pose orientation from images [106]. More recently, Huang et al. [44] use Supervised Local Subspace Learning ($SL^2$) to learn a local linear model which shows prominent potential to provide accurate head pose estimation when the training data is pretty sparse and non-uniformly sampled. Haj et al. in [36] apply Partial Least Squares (PLS) regression to alleviate the negative effect on pose estimation when there exists misalignment of head location in the image.

Regression based methods are characterized with fast prediction speed and capability to output continuous pose angles, thus they attract lots of research interest in previous literatures. However, they also suffer from obvious drawbacks. There is no guarantee that the learned mapping under regression framework can properly reflect the connection between the image feature and the pose label, since pose-unrelated factors would also affect the mapping process. Another weakness of regression based methods is they are highly sensitivity to noise, which limits their capability to handle the extensive variations among pose-related and -unrelated factors.

**Deformable model based approaches** take advantage of different types of flexible facial model to refer head pose. Usually the facial model is constituted by both the shape model (specific facial points and their 2D coordinates in an image) and the appearance model (2D texture of corresponding facial points) [151]. The specific facial points, such as the corners of eyes, mouth and nose tip, are notated as landmarks. By fitting landmarks to the correct responding positions in a facial image, the head pose can be estimated from

the geometric shape formatted by the landmarks. For instance, Active Shape Model (ASM) [20] find the primary modes of shape variation among all the training faces and use the largest principal component to recover the facial shape of the testing face. Then the pose parameters can be estimated by applying Bayesian inference [155]. Saragih et al. [117] employ Constrained Local Model (CLM) to estimate global (scale, pose and face location) and local (identity and expression) parameters simultaneously. Mean-Shift algorithm is applied on the response map of each landmark to find the local optimal under the global constraint. Zhu et al. [158] employ tree structured deformable part model [60] to locate facial landmarks and use view-based models to describe the topological changes caused by head pose variations. The main challenge of these approaches is how to effectively incorporate domain knowledge into traditional models [47], and how to efficiently solve them [48].

There are multiple limitations for deformable model based methods when regarding pose estimation task. First, the pose prediction accuracy of flexible model based methods extremely relies on the precision of the landmark locations, in other word, they are quite sensitive to the location error/noise; second, pose-unrelated factors such as identity and facial expression will affect the aligned shape, which contaminates the positional information of landmarks for predicting pose; third, these methods need to fit all the landmarks before inferring the head pose, while solving the optimization problem for landmarks localization is fairly time-consuming regarding the efficiency.

**Manifold embedding based approaches** try to acquire a low-dimensional manifold where data points with consecutive poses lies on. Either linear embedding techniques like Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) [9] or non-linear embedding methods such as Isomap [42, 125], Locally Linear Embedding (LLE) [5] and Local Embedded Analysis (LEA) [29] can be employed to find the low-dimensional space. Head pose estimation can be carried out in this space using classification [42, 29] or regression [5] strategies.

The key challenge of manifold embedding based approaches is how to ensure pose variations can dominate over pose-unrelated factors during the embedding process. To address this problem, Hu et al. [42] apply Isomap embedding to model each individual manifold as an ellipse. The ellipses from different individuals can be normalized to a unified embedding by carrying out an ellipse fitting process. However, this approach is designed based on the classification framework and it does not utilize the pose label information which are available in the training set in a supervised way. Balasubramanian et al. [5] propose the biased manifold embedding method for person-independent head pose estimation. They define a distance metric based on pose angles. This metric is applied to bias the distance between data points in high-dimensional feature space. And the effect of identity variations can be eliminated by removing pose-unrelated dimensions. However, these approaches mainly focus on identity variations and they still have the tendency to build manifolds for other pose-unrelated factors as well as pose. Besides, they have limited representation ability for out-of-sample data that results from the descriptive model used. Moreover, they can only handle at most 2-dimensional pose estimation due to the manifold degradation issue. It is not a trivial task to extend them to 3-dimensional pose estimation.

## 2.3 Method

This section describes our approach in details. First, we discuss the motivation of the coarse-to-fine pose estimation framework. Then we propose the instance parametric subspace and the uniform geometry representation. The instance parameterization can be achieved by conducting instance-dependent mappings and pose-related/unrelated factorization in the subspace. Finally, an efficient pose referring solution is provided to estimate head pose in the testing image. An overview of our approach is illustrated in Figure 2.2.

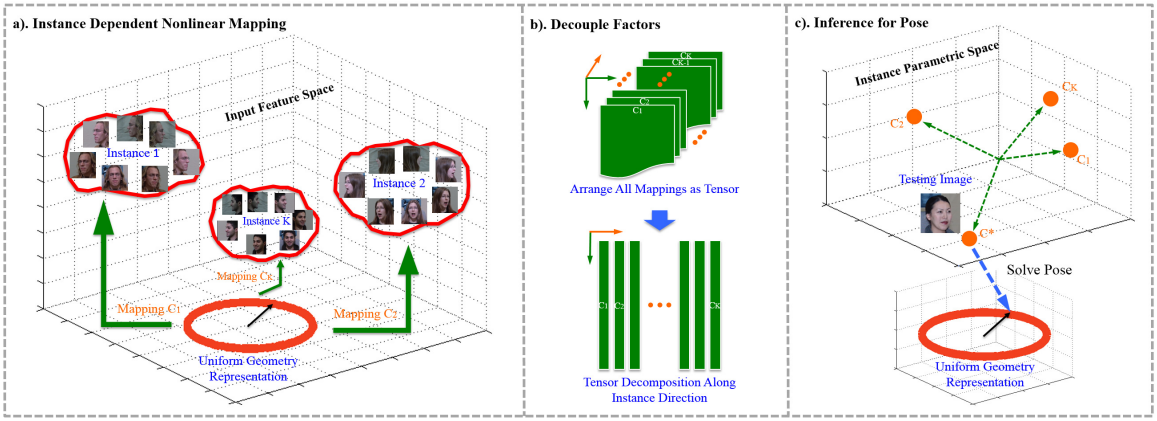



Figure 2.2: Illustration of the training and learning procedure of our approach. We take 1D yaw estimator for example. The 3D head pose estimation have a similar framework but replace the circle with a 3-sphere for the uniform geometry representation. a). Learn the instance dependent mappings from the uniform geometry representation to each instance manifold; b). Arrange the set of instance dependent mapping coefficients matrix as a tensor and carry out tensor decomposition along the instance direction to decouple the pose-related and -unrelated factors; c). Given a single testing image, parameterize the testing instance in the instance parametric space and search the space of the uniform geometry representation to find the pose solution.

### 2.3.1 Coarse-to-fine Pose Estimation Framework

The appearance of human head is almost symmetrical with respect to the yaw angle variations, for example the left profile is almost the same as the right profile. Taking advantage of this fact, we propose a cascade pose estimation approach in a coarse-to-fine framework: 1D yaw estimator on the coarse layer and 3D pose estimator on the fine layer.

The single estimator on the coarse layer only needs to provide a fast and rough prediction of the yaw angle. Based on the coarse prediction, the 1D yaw variation space can be divided into multiple discrete intervals. Each 3D pose estimator on the fine layer is responsible for one interval. Once the testing image is categorized into one of the intervals based on the coarse estimation of yaw angle, the corresponding 3D pose estimator in the fine layer is activated to estimate pitch, yaw and roll simultaneously.

This coarse-to-fine framework can reduce the trained model size of the estimators on the fine layer, since it only needs to train pose estimators for left profile. A testing image with right profile can be processed by left profile pose estimators with flip operation. Besides, splitting the yaw variation space into multiple intervals can shrink the solution space for the 3D pose estimator to find the true pose, which can accelerate the prediction speed in

the testing phase. Moreover, a reduced searching space can make it easier to find the global optimum when inference method, which is prone to local optimal, is employed.

### 2.3.2 Instance Parametric Subspace

Suppose there are $K$ instances (a certain combination of pose-unrelated factors, i.e. identity, expression, illumination and etc.) in the training data set. For the $k$-th instance $\mathcal{I}^k$, we have a sequence of images $Y^k = \{\mathbf{y}_i^k \in \mathbb{R}^D, i = 1, \cdots, N_k\}$, and their corresponding head pose angles $\Theta^k = \{\theta_i^k \in \mathbb{R}^d, i = 1, \cdots, N_k\}$, where $\mathbb{R}^D$ and $\mathbb{R}^d$ are $D$-dimensional image feature space and $d$-dimensional pose variation space respectively (e.g. $d = 1$ represents the 1D yaw angle, $d = 3$ represents the 3D pose angle including yaw, pitch and roll). Notice that image sequences for different instances do not need to have the same length. These high-dimensional multiple views are expected to lie on a low-dimensional manifold $\mathcal{M}^k \in \mathbb{R}^e$.

The allure of manifold embedding techniques is that the pattern of head movement can be preserved during the dimension reduction in the high-dimensional feature space. However, as we discussed in Section 2.2, the most significant drawback of existing manifold embedding based methods is that they have the tendency to build manifolds for pose-unrelated factors as well as pose [83, 5]. For example, consider the cases that two images of different persons in the same pose and two images of the same person in slightly different poses. The fist pair of images would have a larger metric distance than the second pair in the embedding space since the identity factor can dominate over pose when the pose variation is small.

To address this problem, we resort to Homeomorphic Manifold Analysis (HMA) [126, 25] to formulate our framework. Research in [83, 42, 5] indicates that manifolds arise from head pose variation should have similar geometry in viewing space. Pose-unrelated factors, such as identity, facial expression, illumination and other latent variables, will result in geometric deformation across instance manifolds. The key idea of our approach

is that different instance manifolds can be viewed as points distributed in a latent *instance parametric space*, where the spacial distribution of a point (represents a manifold) is dominated by the manifold deformation.

Considering the fact that all the instance manifolds are topologically equivalent (with similar geometry in the viewing space), we can define a uniform geometry representation $\tilde{\mathcal{M}} \in \mathbb{R}^e$ in the viewing space, which is homeomorphic to each instance manifold $\mathcal{M}^k$. In other words, each $\mathcal{M}^k$ is a deformed instance of $\tilde{\mathcal{M}}$. To construct the instance parametric space, we can learn an instance-dependent nonlinear mapping from $\tilde{\mathcal{M}}$ to $\mathcal{M}^k$. The mapping coefficient $\mathbf{C}^k$ encodes all the pose-unrelated factors of the $k$-th instance. We get $K$ mapping coefficient matrices $\{\mathbf{C}^1, \cdots, \mathbf{C}^K\}$ for all the $K$ instances in the training data. By arranging them as a tensor and carrying out tensor decomposition along the instance dimension, we can find $K$ instance bases to span the instance parametric space.

There are several benefits to introduce the instance parametric space. First, we can handle multiple pose-unrelated factors simultaneously under a uniform framework since all instance factors are factorized in the space. Second, the space guarantees the generative ability of our approach. We can directly synthesize new instances to approximate the out-of-sample testing data without repeating the whole embedding process. Moreover, we can use $L$ instead of $K$ ($L < K$) instance bases to span a more compact subspace, which can significantly improve the computational efficiency in terms of both time and space without sacrificing the prediction accuracy.

### 2.3.3 Uniform Geometry Representation

The geometry of $\tilde{\mathcal{M}}$ is important for manifold embedding since it provides prior information to model the topology of the instance manifold $\mathcal{M}^k$. It should faithfully represent the structure of the head motion as well as simple enough for fast computing.

Let $\theta = (\alpha, \beta, \gamma)^T \in \mathbb{R}^3$ denotes the 3D head pose, where $\alpha, \beta, \gamma$ refer pitch, yaw and roll respectively. To find a faithful structure of $\tilde{\mathcal{M}}$, we consider the case that the head is

stationary where the camera is moving around to simulate the head pose variations. The camera should moves along a circle to simulate the 1D yaw variations $\beta$. In this case, the motion trajectory of the camera is a partial viewing circle centered at the head. Similarly, when pitch variations $\alpha$ is involved, the motion trajectory should be upgraded to a partial viewing sphere to simulate the 2D head movement. To get the 3D pose variations, we can image that at each point on the surface of the viewing sphere, the camera has another freedom to move along a tangent partial circle to obtain the roll movement $\gamma$. The geometry of the motion trajectory is a 3-sphere embedded in $\mathbb{R}^4$.

Recall here we only need to preserve the topology of the instance manifold not the metrical measurement, we use a partial unit circle and a partial unit 3-sphere to model the uniform representations of the 1D (yaw) and the 3D (pitch, yaw and roll) head pose variations respectively in our coarse-to-fine framework.

**1D yaw estimator on the coarse layer.** It provides discrete yaw prediction to categorize the testing image into one of several yaw intervals. We use a half unit circle to model the uniform geometry representation for yaw varies from $-90°$ to $90°$. The embedding coordinates $\mathbf{x} \in \mathbb{R}^2$ on the unit circle is:

$$\mathbf{x} = \begin{bmatrix} \cos\beta \\ \sin\beta \end{bmatrix}, \ where\ \beta \in [-\frac{\pi}{2}, \frac{\pi}{2}]. \tag{2.1}$$

The output of the 1D yaw estimator on the coarse layer is a discrete label, which indicates the corresponding interval. Since most head movement occurs within the yaw interval from $-45°$ to $45°$ and face appearance is symmetrical regarding yaw variations, we split the yaw range into 4 intervals, $[-90°, -45°)$, $[-45°, 0°]$, $(0°, 45°]$ and $(45°, 90°]$, to best balance the trade off between the estimation accuracy and the computational complexity.

**3D head pose estimators on the fine layer.** It output continuous estimation result for pitch, yaw and roll simultaneously. There should be four estimators on the fine layer regarding four discrete intervals. However, considering the fact that the left and right

head profiles are symmetric, we only need two estimators on the fine layer: one for $\beta \in [-90°, -45°)$ and the other for $\beta \in [-45°, 0°]$. Testing image with $\beta \in (0°, 90°]$ can be horizontally flipped to use the fine estimators for pose prediction. We use a partial 3-sphere to model the uniform geometry representation. The embedding coordinates $\mathbf{x} \in \mathbb{R}^4$ on the partial 3-sphere is:

$$\mathbf{x} = \begin{bmatrix} \cos\alpha \\ \sin\alpha\cos\beta \\ \sin\alpha\sin\beta\cos\gamma \\ \sin\alpha\sin\beta\sin\gamma \end{bmatrix}, \tag{2.2}$$

where $\beta \in [-\frac{\pi}{4}, 0]$ or $[-\frac{\pi}{2}, -\frac{\pi}{4}]$, $\alpha \in [-\frac{\pi}{3}, \frac{\pi}{3}]$ and $\gamma \in [-\frac{\pi}{3}, \frac{\pi}{3}]$.

During the testing, we first conduct the 1D yaw estimation to narrow the solution space. Then the corresponding fine estimator is activated to provide a continuous 3D pose estimation.

### 2.3.4 Instance Dependent Nonlinear Mapping

In former manifold embedding based approach, the dimension reduction is conducted by mapping from the high-dimensional feature space to the low-dimensional manifold embedding space. However, this strategy suffers from manifold degradation issue when dealing with complicate manifold topology. In the case of 3D pose estimation, there is no guarantee that the embedding procedure can obtain the faithful 3-sphere structure since the embedded manifold would be highly folded or self-intersected. To overcome this problem, we follow the HMA and learn the mapping in the reverse direction [24]: *from the low-dimensional uniform geometry representation to the high-dimensional feature space*. See Figure 2.2 for an illustration. This completely different mapping direction has significant advantages: it can not only avoid the manifold degradation problem but also greatly simplified the instance factorization in instance parametric space since the uniform manifold topology is preserved.

Giving the fact that each of $\{\mathcal{M}^1, \cdots, \mathcal{M}^K\}$ is homeomorphic to $\tilde{\mathcal{M}}$ and the geometry deformation from $\tilde{\mathcal{M}}$ to $\mathcal{M}^k$ is resulted from pose-unrelated variations, we can obtain the factorization by learning a set of nonlinear mapping from $\tilde{\mathcal{M}}$ to $\mathcal{M}^k$. Let us consider the $k$-th instance with input image sequence $Y^k$ and the corresponding head pose $\Theta^k$. The embedding coordinates for the image sequence on the uniform geometric representation are $X^k = \{\mathbf{x}_i^k \in \mathbb{R}^e, i = 1, \cdots, N_k\}$ following the definitions in Equation (2.1) and Equation (2.2). We select $m$ mapping centers $\{\mathbf{z}_j \in \mathbb{R}^e, j = 1, \cdots, M\}$ evenly spaced along $\tilde{M}$. A nonlinear mapping function $[s : \mathbb{R}^e \rightarrow \mathbb{R}]$ can be learned by using Radial Basis Function (RBF) interpolation [105]. This function maps from $e$ dimensional embedding space to 1 of the $D$ dimensional image feature space:

$$s(\mathbf{x}_i) = \sum_{j=1}^{M} \lambda_j \phi(||\mathbf{x}_i - \mathbf{z}_j||) + P(\mathbf{x}_i), \quad \mathbf{x}_i \in \mathbb{R}^e, \tag{2.3}$$

where $||\mathbf{x}_i - \mathbf{z}_j||$ is the Euclidean distance defined in the manifold embedding space, $\lambda_j$ are coefficients, $\phi$ is the RBF with popular choices including:

$$\phi(r) = \begin{cases} r & biharmonic \\ e^{-cr} & Gaussian \\ (r^2 + c^2)^{1/2} & multiquadric \end{cases}$$

The linear polynomial for positive semi-definite kernels is needed to span the null space for regularization and defined as:

$$P(\mathbf{x}_i) = \omega_0 + \omega_1 \mathbf{x}_i. \tag{2.4}$$

The full mapping $[S : \mathbb{R}^e \rightarrow \mathbb{R}^D]$ can be organized as:

$$S(\mathbf{x}_i) = C^k \begin{pmatrix} \psi(\mathbf{x}_i) \\ 1 \\ \mathbf{x}_i \end{pmatrix}, \quad \mathbf{x}_i \in \mathbb{R}^e, \tag{2.5}$$

where the mapping coefficient matrix $C^k$ and the nonlinear kernel mapping $\psi(x)$ have the following form:

$$C^k = \begin{pmatrix} \lambda_{11} \cdots \lambda_{1m}\ \omega_{10}\ \omega_{11} \\ \vdots \\ \lambda_{D1} \cdots \lambda_{Dm}\ \omega_{D0}\ \omega_{D1} \end{pmatrix} \quad and \quad \psi(\mathbf{x}) = \begin{pmatrix} \phi(|x - \mathbf{z}_1|) \\ \vdots \\ \phi(|x - \mathbf{z}_m|) \end{pmatrix}.$$

The dimension of $C^k$ is $D \times (M + 1 + e)$. All of the $N_k$ images in the sequence of the $k$-th instance should satisfy the mapping $C^k$. To ensure the orthogonality of the RBF interpolation, we add the constraint:

$$\sum_{j=1}^{M} \lambda_j Q(\mathbf{z}_j) = 0, \tag{2.6}$$

where $Q(\mathbf{z}_j) = (1, \mathbf{z}_j)^\top$. By arranging Equation (2.4)-(2.6) together, we can obtain a linear system for the $k$-th instance:

$$\begin{pmatrix} \Phi & P^\top \\ Q & \mathbf{0} \end{pmatrix} C^{k\top} = \begin{pmatrix} Y^k \\ \mathbf{0} \end{pmatrix}, \tag{2.7}$$

where $\Phi$ is a $N_k \times M$ matrix:

$$\Phi = \begin{pmatrix} \phi(|\mathbf{x}_1 - \mathbf{z}_1|) & \cdots & \phi(|\mathbf{x}_1 - \mathbf{z}_M|) \\ \vdots & \ddots & \vdots \\ \phi(|\mathbf{x}_{N_k} - \mathbf{z}_1|) & \cdots & \phi(|\mathbf{x}_{N_k} - \mathbf{z}_M|) \end{pmatrix}.$$

Block $P$ has the dimensions of $(1 + e) \times N_k$ with the $i$-th column is $(1\ \mathbf{x}_i^\top)^\top$, while $Q$ is a $(1 + e) \times M$ matrix with the $j$-th column is $(1\ \mathbf{z}_j^\top)^\top$. $Y^k$ is the $N_k \times D$ dimensional image feature matrix, which is arranged as $(\mathbf{y}_1^\top \cdots \mathbf{y}_{N_k}^\top)^\top$ with $\mathbf{y}_i$ is the $D$ dimensional image feature for the $i$-th image in the $k$-th instance sequence. The linear system in Equation (2.7) has a

closed-form solution:

$$C^{k\top} = \begin{pmatrix} \Phi & P^\top \\ Q & \mathbf{0} \end{pmatrix} \backslash \begin{pmatrix} Y^k \\ \mathbf{0} \end{pmatrix}. \tag{2.8}$$

The instance nonlinear mapping can effectively avoid the manifold degradation issue. The faithful manifold geometry representation can be expected even when the topology is complicate. Moreover, the nonlinear mapping $C^k$ is easy to train, which guarantees the instance parametric space can be constructed in a very efficient way.

### 2.3.5 Separating Pose-related and -unrelated Factors

Instance manifolds are distributed in the instance parametric space while the mapping coefficients encode manifold deformations from the uniform geometry representation to every instance manifold. We can learn an array of corresponding mapping coefficients $\{C^1 \cdots C^K\}$ for all of the $K$ instances in the training data. It is worth emphasizing that this coefficients array encodes both within-manifold pose variations (pose-related) and across-manifold instance variations (pose-unrelated).

The array of coefficients can be arranged as a tensor $\mathcal{C}$ with the dimensions of $D \times (M + 1 + e) \times K$. Following the tensor decomposition in [64], the pose-related and -unrelated factors can be separated by carrying out tensor decomposition:

$$\mathcal{C} = \mathcal{A} \times_3 \mathbf{I}, \tag{2.9}$$

where $\mathcal{A}$ is a $D \times (M + 1 + e) \times J$ tensor containing all the bases of the RBF space, which can be used to refer pose. And $\mathbf{I} = (I^1 \cdots I^K)$ is a $J \times K$ matrix containing all the bases that span the instance parametric space.

We flatten the tensor $\mathcal{C}$ to matrix $\mathbf{C}$ with $D(M+1+e)$ rows and $K$ columns by reshaping each $C^k$ into a $D(M+1+e)\times 1$ vector. Singular Value Decomposition (SVD) can be carried

out to decouple row bases and column bases:

$$\mathbf{C} = (\mathbf{US})_{D(M+1+e)\times K}\mathbf{V}^{\top}_{K\times K}. \tag{2.10}$$

The columns of **US** are bases of pose variation. The columns of $\mathbf{V}^T = \{I^1, \ldots, I^K\}$ are bases of instance parametric space where $I^k$ is the $k$-th instance basis.

We can expect a full instance parametric space by using $\{I^1, \ldots, I^K\}$. A new testing instance can be approximated by a linear combination of all the bases. However, it is not necessary to employ all of the $K$ bases to achieve the approximation. Ignoring instance bases with trivial eigenvalues will have little effect on generating new instance parameterization since most instance variation directions have been preserved among the bases with significant eigenvalues. From the SVD decomposition, we have $\mathbf{S} = diag(\sigma^1, \cdots, \sigma^K)$ containing $K$ eigenvalues $\sigma^1 \geqslant \cdots \geqslant \sigma^K$. The object is to find a subset $\{I^1, \ldots, I^L\} \subset \{I^1, \ldots, I^K\}$ satisfying:

$$\frac{\sum_{k=1}^{L} \sigma^k}{\sum_{k=1}^{K} \sigma^k} > \delta, \tag{2.11}$$

where $\delta$ is set to $0.6 \sim 0.7$ to best balance the trade-off between the computational efficiency and the estimation accuracy.

$\{I^1, \ldots, I^L\}$ span the ***instance parametric subspace***. The size of the offline trained model is greatly compacted to reduce both the space requirement and the computational complexity, which can significantly boost the online speed.

### 2.3.6 Solving for Pose

Given a single testing input $\mathbf{y}$, suppose the instance parameterization in the instant parametric subspace is $I$ and the embedding coordinates of pose is $\mathbf{x}$. From Equation (2.5) and (2.9),

**Algorithm 1** Pose Referring by Alternating Search

$Uniformly\ or\ randomly\ initialize\ \mathbf{w} = (w^1 \cdots w^L)$
$ct \leftarrow 0$
**repeat**
  $\mathbf{w_0} \leftarrow \mathbf{w}$
  $I^* \leftarrow \sum_{l=1}^{L} w^l \cdot I^l$
  $C^* \leftarrow \mathbf{U}\mathbf{S}I^{*\top}$
  $Reshape\ C^*\ into\ D \times (M + 1 + e)\ matrix$
  $\hat{\mathbf{y}} \leftarrow C^* \begin{pmatrix} \psi(\mathbf{x}) \\ 1 \\ \mathbf{x} \end{pmatrix}$
  $Search\ \mathbf{x}^*\ along\ \tilde{\mathcal{M}}\ to\ minimize\ \|\mathbf{y} - \hat{\mathbf{y}}\|^2$
  **for** $l \leftarrow 1 \rightarrow L$ **do**
    $error \leftarrow \|\mathbf{y} - \hat{\mathbf{y}}\|^2$
    $w^l \leftarrow exp(\frac{-error}{2\sigma^2})$
    $ct \leftarrow ct + 1$
  **end for**
  $\mathbf{w} = (w^1 \cdots w^L)/\sum_{l=1}^{L} w^l$
**until** $\|\mathbf{w} - \mathbf{w_0}\| < \epsilon$ **or** $ct > max\_iter$
**return** $\mathbf{x}^*$

the estimation of $\mathbf{y}$ is:

$$\hat{\mathbf{y}} = S(I, \mathbf{x}) = \mathcal{A} \times_3 I \times_2 \begin{pmatrix} \psi(\mathbf{x}) \\ 1 \\ \mathbf{x} \end{pmatrix}. \tag{2.12}$$

The head pose estimation problem is equivalent to find both $I^*$ and $\mathbf{x}^* \in \mathbb{R}^e$ to minimize the loss function:

$$J = \|\mathbf{y} - \mathcal{A} \times_3 I \times_2 \begin{pmatrix} \psi(\mathbf{x}) \\ 1 \\ \mathbf{x} \end{pmatrix}\|^2. \tag{2.13}$$

We employ an two-step alternative iterative optimization procedure to find $I^*$ and $\mathbf{x}^*$ that minimize the $J$. In the first step, we linearly combine instance bases $\{I^1, \ldots, I^L\}$ with coefficients $\{w^1, \cdots, w^L\}$ to generate the new instance $I^*$. $w^l$ is re-weighted based on the reconstruction error:

$$w^l = exp(\frac{\|\mathbf{y} - \hat{\mathbf{y}}\|^2}{2\sigma^2}),\ l = 1, \cdots, L. \tag{2.14}$$

In the next step we fix $I^*$ and search along $\tilde{\mathcal{M}}$ to find $\mathbf{x}^*$ that minimize the reconstruction error given $I^*$ fixed. The alternative updating of $I^*$ and $\mathbf{x}^*$ continues until convergence or maximal iteration times reached. Algorithm 1 show details of the pose inference process.

## 2.4 Experiments

In this section, we carry out a series of experiments to demonstrate the validity of our approach and evaluate its performance. Several state-of-the-art approaches are compared with our approach on both experimental and faces in-the-wild databases.

### 2.4.1 Databases and Settings

Most pubic face databases come without accurate pose annotations due to the difficulty of manual or automatic pose labeling. Three widely used databases with pose annotations are used in our experiments:

- **CMU-MultiPIE database** [35] contains head images of 336 subjects (identity) illuminated by 15 light directions under 6 expressions. The yaw angles vary from $-90°$ (left profile) to $90°$ (right profile) spaced by $15°$, so there are 13 images in the sequence for each instance.

- **BU-4DFE database** [149] has 100 subjects and each subject performs 6 expressions in front of the 3D face scanner. We render the 3D face model to synthesize head images under different poses. The yaw angle variations are from $-90°$ to $90°$ while the pitch angles vary from $-30°$ to $30°$ and the roll vary from $-30°$ (left bent) to $30°$ (right bent). $15°$ is set as the interval for all the three pose dimensions.

- **Annotated Face In-the-wild database (AFW)** [158] collects 205 images with 468 faces (multiple faces in a single image) from Flickr.com. There are discrete pose

annotations for yaw from $-90°$ to $90°$ with $15°$ spacing in the database labels, which can be used for quantitative evaluations.

Databases constructed under a experimental environment (CMU-MultiPIE and BU-4DFE) can provide accurate pose labels, which facilitates model training and quantitative evaluation of general performance. However, they have limited coverage of variations among identity, facial expression and illumination condition in the perspective of both reality and extensiveness compared to faces in-the-wild database ($AFW$). In contrast, the limitation of faces in-the-wild database is that it lacks detailed pose labels for model training and performance testing, thus we employ experimental databases to train the estimator and test it on faces in-the-wild database to demonstrate the generalization ability of our approach on the challenging real-world data. See Figure 2.1 for examples of the three different databases.

All the images are pre-processed before the training and testing procedure. First, head images are cropped from the background manually (CMU-MultiPIE), or automatically according to facial landmarks annotations (BU-4DFE, AFW). All the cropped images are resized to $64 \times 64$ pixels for features extraction. We choose the Histogram of Oriented Gradients (HOG) features with the consideration of the well balance between computational efficiency and robust performance. Mean Absolute Error (MAE) and Standard Deviation (SD) are used as the performance measurement.

To best evaluate the performance, we compare our method with multiple approaches that are the most representative ones in the corresponding categories:

- **Biased Manifold Embedding Estimator (BME)** [5] biases traditional Isomap, LLE and LE toward unified manifold embedding to handle pose-unrelated factor. The BME estimator can provide 1D yaw estimation with continues result.

- **Supervised Local Subspace Learning Estimator ($SL^2$)** [44] has the potential to handle sparse and non-uniformly sampled training set for head pose estimation. It can estimate full 3D pose variations and output continuous result.

- **Partial Least Squares Estimator (PLS)** [36] demonstrates the capability to handle misalignment that would place a negative effect on the pose estimation performance. The PLS estimator provides 2-dimensional (pitch and yaw) continues result.

- **Tree Structured Part Model Estimator (TSPM)** [158] aligns facial feature points first by fitting a tree structured facial deformable model. Then head pose estimation is conducted bases on the locations of facial landmarks.

- **Hybrid Estimator** [9, 107] use the same coarse-to-fine framework as our approach. On the coarse layer, we use the multi-class Support Vector Machine (SVM) [9] to provide discrete yaw estimation. On the fine layer, we use Gaussian Process Regression (GPR) [107] to predict continuous 3D pose given a yaw interval.

### 2.4.2 Evaluation on Controlled Datasets

The controlled databases, CMU-MultiPIE and BU-4DFE, have accurate pose annotations. We carry out quantitative experiments to compare the performance of different methods.

For the training instances, we randomly pick out 20 subjects under 3 light directions (45°, 90°, 135°) and 6 facial expressions with a total of $20 \times 3 \times 6 = 360$ instances from CMU-MultiPIE to compose dataset (i), and 60 subjects under 6 facial expressions totally $60 \times 6 = 360$ instances from BU-4DFE to compose dataset (ii). For the testing instances, 10 instances are randomly picked out from CMU-MultiPIE and BU-4DFE to compose dataset (iii) and dataset (iv) respectively. There is no overlap between the training and testing instances.

Since CMU-MultiPIE and BU-4DFE are experimental data, to best evaluate the generalize ability to faces in-the-wild data of different approaches, we carry out experiments with both it intra-database (training and testing in the same database) and *inter-database* (training and testing in different databases) settings:

- **Experiment A** uses dataset (i) from CMU-MultiPIE for training, and dataset (iii) from CMU-MultiPIE for intra-database testing.
- **Experiment B** uses dataset (ii) from BU-4DFE for training, and dataset (iv) from BU-4DFE for intra-database testing.
- **Experiment C** uses dataset (i) from CMU-MultiPIE for training, and dataset (iv) from BU-4DFE for inter-database testing.
- **Experiment D** uses dataset (ii) from BU-4DFE for training, and dataset (iii) from CMU-MultiPIE for inter-database testing.

For our approach, we use dataset (i) from CMU-MultiPIE to train the 1D yaw estimator on the coarse layer. The number of the mapping centers is empirically set to $m = 5$ since only discrete angle need to be predicted. We use dataset (ii) from BU-4DFE to train the 3D pose estimators on the fine layer. The numbers of the mapping centers are set to $m = \{7, 9, 7\}$ for pitch, yaw and roll. Especially, the output of the 1D yaw estimator on the coarse layer is discretized as:

$$c_{yaw} = \begin{cases} 1, & when \quad \theta < -45^\circ \\ 2, & when \quad -45^\circ \leqslant \theta \leqslant 0^\circ \\ 3, & when \quad 0^\circ \leq \theta \leqslant 45^\circ \\ 4, & when \quad \theta > 45^\circ \end{cases} \tag{2.15}$$

where labels $c_{yaw} = \{1, 2, 3, 4\}$ represent left profile, left frontal face, right frontal face and right profile respectively. We train two 3D estimators on the fine layer: one for left profile and the other for left frontal face. Our approach has the same prediction accuracy on Experiment A and D, and same prediction accuracy on Experiment B and C since the combinations of the training and testing are the same.

The comparative results in Table 2.1 shows that our method outperforms others with substantial margins in all the four different experimental settings. It needs to be emphasized that in Experiment C and D, the superiority of our approach is extremely obvious when

Table 2.1: Comparisons of the prediction accuracy (mean absolute error in degree) of different methods. (i) and (ii) are training datasets from CMU-MultiPIE and BU-4DFE. (iii) and (iv) are testing datasets from CMU-MultiPIE and BU-4DFE. For our approach, we use (i) to train the 1D yaw estimator on the coarse layer and (ii) to train two 3D pose estimators on the fine layer. The results shows that the superiority of our approach is more obvious compared with others in C and D where the training and testing are carried out in different datasets. This fact highlights the strong generative capability of our approach to deal with out-of-sample testings inputs.

| Method<br>MAE±SD | Experiment A<br>Train on (i), Test on (iii) | Experiment B<br>Train on (ii), Test on (iv) |
|---|---|---|
| BME-Isomap [5] | 8.5 ± 6.3° | 12.5 ± 8.7° |
| BME-LLE [5] | 7.7 ± 5.4° | 12.4 ± 8.1° |
| BME-LL [5] | 7.1 ± 5.1° | 10.9 ± 7.6° |
| $SL^2$ [44] | 4.9 ± 3.7° | 6.3 ± 5.2° |
| PLS [36] | 6.2 ± 4.6° | 7.3 ± 5.1° |
| GPR [107] | 7.2 ± 5.6° | 9.7 ± 8.4° |
| SVM [9] + GPR [107] | 6.7 ± 5.1° | 7.8 ± 6.4° |
| Ours | **4.6** ± **3.9**° | **5.8** ± **5.4**° |

| Method<br>MAE±SD | Experiment C<br>Train on (i), Test on (iv) | Experiment D<br>Train on (ii), Test on (iii) |
|---|---|---|
| BME-Isomap [5] | 15.4 ± 10.1° | 10.4 ± 7.8° |
| BME-LLE [5] | 14.3 ± 9.6° | 10.1 ± 6.6° |
| BME-LL [5] | 12.6 ± 9.1° | 9.0 ± 6.3° |
| $SL^2$ [44] | 9.0 ± 6.5° | 7.2 ± 4.9° |
| PLS [36] | 8.7 ± 5.3° | 7.5 ± 6.2° |
| GPR [107] | 11.4 ± 9.8° | 8.1 ± 6.0° |
| SVM [9] + GPR [107] | 9.5 ± 7.1° | 6.5 ± 4.4° |
| Ours | **5.8** ± **5.4**° | **4.6** ± **3.9**° |

the training and testing are performed on distinct databases, see Figure 2.3 for a more clear comparison. This fact highlights the strong capability of our approach to handle extensive instance variations and its prominent generative ability, which is a merit of instance parameterization. Considering the fact that there are only limited number of images with annotations available during training which can not cover the huge instance variations in testing cases, the generative capability is important for robust performance on faces in-the-wild data. We will explain this point soon in next section.

The results shows that the cascade of multi-class SVM + GPR can boost the prediction accuracy compared with the case where only GPR is employed. Multi-class SVM that divides the pose variation space into more compact intervals, can significantly alleviate the ambiguity during the procedure of the non-linear mappings. However, the cascade

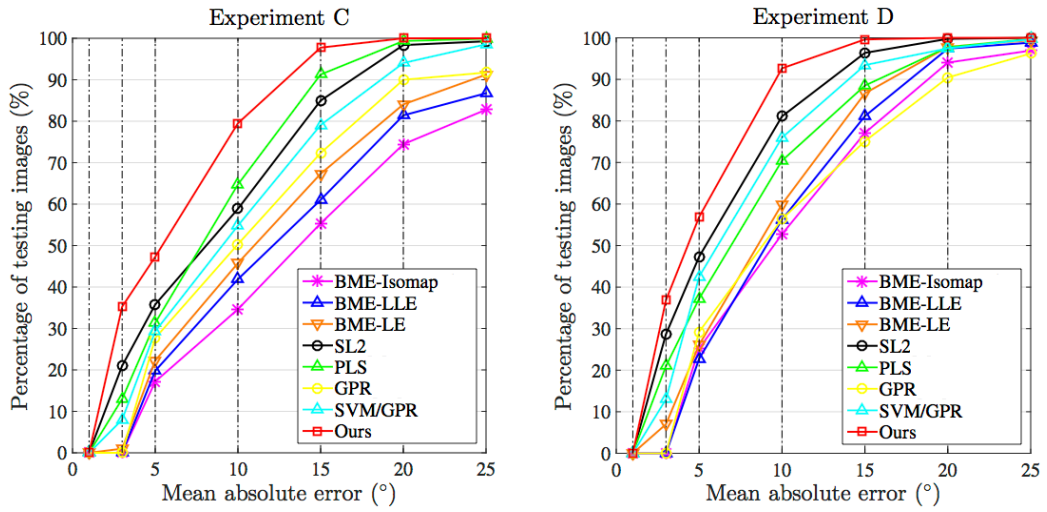



Figure 2.3: The fraction of estimation accuracy of Experiment C and D on CMU-MultiPIE and BU-4DFE. The x- and y-axis represent the Mean Absolute Error (MAE) and the fraction of images in the testing set. In Experiment C, all the compared methods except ours are trained in CMU-MultiPIE and tested in BU-4DFE. While in Experiment D, all the compared methods except ours are trained in BU-4DFE and tested in CMU-MultiPIE. Notice our method outperforms others with a steepest curve and fastest converging rate to 100%. It proves the better generalization ability of our approach regarding cross-database training and testing procedure.

Table 2.2: Comparisons of the performance of different approaches on AFW. The database contains only discrete pose annotations with 15° interval. We use CMU-MultiPIE and BU4DFE to train the estimators of all the methods expect TSPN. TSPN indep. and share. represent TSPN with indepedent model and fully shared model respectively. The continuous yaw predicitons are bucketed with discrete error tolerance ($\leqslant 15°$ and $\leqslant 30°$). Our method has the best performance in the terms of the fraction of testing images with prediction errors less then given tolerances.

| Algorithm | $MAE \leqslant 15°$ | $MAE \leqslant 30°$ |
|---|---|---|
| | Fraction (%) | Fraction (%) |
| BME [5] | 56.3 | 73.1 |
| $SL^2$ [44] | 73.5 | 81.4 |
| PLS [36] | 78.7 | 86.2 |
| Multi-class SVM [9] | 65.9 | 78.3 |
| TSPM indep. [158] | 81.0 | 84.0 |
| TSPM share. [158] | 76.9 | 83.0 |
| Ours | **86.3** | **92.7** |

performance is still inferior to ours due to the sensitiveness to noise and instance variations. We also notice that BMEs have very limited performance especially when the testing is performed on BU-4DFE (Experiment B and C) where 3D pose variations and multiple pose-unrelated factors exist. The manifold degradation in this case adversely affects the manifold embedding process, which highlights the significance of the proposed uniform geometry representation and mapping directions to effectively factorize pose-related and -unrelated factors.

### 2.4.3 Evaluation on Faces In the Wild

Experiments on faces in the wild are conducted to demonstrate the performance of different approaches on real-world data. The experiments on AFW are carried out as follows. We use the annotations of facial landmarks as face detector to crop faces out. Faces that are larger than $64 \times 64$ are preserved for experiments. This leaves totally 205 images with 459 faces in the testing set. Since there are only 1D discrete yaw annotations available in AFW, we use dataset (i) from CMU-MultiPIE and dataset (ii) from BU-4DFE together to train pose estimators for BME, $SL^2$, PLS, Multi-class SVM and our approach. For these estimators, the results of continuous yaw predictions are bucketed with a 15° or 30° spacing for performance evaluation. For TSPM, we use the reported results in [158] for both the independent model and the fully shared model.

Experimental results of different methods are compared in Table 2.2. It shows that our method outperforms other approaches in terms of the fraction of testing images with prediction errors less then given tolerances. The uniform factorization framework and the generative ability of our approach guarantees the robust and accurate pose prediction on the challenging unconstrained data, where drastic variations among identity, facial expression, illumination and other latent factors exists. Another significant advantage of our approach is that, unlike other estimators that can provide only 1- or 2-dimensional pose estimation, our estimators can provide full 3D pose prediction by the 3-sphere manifold embedding. Please refer to Figure 2.4 for examples of pose estimation results when applying our approach on AFW dataset in uncontrolled settings.

### 2.4.4 Validity of Instance Parametric Subspace

We carry out experiments to verify the validity of the proposed instance parametric subspace. 100 instances from CMU-MultiPIE are randomly selected to train a 1D yaw estimator. Each instance contains 13 images for a certain subject (identity) with one of the 6 facial expression under certain illumination condition. Since the estimator needs to provide continuous pose


Figure 2.4: Examples of the pose prediction of our approach on AFW. There are extensive variations on the identity, facial expression, illumination and background clutters. The robust performance proves the effectiveness of the proposed instance parameterization to handle multiple pose-unrelated instance variations in uncontrolled settings.

prediction, the number of mapping centers is set to $m = 15$. We increase the number of instance bases $\{I^1, \cdots, I^L\}$ incrementally with the order of decreasing eigenvalues. 10 new testing instances are picked out to evaluate the performance of the pose estimation.

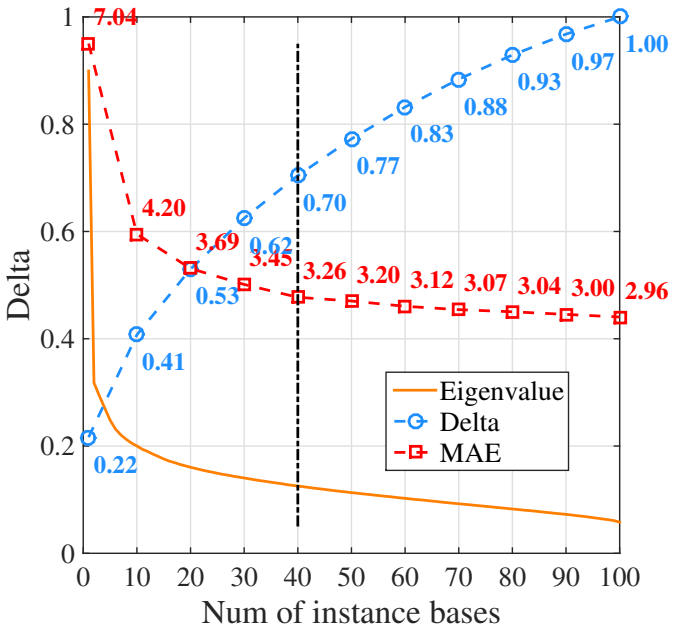



Figure 2.5: The relation between the prediction accuracy (y) and the number of instance bases (x). We can use the value of $\delta$ as a threshold to truncate instance bases with trivial eigenvalues. The remaining eigen instance bases can efficiently span the instance parametric subspace. The magenta dash line shows an example when $\delta = 0.7$, 40 out of all the 100 eigen instance bases are employed to span the subspace, and the MAE on the testing instances is 3.26°.

The relationship between the size of instance bases and prediction accuracy for the yaw estimator is showed in Figure 2.5. Quantitative results are given in Table 2.3. It shows that the MAE decreases fast when adding the first a few instance bases with significant eigenvalues. The MAE converges as instance bases with trivial eigenvalues are added. At the same time, the testing time and the trained model size grows linearly as the number of instance bases increased. We use $\delta$ as a threshold to truncate instance bases with trivial eigenvalues. For instance, $\delta = 0.6 \sim 0.7$ can best balance the trade-off between the prediction accuracy and the trained model size and testing time. This experiment proves that the eigen instance bases can effectively span the instance parametric subspace, which can significantly reduce the computational cost without sacrificing the accuracy.

Table 2.3: Quantitative results of Figure 2.5. The first column is the number of instance bases used to span the instance parametric space. The second column indicates the ration of space variation defined in 2.11. The third and the fourth columns show the average trained model size and testing time per image. The last column is prediction accuracy. We can refer that the instance parametric subspace spanned by 30 ~ 40 instance bases can best balance the prediction accuracy and the computational cost.

| Bases Number | Ratio $\delta$ | Model Size ($KB$) | Testing Time ($ms$) | MAE ±SD (°) |
|---|---|---|---|---|
| 1 | 0.216 | 1.1 | 18.5 | 7.04 ± 6.24 |
| 10 | 0.407 | 7.9 | 26.1 | 4.20 ± 4.15 |
| 20 | 0.529 | 15.5 | 32.7 | 3.69 ± 3.72 |
| **30** | **0.625** | **23.2** | **42.9** | **3.45 ± 3.54** |
| **40** | **0.704** | **29.4** | **56.3** | **3.26 ± 3.31** |
| 50 | 0.773 | 38.6 | 70.9 | 3.20 ± 3.26 |
| 60 | 0.832 | 45.2 | 91.3 | 3.12 ± 3.20 |
| 70 | 0.884 | 54.9 | 102.6 | 3.07 ± 3.11 |
| 80 | 0.929 | 67.2 | 124.0 | 3.04 ± 3.09 |
| 90 | 0.968 | 78.4 | 135.8 | 3.00 ± 3.07 |
| 100 | 1.000 | 85.9 | 158.3 | 2.96 ± 2.99 |

## 2.5 Discussion

In this study, we presented a novel head pose estimation approach. We propose the instance parametric subspace to handle multiple instance variations in a generative way. The coarse-to-fine framework, which employs a unit circle on the coarse layer and a 3-sphere on the fine layer to model the uniform geometry representation, can significantly alleviate the manifold degradation problem by learning instance-dependent nonlinear mappings in an unconventional direction. Experiments on both experimental and in-the-wild databases demonstrate the superior performance of our approach compared with state-of-the-arts in terms of prediction accuracy. A limitation of our approach is that our current approach relies on an external face detector to locate the face. In future work, we plan to combine the face detection and pose estimation together in a unified system for complete face detection and head pose estimation.

# Chapter 3

# Facial Landmark Tracking

We propose a novel method for real-time face alignment in videos based on a recurrent encoder-decoder network model. Our proposed model predicts 2D facial point heat maps regularized by both detection and regression loss, while uniquely exploiting recurrent learning at both spatial and temporal dimensions. At the spatial level, we add a feedback loop connection between the combined output response map and the input, in order to enable iterative coarse-to-fine face alignment using a *single network model*, instead of relying on traditional cascaded model ensembles. At the temporal level, we first decouple the features in the bottleneck of the network into *temporal-variant factors*, such as pose and expression, and *temporal-invariant factors*, such as identity information. Temporal recurrent learning is then applied to the decoupled temporal-variant features. We show that such feature disentangling yields better generalization and significantly more accurate results at test time. We perform a comprehensive experimental analysis, showing the importance of each component of our proposed model, as well as superior results over the state of the art and several variations of our method in standard datasets. [1]

[1]The project page: https://sites.google.com/site/xipengcshomepage/eccv2016

## 3.1 Introduction

Face landmark detection plays a fundamental role in many computer vision tasks, such as face recognition/verification, expression analysis, person identification, and 3D face modeling. It is also the basic technology component for a wide range of applications like video surveillance, emotion recognition, augmented reality on faces, etc. In the past few years, many methods have been proposed to address this problem, with significant progress being made towards systems that work in real-world conditions ("in the wild").

Multiple lines of research have been explored for face alignment in last two decades. Early research includes methods based on active shape models (ASMs) [19, 81] and active appearance models (AAMs) [30]. ASMs iteratively deform a shape model to the target face image, while AAMs impose both shape and object appearance constraints in the optimization process. Recent advances in the field are largely driven by *regression-based techniques* [145, 15, 153, 63, 154]. These methods usually take advantage of large-scale annotated training sets (lots of faces with labeled landmark points), achieving accurate results by learning discriminative regression functions that directly map facial appearance to landmark coordinates. The features extracted for regressing landmarks can be either hand-crafted features [145, 15], or features extracted from convolutional neural networks [153, 63, 154]. Although these methods can achieve very reliable results in standard benchmark datasets, they still suffer from limited performance in challenging scenarios, e.g., involving large face pose variations and heavy occlusions.

A promising direction to address these challenges is to consider video-based face alignment (i.e., sequential face landmark detection) [121], leveraging temporal information and identity consistency as additional constraints [139]. Despite the long history of research in rigid and non-rigid face tracking [10, 88, 21, 91], current efforts have mostly focused on face alignment in still images [113, 153, 136, 156]. When videos are considered as input, most methods perform landmark detection by independently applying models trained on still images in each frame in a tracking-by-detection manner [140], with notable exceptions such

as [2, 102], which explore incremental learning based on previous frames. These methods do not take full advantage of the temporal information to predict face landmarks for each frame. How to effectively model long-term temporal constraints while handling large face pose variations and occlusions is an open research problem for video-based alignment.

In this work, we address this problem by proposing a novel recurrent encoder-decoder deep neural network model (see Figure 3.1), named as ***RED-Net***. The encoding module projects image pixels into a low-dimensional feature space, whereas the decoding module maps features in this space to 2D facial point maps, which are further regularized by a regression loss.

Our encoder-decoder framework allows us to explore spatial refining of our landmark prediction results, in order to handle faces with large pose variations. More specifically, we introduce a feedback loop connection between the aggregated 2D facial point maps and the input. The intuition is similar to cascading multiple regression functions [145, 153] for iterative coarse-to-fine face alignment, but in our approach the iterations are modeled jointly with shared parameters, using a single network model. It provides significant parameter reduction when compared to traditional methods based on cascaded neural networks. A recurrent structure also avoids the effort to explicitly divide the task into multiple stage prediction problems. This subtle difference makes the recurrent model more elegant in terms of holistic optimization. It can implicitly track the prediction behavior in different iterations for a specific face example, while cascaded predictions can only look at the immediate previous cascade stage. Our design also shares the same spirit of residual networks [39]. By adding feedback connections from the predicted heatmap, the network only needs to implicitly predict the residual from previous predictions in subsequent iterations, which is arguably easier and more effective than directly predicting the absolute landmark locations.

For more effective temporal modeling, we first decouple the features in the bottleneck of the network into temporal-variant factors, such as pose and expression, and temporal-invariant factors, such as identity. We disentangle the features into two components,

where one component is used to learn face recognition using identity labels, and the other component encodes temporal-variant factors. To utilize temporal coherence in our framework, we apply recurrent temporal learning to the temporal-variant component. We used *Long Short Term Memory (LSTM)* to implicitly abstract motion patterns by looking at multiple successive video frames, and use this information to improve landmark fitting accuracy. Landmarks with large pose variation are typically outliers in a landmark training set. By looking at multiple frames, it helps to reduce the inherent prediction variance.

We show in our experiments that our encoder-decoder framework and its recurrent learning in both spatial and temporal dimensions significantly improve the performance of sequential landmark localization. In summary, our work makes the following **contributions**:

- We propose a novel recurrent encoder-decoder network model for real-time sequential face landmark detection. To the best of our knowledge, this is the first time a recurrent model is investigated to perform video-based facial landmark detection.

- Our proposed *spatial recurrent learning* enables a novel iterative coarse-to-fine face alignment using a single network model. This is critical to handle large face pose changes and a more effective alternative than cascading multiple network models in terms of accuracy and memory footprint.

- Different from traditional methods, we apply *temporal recurrent learning* to temporal-variant features which are decoupled from temporal-invariant features in the bottle-neck of the network, achieving better generalization and more accurate results.

- We provide a detailed experimental analysis of each component of our model, as well as insights about key contributing factors to achieve superior performance over the state of the art.

## 3.2 Related Work

Face alignment has a long history of research in computer vision. Here we briefly discuss face alignment works related to our approach, as well as advances in deep learning, like the development of recurrent and encoder-decoder neural networks.

**Regression-based face landmark detection.** Recently, regression-based face landmark detection methods [3, 128, 145, 15, 153, 2, 156, 136, 52, 144, 159] have achieved significant boost in the generalization performance of face landmark detection, compared to algorithms based on statistical models such as Active shape models [19, 81] and Active appearance models [30]. Regression-based approaches directly regress landmark locations based on features extracted from face images. Landmark models for different points are learned either in an independent manner or in a joint fashion [15]. When all the landmark locations are learned jointly, implicit shape constraints are imposed because they share the same or partially the same regressors. This paper performs landmark detection via both a classification model and a regression model. Different from most previous methods, this work deals with face alignment in a video. It jointly optimizes detection output by utilizing multiple observations from the same person.

**Cascaded models for landmark detection.** Additional accuracy improvement in face landmark detection performance can be obtained by learning cascaded regression models. Regression models from earlier cascade stages learn coarse detectors, while later cascade stages refine the result based on early predictions. Cascaded regression helps to gradually reduce the prediction variance, thus making the learning task easier for later stage detectors. Many methods have effectively applied cascade-like regression models for the face alignment task [145, 128, 153]. The supervised descent method [145] learns cascades of regression models based on SIFT features. Sun *et. al.* [128] proposed to use three levels of neural networks to predict landmark locations. Zhang *et. al.* [153] studied the problem via cascades of stacked auto-encoders which gradually refine the landmark position with higher resolution inputs. Compared to these efforts which explicitly define cascade structures, our

method learns a spatial recurrent model which implicitly incorporates the cascade structure with shared parameters. It is also more "end-to-end" compared to previous works that divide the learning process into multiple stages.

**Face alignment in videos.** Most face alignment algorithms utilize temporal information by initializing the location of landmarks with detection results from the previous frame, performing alignment in a tracking-by-detection fashion [140]. Asthana *et. al.* [2] and Peng *et. al.* [102] proposed to learn a person specific model using incremental learning. However, incremental learning (or online learning) is a challenging problem [132], as the incremental scheme has to be carefully designed to prevent model drifting [133]. In our framework, we do not update our model online. All the training is performed offline and we expect our LSTM unit to capture landmark motion correlations.

**Recurrent neural networks.** Recurrent neural networks (RNNs) are widely employed in the literature of speech recognition [80] and natural language processing [79]. They have also been recently used in computer vision. For instance, in the tasks of image captioning [56] and video captioning [147], RNNs are usually employed for text generation. RNNs are also popular as a tool for action classification. As an example, Veeriah *et. al.* [138] use RNNs to learn complex time-series representations via high-order derivatives of states for action recognition.

**Encoder-decoder networks** Encoder and decoder networks are well studied in machine translation [17] where the encoder learns the intermediate representation and the decoder generates the translation from the representation. It is also investigated in speech recognition [74] and computer vision [4, 41]. Yang *et. al.* [146] proposed to decouple identity units and pose units in the bottleneck of the network for 3D view synthesis. However, how to fully utilize the decoupled units for correspondence regularization [73] is still unexplored. In this work, we employ the encoder to learn a joint representation for identity, pose, expression as well as landmarks. The decoder translates the representation to landmark heatmaps. Our spatial recurrent model loops the whole encoder-decoder framework.

## 3.3 Method

The task is to locate facial landmarks in sequential images using an end-to-end deep neural network. Figure 3.1 shows an overview of our approach. The network consists of a series of nonlinear and multi-layered mappings, which can be functionally categorized as four modules: **(1)** encoder-decoder $f_{enc}$ and $f_{dec}$, **(2)** spatial recurrent learning $f_{srn}$, **(3)** temporal recurrent learning $f_{trn}$, and **(4)** constrained identity disentangling $f_{cls}$. Details of the novelty are described in following sections.

### 3.3.1 Encoder-Decoder

The input of the encoder-decoder is a single video frame $\mathbf{x} \in \mathbb{R}^{W \times H \times 3}$ and the output is a response map $\mathbf{z} \in \mathbb{R}^{W \times H \times C_z}$ which indicates landmark locations. $C_z = 7$ or 68 depending on the number of landmarks to be predicted.

The *encoder* performs a sequence of convolution, pooling and batch normalization [50] to extract a low-dimensional representation $\mathbf{e}$ from both $\mathbf{x}$ and $\mathbf{z}$:

$$\mathbf{e} = f_{enc}\big(\mathbf{x}, \mathbf{z}; \theta_{enc}\big), \;\; f_{enc} : \mathbb{R}^{W \times H \times C} \rightarrow \mathbb{R}^{W_e \times H_e \times C_e}, \tag{3.1}$$

where $f_{enc}\big(\cdot; \theta_{enc}\big)$ denotes the encoder mapping with parameters $\theta_{enc}$. We concatenate $\mathbf{x}$ and $\mathbf{z}$ along the channel dimension thus $C = 3 + C_z$. The concatenation is fed into the encoder as an updated input.

Symmetrically, the *decoder* performs a sequence of unpooling, convolution and batch normalization to upsample the representation code to the response map:

$$\mathbf{z} = f_{dec}(\mathbf{e}; \theta_{dec}), \;\; f_{dec} : \mathbb{R}^{W_e \times H_e \times C_e} \rightarrow \mathbb{R}^{W \times H \times C_z}, \tag{3.2}$$

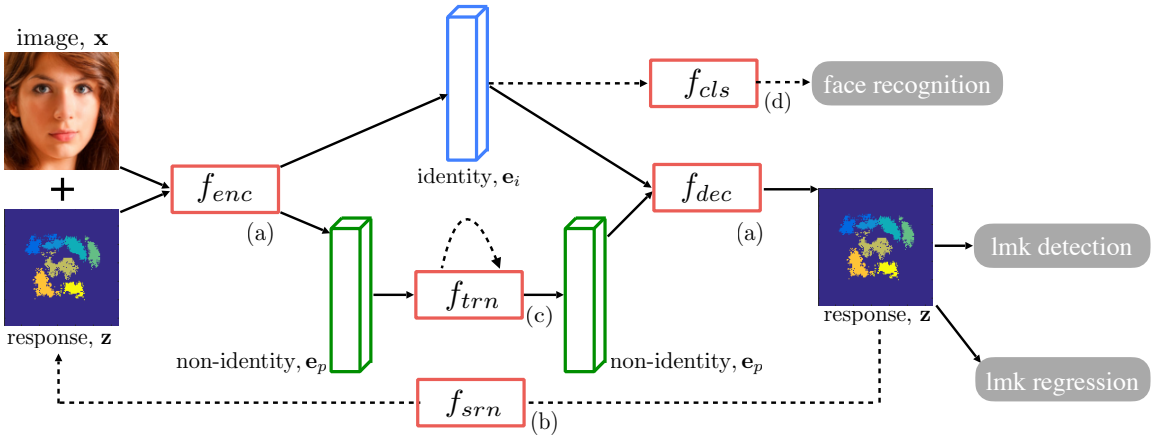



Figure 3.1: Overview of the recurrent encoder-decoder network: **(a)** encoder-decoder (Section 3.3.1); **(b)** spatial recurrent learning (Section 3.3.2); **(c)** temporal recurrent learning (Section 3.3.3); and **(d)** supervised identity disentangling (Section 3.3.4). $f_{enc}, f_{dec}, f_{srn}, f_{trn}, f_{cls}$ are potentially nonlinear and multi-layered mappings.

where $f_{dec}(\cdot;\theta_{dec})$ denotes the decoder mapping with parameters $\theta_{dec}$. $\mathbf{z}$ has the same $W \times H$ dimension as $\mathbf{x}$ but $C_z$ channels for $C_z$ landmarks. Each channel presents pixel-wise confidences of the corresponding landmark.

The encoder-decoder design plays an important role in our task. **First**, the decoder's output $\mathbf{z}$ has the same resolution (but a different number of channels) as the input image $\mathbf{x}$. Thus it is easy to directly concatenate $\mathbf{z}$ with $\mathbf{x}$ along the channel dimension. The concatenation provides pixel-wise spatial cues to update the landmark prediction by the proposed *spatial recurrent learning* ($f_{srn}$). We will explain it soon in Section 3.3.2.

**Second**, the encoder-decoder network can achieve a low-dimensional representation $\mathbf{e}$ in the bottleneck. We can utilize the domain prior to decouple $\mathbf{e}$ into two parts: the identity code $\mathbf{e}_i$, which is temporal-invariant as we are tracking the same person; and the non-identity code $\mathbf{e}_p$, which models temporal-variant factors [97, 96] such as head pose, expression, illumination, and etc.

In Section 3.3.3, we propose the *temporal recurrent learning* ($f_{trn}$) to model the changes of $\mathbf{e}_p$. In Section 3.3.4, we show how to speed up the network training by carrying out the *supervised identity disentangling* ($f_{cls}$) on $\mathbf{e}_i$.

**Third**,the encoder-decoder network enables a fully convolutional design. The bottleneck embedding $\mathbf{e}$ and output response map $\mathbf{z}$ are feature maps instead of fully-connected neurons

that are often used in ordinary convolutional neural networks. This design is highly memory-efficient and can significantly speed up the training and testing [72], which is preferred by video-based applications.

### 3.3.2 Spatial Recurrent Learning

The purpose of spatial recurrent learning is to pinpoint landmark locations in a coarse-to-fine manner. Unlike existing approaches [128, 153] that employ multiple networks in cascade, we accomplish the coarse-to-fine search in a single network in which the parameters are jointly learned in successive recurrent steps.

The spatial recurrent learning is performed by iteratively feeding back the previous prediction, stacked with the image as shown in Figure 3.2, to eventually push the shape prediction from an initial guess to the ground truth:

$$\mathbf{z}_k = f_{srn}\big(\mathbf{x}, \mathbf{z}_{k-1}; \theta_{srn}\big), \;\; k = 1, \cdots, K \tag{3.3}$$

where $f_{srn}\big(\cdot; \theta_{srn}\big)$ denotes the spatial recurrent mapping with parameters $\theta_{srn}$. $\mathbf{z}_0$ is the initial response map, which could be a response map generated by the mean shape or the output of the previous frame.

In our conference version [94], detection-based supervision is performed in every recurrent step. It is robust to appearance variations but lacks precision, because pixels within a certain radius around the ground-truth location are labeled using the same value. To address this limitation, motivated by [13], we propose to further explore the spatial recurrent learning by performing detection-followed-by-regression in successive steps.

Specially, we carry out a two-step recurrent learning by setting $K = 2$. The first step performs *landmark detection* that aims to locate 7 major facial components (*i.e.* $C = 7$ in Equation (3.2)). The second step performs *landmark regression* that refines all 68 landmarks

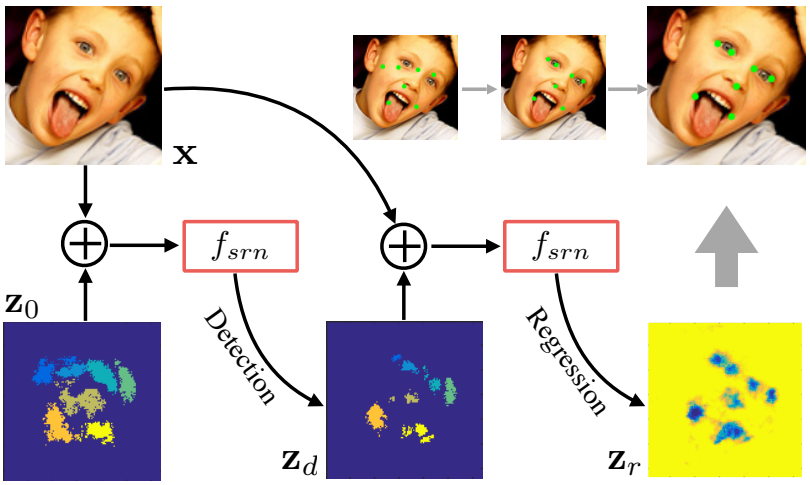



Figure 3.2: An unrolled illustration of *spatial recurrent learning*. The response map is pretty coarse when the initial guess is far away from the ground truth if large pose and expression exist. It eventually gets refined in the successive recurrent steps.

positions (*i.e.* $C = 68$). For clarity, we use $C_d$ and $C_r$ to denote the number of channels output by the detection and the regression steps, respectively.

The landmark detection step guarantees fitting robustness especially in large pose and partial occlusions. The encoder-decoder aims to output a binary map of $C_d$ channels, one for each major facial component. The detection step outputs:

$$\mathbf{z}_d = f_{dec}\big(f_{enc}(\mathbf{x}, \mathbf{z}_0; \theta_{enc}); \theta_{dec}\big), \ \mathbf{z}_d \in \mathbb{R}^{W \times H \times C_d}, \tag{3.4}$$

where the detection task can be trained using pixel-wise sigmoid cross-entropy loss function:

$$\ell_d = \frac{1}{M_d} \sum_{c=1}^{C_d} \sum_{i=1}^{W} \sum_{j=1}^{H} z_{ij}^c log y_{ij}^c + (1 - z_{ij}^c) log(1 - y_{ij}^c) \tag{3.5}$$

where $M_d = C_d \times W \times H$. Here $z_{ij}^c$ denotes the sigmoid output at pixel location $(i, j)$ in $\mathbf{z}_d$ for the $c$-th landmark. $y_{ij}^c$ is the ground-truth label at the same location, which is set to 1 to mark the presence of the corresponding landmark and 0 for the remaining background.

Note that this loss function is different from the N-way cross-entropy loss used in our previous conference paper [94]. It allows multiple class labels for a single pixel, which helps to tackle the landmark overlaps.

The landmark regression step improves the fitting accuracy from the outputs of the previous detection step. The encoder-decoder aims to output a heatmap of $C_r$ channels, one for each landmark. The regression step outputs:

$$\mathbf{z}_r = f_{dec}\big(f_{enc}(\mathbf{x}, \mathbf{z}_{det}; \theta_{enc}); \theta_{dec}\big),\ \mathbf{z}_r \in \mathbb{R}^{W \times H \times C_r}, \tag{3.6}$$

where the regression task can be trained using pixel-wise $L_2$ loss function:

$$\ell_r = \frac{1}{M_r} \sum_{c=1}^{C_r} \sum_{i=1}^{W} \sum_{j=1}^{H} \|z_{ij}^c - y_{ij}^c\|_2^2, \tag{3.7}$$

where $M_r = C_d \times W \times H$. Here $z_{ij}^c$ denotes the heatmap value of the $c$-th landmark at pixel location $(i, j)$ in $\mathbf{z}_r$ for the $c$-th landmark. $y_{ij}^c$ is the ground-truth value at the same location, which obeys a Gaussian distribution centered at the landmark with a pre-defined standard deviation.

Now the spatial recurrent learning (Equation (3.3)) can be achieved by minimizing the detection loss (Equation (3.5)) and the regression loss (Equation (3.7)), simultaneously:

$$\underset{\theta_{enc}, \theta_{dec}}{\operatorname{argmin}}\ \ell_d + \lambda \ell_r, \tag{3.8}$$

where $\lambda$ balances the loss between the two tasks. Note that the spatial recurrent learning do not introduce new parameters but sharing the same parameters of the encoder-decoder network, *i.e.* $\theta_{srn} = \{\theta_{enc}, \theta_{dec}\}$.

The spatial recurrent learning is highly memory efficient. It is capable of end-to-end training, which is a significant advantage compared with the cascade framework [13]. More importantly, the network can jointly learn the coarse-to-fine fitting strategy in recurrent steps, instead of training cascaded networks independently [128, 153], which guarantees robustness and accuracy in challenging conditions.

### 3.3.3 Temporal Recurrent Learning

In addition to the spatial recurrent learning, we also propose a temporal recurrent learning to model factors, *e.g.* head pose, expression, and illumination, that may change over time. These factors affect the landmark locations significantly. Thus we can expect improved tracking accuracy by modeling their temporal variations.

As mentioned in Section 3.3.1, the bottleneck embedding $\mathbf{e}$ can be decoupled into two parts: the identity code $\mathbf{e}_i$ and the non-identity code $\mathbf{e}_p$:

$$\mathbf{e}_i \in \mathbb{R}^{W_e \times H_e \times C_i}, \mathbf{e}_p \in \mathbb{R}^{W_e \times H_e \times C_p}, C_e = C_i + C_p, \tag{3.9}$$

where $\mathbf{e}_i$ and $\mathbf{e}_p$ model the temporal-invariant and -variant factors, respectively. We leave $\mathbf{e}_i$ to Section 3.3.4 for additional identity supervision, and exploit variations of $\mathbf{e}_p$ via the recurrent model. Please refer to Figure 3.3 for an unrolled illustration of the proposed temporal recurrent learning.

Mathematically, given $T$ successive video frames $\{\mathbf{x}^t; t = 1, \cdots, T\}$, the encoder extracts a sequence of embeddings $\{\mathbf{e}_i^t, \mathbf{e}_p^t; t = 1, \cdots, T\}$. Our goal is to achieve a nonlinear mapping $f_{trn}$, which simultaneously tracks a latent state $h^t$ and updates $\mathbf{e}_p^t$ at time $t$:

$$\begin{aligned} h^t &= p(\mathbf{e}_p^t, h^{t-1}; \theta_{trn}), \quad t = 1, \cdots, T \\ \mathbf{e}_p^{t*} &= q(h^t; \theta_{trn}), \end{aligned} \tag{3.10}$$

where $p(\cdot)$ and $q(\cdot)$ are functions of $f_{trn}(\cdot; \theta_{trn})$ with parameters $\theta_{trn}$. $\mathbf{e}_p^{t*}$ is the update of $\mathbf{e}_p^t$.

The temporal recurrent learning is trained using $T$ successive frames. At each frame, the detection and regression tasks are performed for the spatial recurrent learning. The recurrent learning is performed by minimizing Equation (3.8) at every time step $t$:

$$\underset{\theta_{enc}, \theta_{dec}, \theta_{trn}}{\operatorname{argmin}} \sum_{t=1}^{T} \ell_d^t + \lambda \ell_r^t, \tag{3.11}$$

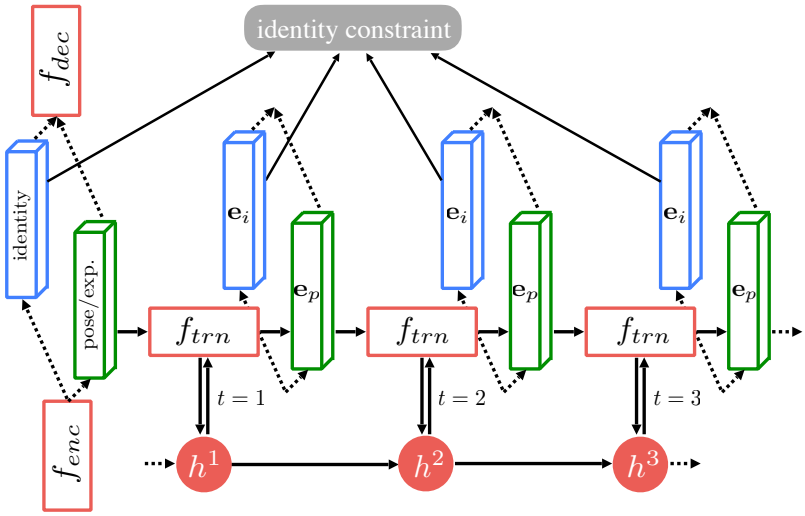



Figure 3.3: An unrolled illustration of *temporal recurrent learning*. $C_i$ encodes temporal-invariant factor which subjects to the same identity constraint. $C_p$ encodes temporal-variant factors which is further modeled in $f_{trn}$.

where $\theta_{trn}$ denotes network parameters of the temporal recurrent learning, *e.g.* parameters of LSTM units. It is worth mentioning that, we perform recurrent learning in both spatial and temporal dimensions by jointly optimizing $\{\theta_{enc}, \theta_{dec}, \theta_{trn}\}$ in Equation (3.11).

The temporal recurrent module is memorizing as well as modeling the changing pattern of the temporal-variant factors. Our experiments indicated that the offline learned model can significantly improve the online fitting accuracy and robustness, especially when large variations or partial occlusions happen.

### 3.3.4 Supervised Identity Disentangling

There is no guarantee that temporal-invariant and -variant factors can be completely decoupled in the bottleneck by simply splitting the bottleneck representation $\mathbf{e}$ into two parts. More supervised information is required to achieve the disentangling. To address this issue, we propose to apply a face recognition task on the identity code $\mathbf{e}_i$, in addition to the temporal recurrent learning applied on non-identity code $\mathbf{e}_p$.

The supervised identity disentangling is formulated as an $N$-way classification problem. $N$ is the number of unique individuals present in the training sequences. In general, we associate the identity representation $\mathbf{e}_i$ with a one-hot encoding $\mathbf{z}_i$ to indicate the score of

each identity:

$$\mathbf{z}_i = f_{cls}(\mathbf{e}_i; \theta_{cls}),\ f_{cls} : \mathbb{R}^{W_e \times H_e \times C_i} \rightarrow \mathbb{R}^N, \tag{3.12}$$

where $f_{cls}(\cdot; \theta_{cls})$ is the identity classification mapping with parameters $\theta_{cls}$. The identity task is trained using $N$-way cross-entropy loss:

$$\ell_{cls} = \frac{1}{N} \sum_{n=1}^{N} z^n log y^n + (1 - z^n) log(1 - y^n), \tag{3.13}$$

where $z^n$ denotes the softmax activation of the $n$-th element in $\mathbf{z}_i$. $y^n$ is the $n$-th element of the identity annotation $\mathbf{y}_i$, which is a one-hot vector with a 1 for the correct identity and all 0s for others.

Now we can jointly train all the three tasks, *i.e.* $f_{srn}$, $f_{trn}$, and $f_{cls}$. Based on Equation (3.11) and (3.13), we simultaneously minimize the detection and regression loss together with the identity loss at every time step $t$:

$$\underset{\theta_{enc}, \theta_{dec}, \theta_{trn}, \theta_{cls}}{\operatorname{argmin}} \sum_{t=1}^{T} \ell_{det}^t + \lambda \ell_{reg}^t + \gamma \ell_{cls}^t, \tag{3.14}$$

where $\gamma$ weights the identity constraint. An obvious advantage of our approach is that the whole network can be trained end-to-end by optimizing all parameters $\{\theta_{enc}, \theta_{dec}, \theta_{trn}, \theta_{cls}\}$ simultaneously, which guarantees an efficient learning.

It has been shown in [154] that learning the face alignment task together with correlated tasks, *e.g.* head pose, can improve the fitting performance. We have a similar observation when adding face recognition task to the alignment task. More importantly, we find that the additional identity task can effectively speed up the training of the entire encoder-decoder network. In addition to more supervision, the identity task helps to decouple the identity and non-identity factors more completely, which facilitates the training of the temporal recurrent learning.

## 3.4 Network Architecture

We present the architecture details of proposed modules: $f_{enc/dec}$, $f_{srn}$, $f_{trn}$, and $f_{cls}$. All the four modules are designed in a single network that can be trained end-to-end. We introduce two variant designs of $f_{enc/dec}$, based on which $f_{srn}$, $f_{trn}$, and $f_{cls}$ are designed accordingly.

### 3.4.1 The Design of $f_{enc}$ and $f_{dec}$

To best evaluate the proposed method, we investigate two variant designs of the encoder-decoder: VGGNet [122] based and ResNet [39] based. The VGGNet-based design has a symmetrical structure between the encoder and decoder; while the ResNet-based design has an asymmetrical structure due to the usage of the residual modules.

**VGGNet-based design.** Table 3.1 presents the network specification. Figure 3.4 (left) shows the network architecture. The encoder is designed based on a variant of the VGG-16 network [122, 57]. It has 13 convolutional layers of constant $3 \times 3$ filters. We can, therefore, initialize the training process from weights trained on large datasets for object classification. We remove all fully connected layers in favor of a fully convolutional manner [72], which can effectively reduce the number of parameters from 117M to 14.8M [4]. The bottleneck feature maps are split into two parts for the identity and non-identity codes, respectively. This design preserves rich spatial information in 3D feature maps rather than 1D feature vectors, which is important for landmark localization.

We use max-pooling to halve the feature resolution at the end of each convolutional block. The pooling window size is $2 \times 2$ and the stride is 2. Although max-pooling can help to achieve translation invariance, it would cause a considerable loss of spatial information especially when multiple max-pooling layers are applied in a cascade. To solve this issue, we use a 2-bit code to record the index of the maximum activation selected in a $2 \times 2$ pooling window [152]. As illustrated in Figure 3.4 (right), the memorized index is then used in the corresponding unpooling layer to place each activation back to its original

Table 3.1: Specification of the VGGNet-based $f_{enc/dec}$ design: block name (**Top**), feature map dimension (**Middle**), and layer configuration (**Bottom**). $[3 \times 3, 64]$ means there are 64 filters (channels), each has a size of $3 \times 3$. Pooling or unpooling operations are performed after or before each module. The pooling window is $2 \times 2$ with a stride of 2.

| $A_0$ | $A_1$ | $A_2$ | $A_3$ | $A_4$ |
|---|---|---|---|---|
| $128 \times 128$ | $64 \times 64$ | $32 \times 32$ | $16 \times 16$ | $8 \times 8$ |
| $2\times$ conv $[3 \times 3, 64]$ pooling | $2\times$ conv $[3 \times 3, 128]$ pooling | $3\times$ conv $[3 \times 3, 256]$ pooling | $3\times$ conv $[3 \times 3, 512]$ pooling | $3\times$ conv $[3 \times 3, 512]$ - |
| | $B_4$ | $B_3$ | $B_2$ | $B_1$ |
| | $16 \times 16$ | $32 \times 32$ | $64 \times 64$ | $128 \times 128$ |
| | unpooling $3\times$ conv $[3 \times 3, 512]$ | unpooling $3\times$ conv $[3 \times 3, 512]$ | unpooling $3\times$ conv $[3 \times 3, 256]$ | unpooling $2\times$ conv $[3 \times 3, 128]$ |

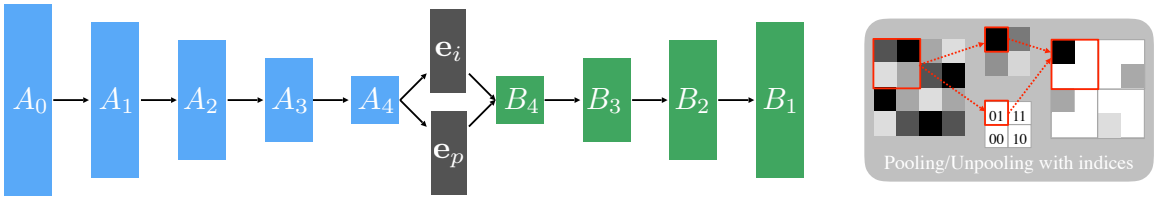



Figure 3.4: **Left:** the architecture of the VGGNet-based $f_{enc/dec}$ design. The encoder ($A_{0-4}$) and the decoder ($B_{4-1}$) are nearly symmetrical except that $f_{enc}$ has one more block $A_0$. $A_0$ downsamples the input image from $256 \times 256$ to $128 \times 128$. So $\mathbf{x}$ and $\mathbf{z}$ have the same resolution and can be easily concatenated along the channel dimension. **Right:** an illustration of the pooling/unpooing with indices. The corresponding pooling and unpooling share pooling indices using a 2-bit switch in each $2 \times 2$ pooling window.

location. This strategy is particularly useful when the decoder recovers the input structure from highly compressed feature maps. Besides, it is more efficient to store spatial indices than to memorize entire feature maps of float precision as indicated in FCNs [72].

The decoder is nearly symmetrical to the encoder with a mirrored configuration but replacing all max-pooling with unpooling layers. The encoder is slightly deeper than the decoder with one more covolutional block $A_0$ at the beginning. $A_0$ downsamples the input image from $256 \times 256$ to $128 \times 128$. So $\mathbf{x}$ and $\mathbf{z}$ have the same resolution and can be easily concatenated along the channel dimension. We find that batch normalization [50] can significantly boost the training speed since it reduces internal shifting in the mini batch. Thus, we apply batch normalization as well as rectified linear unit (ReLU) [85] after each convolutional layer.

**ResNet-based design.** Table 3.2 presents the network specification. Figure 3.5 (left) shows the network architecture. The encoder is designed based on a variant of the ResNet-

Table 3.2: Specification of the ResNet-based $f_{enc/dec}$ design: block name (**Top**), feature map dimension (**Middle**), and layer configuration (**Bottom**). We use conv/decov layers with a stride of 2 to halve or double the feature map dimensions. Thus no pooling/unpooling layer is used. The skip connections $E_{1-3}$ are specified in Table 3.3.

| $C_0$ | $C_1$ | $C_2$ | $C_3$ | $C_4$ |
|---|---|---|---|---|
| $128 \times 128$ | $64 \times 64$ | $32 \times 32$ | $16 \times 16$ | $8 \times 8$ |
| $1\times$ conv<br>$\begin{bmatrix} 7 \times 7, 64 \\ \text{strid, } 2 \end{bmatrix}$ | $3\times$ conv<br>$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix}$ | $8\times$ conv<br>$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix}$ | $36\times$ conv<br>$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix}$ | $3\times$ conv<br>$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix}$ |
| | $D_4$ | $D_3$ | $D_2$ | $D_1$ |
| | $16 \times 16$ | $32 \times 32$ | $64 \times 64$ | $128 \times 128$ |
| | $1\times$ dconv<br>$\begin{bmatrix} 2 \times 2, 512 \\ \text{stride, } 2 \\ 1 \times 1, 1024 \end{bmatrix}$ | $1\times$ dconv<br>$\begin{bmatrix} 2 \times 2, 256 \\ \text{stride, } 2 \\ 1 \times 1, 512 \end{bmatrix}$ | $1\times$ dconv<br>$\begin{bmatrix} 2 \times 2, 128 \\ \text{stride, } 2 \\ 1 \times 1, 256 \end{bmatrix}$ | $1\times$ dconv<br>$\begin{bmatrix} 2 \times 2, 64 \\ \text{stride, } 2 \\ 1 \times 1, 128 \end{bmatrix}$ |

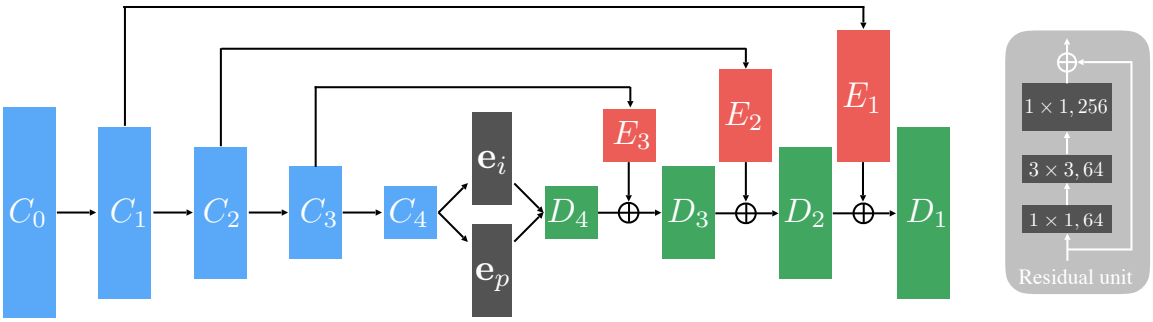



Figure 3.5: **Left:** the architecture of ResNet-based $f_{enc/dec}$ design (**Left**). The encoder ($C_{0-4}$) and the decoder ($D_{4-1}$) are asymmetrical. $f_{enc}$ is much deeper than $f_{dec}$, *i.e.* 151 vs. 4 layers. $C_0$ downsamples the input image from $256 \times 256$ to $128 \times 128$. Skip connections ($E_{1-3}$) are used to bridge hierarchical spatial information at different resolutions. **Right:** an example of residual unit used in $C_1$. $1 \times 1$ convolutional layers are used in the residual unit to cut down the number of filter parameters.

152 [39], which has 50 residual units of totally 151 convolutional layers. Figure 3.5 (right) shows a residual unit used in $C_1$. $1 \times 1$ convolutional layers are used to cut down the number of filter parameters. According to [39], the residual shortcut guarantees efficient training of the very deep network, as well as improved performance compared with vanilla design [122]. Stride-2 convolutions instead of max poolings are used to halve the feature map resolution at the end of each block.

Different from the VGGNet-based design, the encoder and decoder are asymmetrical. The encoder is much deeper than the decoder, and the decoder has only 4 upsampling blocks of totally 4 convolutional layers. A practical consideration behind this design is that the encoder has to tackle a complicated task, *e.g.* understand the image and translate it to a

Table 3.3: Specification of the skip connections. Note that $E_3$ and $C_1$, $E_2$ and $C_2$, $E_1$ and $C_1$ share the same configurations. The bridged features are directly added to the outputs of $D_{4-1}$ at the corresponding resolutions.

| $E_3$ | $E_2$ | $E_1$ |
|---|---|---|
| $16 \times 16$ | $32 \times 32$ | $64 \times 64$ |
| $3\times$ conv $\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix}$ | $3\times$ conv $\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix}$ | $3\times$ conv $\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix}$ |

low-dimensional embedding, while the decoder's task is relatively simpler, *e.g.* recover a set of response maps to mark landmark locations from the embedding. We use stride-2 de-convolutions to double the feature map resolution in each block. Similar to the VGGNet-based design, an additional convolutional block $C_0$ is used to downsample the inpuy image from $256 \times 256$ to $128 \times 128$. So $\mathbf{x}$ and $\mathbf{z}$ have the same resolution for an easy channel-wise concatenation.

Another difference between the ResNet-based design and the VGGNet-based design is the usage of skip connections $E_{3-1}$ [87] as shown in Figure 3.5 and specified in Table 3.3. These skip connections are used to bridge hierarchical spatial information between the encoder and decoder at different resolutions. They provide shortcuts of the gradient flow in backpropagation for efficient training. Besides, they also enable us to use a shallow decoder design since rich spatial information can be delivered through the shortcuts.

### 3.4.2 The Design of $f_{srn}$ and $f_{trn}$

The design of the proposed $f_{srn}$ and $f_{trn}$ aims to tradeoff between network complexity and training or testing efficiency.

**Spatial recurrent learning.** We perform a two-step spatial recurrent learning. Particularly, the first step performs landmark detection to locate 7 major facial components that are robust to variations, *i.e.* four corners of left/right eyes, one nose tip, and two corners of the mouth. The second step performs landmark regression to refine the predicted locations

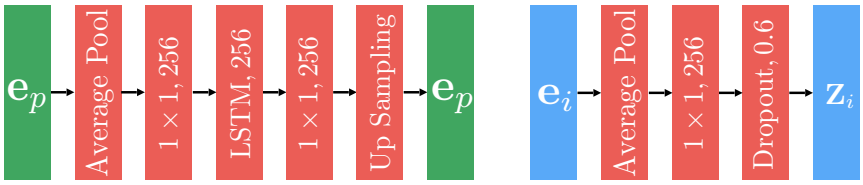



Figure 3.6: **Left:** the architecture of $f_{trn}$. We use average pooling to cut down the input dimension of LSTM and recover the dimension by upsampling. **Right:** the architecture of $f_{cls}$. We set $\mathbf{z}_i \in \mathbb{R}^{256}$ to achieve a compact identity representation.

of all 68 landmarks. This coarse-to-fine strategy guarantees efficient and robust spatial recurrent learning.

As mentioned in Section 3.3.2, the landmark detection task outputs a binary map of $C_d = 7$ channels, in which the values within a radius of 5 pixels around the ground truth are set to 1 and the values for the remaining background are set to 0. The landmark regression task outputs a heat map of $C_r = 68$ channels, in which the correct locations are represented by Gaussian with a standard deviation of 5 pixels. The two tasks share the weights of the entire encoder-decoder except for the last convolutional layer, which uses $1 \times 1$ convolutional layers to adapt to either the binary map or the heat map.

In either landmark detection or regression, the foreground pixels are much less than the background ones, which lead to highly unbalanced loss contributions. To solve this issue, we enlarge the foreground loss defined in Equation (3.8) and (3.11) by multiplying a constant weight (15 in most cases) to focus more on foreground pixels.

**Temporal recurrent learning.** We specify the configuration of $f_{trn}$ in Figure 3.6 (left). A Long Short Term Memory (LSTM) module [40, 87] is used to model the temporal variations of the identity code. There are 256 hidden neurons are used in LSTM. We empirically set the number of successive frames as $T = 10$ in Equation (3.11). The prediction loss is calculated at each time step. Directly feeding the non-identity code $\mathbf{e}_p$ into LSTM layers would lead to a slow training as it needs a large number of neurons for both the input and output. Instead, we apply average pooling to compress $\mathbf{e}_p$ to a $256d$ vector before inputting to the LSTM and recover it by unpooling with indices. Please check Figure 3.4 (left) for details.

### 3.4.3 The Design of $f_{cls}$

The design of $f_cls$ is shown in Figure 3.6 (right). The purpose of $f_{cls}$ is to apply additional identity constraint on $\mathbf{e}_i$, so the identity and non-identity codes can be decoupled more completely. Specially, $f_{cls}$ takes $\mathbf{e}_i$ as the input and output a $256d$ feature vector for the identity representation. Instead of using a very long feature vector in former face recognition networks [131], *e.g.* $4096d$, we use a compact one, *e.g.* $256d$, to reduce the computational cost for efficient training [118, 130]. We apply 0.6 dropout on the $256d$ vector to avoid overfitting. The vector is then followed by a fully-connected layer of $N$ neurons to output an one-hot vector for the identity prediction, where $N$ is the number of different subjects in training sequences. We use the cross-entropy loss defined in Equation (3.13) to train the identity task.

## 3.5 Experiments

We first introduce the datasets and settings. Then we carry out comprehensive module-wise study to validate the proposed method in various aspects. Finally, we compare our method with state-of-the-arts on both controlled and in-the-wild datasets.

### 3.5.1 Datasets and Settings

**Datasets.** We conduct our experiments on both image and video datasets. These datasets are widely used in face alignment and recognition. They present challenges in multiple aspects such as large pose, extensive expression, severe occlusion and dynamic illumination. Totally 7 datasets are used:

- Annotated Facial Landmarks in the Wild (AFLW) [59]
- Labeled Faces in the Wild (LFW) [66]
- Helen facial feature dataset (Helen) [65, 113]

Table 3.4: The image and video datasets used in training and evaluation. We split AFLW and 300-VW into two parts for training and evaluation, respectively. LFW, Helen, LFPW, TF, and FM are used for training only. Note that LFW, TF, FM and 300-VW have both landmark and identity annotations; while the others have only landmark annotations.

| Method | in-the-wild | img # | vid # | lmk # | sub # | train # | test # |
|---|---|---|---|---|---|---|---|
| AFLW [59] | yes | 21,080 | - | 21pt | - | 16,864 | 4,216 |
| LFW [66] | yes | 12,007 | - | 7pt | 5,371 | 12,007 | 0 |
| Helen [65] | yes | 2,330 | - | 194pt | - | 2,330 | 0 |
| LFPW [6] | yes | 1,035 | - | 68pt | - | 1,035 | 0 |
| TF [28] | no | 500 | 5 | 68pt | 1 | 0 | 500 |
| FM [102] | yes | 2,150 | 6 | 68pt | 6 | 0 | 2,150 |
| 300-VW [121] | yes | 114,000 | 114 | 68pt | 105 | 90,000 | 24,000 |

- Labeled Face Parts in the Wild (LFPW) [6, 113]
- Talking Face (TF) [28]
- Face Movies (FM) [102]
- 300 face Videos in the Wild (300-VW) [121]

We list configurations and setups of each dataset in Table 3.4. Different datasets have different landmark annotation protocol. For Helen, LFPW, TF, FM and 300-VW, we follow [113, 115] to obtain both 68- and 7-landmark annotation. For AFLW, we generat 7-landmark annotations using the original 21 landmarks. The landmark annotation of LFW is given by [66]. For identity labels, we manually label all videos in TF, FM, and 300-VW. It is easy since the identity is unique is a given video.

AFLW and 300-VW have the largest number of labeled images. They are also more challenging than others due to the extensive variations. Therefore, we use them for both training and evaluation. More specifically, 80% of the images in AFLW and 90 out of 114 videos in 300-VW are used for training, and the rest are used for evaluation. We sample videos to roughly cover the three different scenarios defined in [18], *i.e.* "Scenario 1", "Scenario 2" and "Scenario 3", corresponding to well-lit, mild unconstrained and completely unconstrained conditions.

We perform data augmentation by sampling ten variations from each image in the image training datasets. The sampling was achieved by random perturbation of scale (0.9 to 1.1), rotation ($\pm 15^{\circ}$), translation (7 pixels), as well as horizontal flip. To generate sequential training data, we randomly sample 100 clips from each training video, where each clip has 10 frames. It is worth mentioning that no augmentation is applied on video training data to preserve the temporal consistency in the successive frames.

**Compared methods.** We compared the proposed method with both regression based and deep learning based approaches that reported state-of-the-art performance in unconstrained conditions. Totally 8 methods are compared:

- Discriminative Response Map Fitting (DRMF) [3]
- Explicit Shape Regression (ESR) [15]
- Supervised Descent Method (SDM) [145]
- Incremental Face Alignment (IFA) [2]
- Coarse-to-Fine Shape Searching (CFSS) [156]
- Deep Convolutional Network Cascade (DCNC) [128]
- Coarse-to-fine Auto-encoder Network (CFAN) [153]
- Deep Multi-task Learning (TCDCN) [154]

For image-based evaluation, we follow [3] to provide a bounding box as the face detection output. For video-based evaluation, we follow [**?**] to utilize a tracking-by-detection protocol, where the face bounding box of the current frame is calculated according to the landmark of the previous frame.

**Training strategy.** Our approach is capable of end-to-end training. However, there are only 105 different subjects presented in 300-VW, which hardly provide sufficient supervision for the identity constraint. To make full use of all datasets, we conducted the training through

three steps. **First**, we pre-train the network without $f_{trn}$ and $f_{cls}$ using image-based datasets, *i.e.,* AFLW [59], Helen [65] and LFPW [6]. **Then**, $f_{cls}$ is engaged for identity constraint and fine-tuned together with other modules using image-based LFW [66]. **Finally**, $f_{trn}$ is triggered and the entire network is fine-tuned using video-based dataset, *i.e.* 300-VW [121].

**Experimental Settings.** In every frame, the initial response map $\mathbf{z}_0$ (Equation (3.2)) is generated using the landmark prediction of the previous frame. Parameter $\lambda$ and $\gamma$ (Equation (3.14)) are empirically set so the ratio of $\ell_{det} : \ell_{reg} : \ell_{cls}$ is roughly equal to $1 : 10 : 1$.

For training, we optimize the network parameters by using *stochastic gradient descent* (SGD) with 0.9 momentum. We use fixed learning rate started at 0.01 and manually decreased it to an order of magnitude according to the validation accuracy. $f_{enc}$ is initialized using pre-trained weights of VGG-16 [122] or ResNet-152 [39]. Other modules are initialized with Gaussian initialization [51]. The training clips in a mini-batch have no identity overlap to avoid oscillations of the identity loss. We perform temporal recurrent learning in both forward and backward direction to double the usage of the sequential corpus.

For testing, we split 300-VW so that the training and testing videos do not have identity overlap (16 videos share 7 identities) to avoid overfitting. We use the inter-ocular distance to normalize the *root mean square error* (RMSE) [113] for accuracy evalutaion. A prediction with larger than 10% mean error is reported as a failure [121].

### 3.5.2 Validation of Encoder-decoder Variants

In Section 3.4.1, we proposed two different designs of encoder-decoder: **(1)** VGGNet-based design with symmetrical encoder and decoder, which has been mainly investigated in our former conference paper [94]; and **(2)** ResNet-based design with asymmetrical encoder, *i.e.*, the encoder is much deeper than the decoder. In particular, skip connections are incorporated in bridging the encoder and decoder with hierarchical spatial information at different resolutions.

Table 3.5: Performance comparison of VGGNet-based and ResNet-based encoder-decoder Variants. Network configurations are described in Section 3.4.1. Row 1-2: image-based results on AFLW [59]; Row 3-4: video-based results on 300-VW [121].

| | Mean (%) | Std (%) | Time | Memory |
|---|---|---|---|---|
| VGGNet-based | 6.85 | 4.52 | 43.6$ms$ | 184$Mb$ |
| ResNet-based | 6.33 | 3.61 | 54.9$ms$ | 257$Mb$ |
| VGGNet-based | 5.16 | 2.57 | 42.5$ms$ | 184$Mb$ |
| ResNet-based | 4.75 | 2.10 | 56.2$ms$ | 257$Mb$ |

We compared the performance of two encoder-decoder variants on AFLW [59] and 300-VW [121]. The results are reported in Table 3.5. The results show that the ResNet-based design outperforms the VGGNet-based variant with a substantial margin in terms of fitting accuracy (mean error) and robustness (standard deviation). Much deeper layers, as well as the proposed skipping shortcuts, contribute a lot to the improvement. In addition, the ResNet-based encoder-decoder has very close computational cost to the VGGNet-based variant, *e.g.* the average fitting time per image/frame and the memory usage of a trained model, which should be attributed to the custom residual module design and the proposed asymmetrical encoder-decoder network.

### 3.5.3 Validation of Spatial Recurrent Learning

We validated the proposed spatial recurrent learning on the validation set of AFLW [59]. To better investigate the benefits of spatial recurrent learning, we partitioned the validation set into two image groups according to the absolute value of the yaw angle: **(1)** Common settings where yaw $\in$ [0°-30°); and **(2)** Challenging settings where yaw $\in$ (30°,90°]. The training sets are ensembles of AFLW [59], Helen [65] and LFPW [6] as shown in Table 3.4.

**Validation of detection-followed-by-regression.** To validate the proposed recurrent detection-followed-by-regression, we investigated four different network configurations:

- Single-step prediction using loss defined in Equation (3.5);
- Single-step prediction using loss defined in Equation (3.7);

Table 3.6: Comparison of single-step detection or regression with the proposed recurrent detection-followed-by-regression on AFLW [59]. The proposed method (Last Row) has the best performance especially in challenging settings.

| | Common (%) | | Challenging (%) | |
|---|---|---|---|---|
| | Error | Failure | Error | Failure |
| Single-step Detection | 6.05 | 4.62 | 8.14 | 12.4 |
| Single-step Regression | 5.92 | 4.75 | 7.87 | 14.5 |
| Recurrent Det.+Det. | 5.86 | 3.44 | 7.33 | 8.20 |
| Recurrent Det.+Reg. | 5.71 | 3.30 | 6.97 | 8.75 |

Table 3.7: Comparison of cascade and recurrent learning in the challenging settings of AFLW [59]. The latter improves accuracy with a half memory usage of the former.

| | Mean (%) | Std (%) | Memory |
|---|---|---|---|
| Cascade Det. & Reg. | 6.81 | 4.53 | $468Mb$ |
| Recurrent Det. & Reg. | 6.33 | 3.61 | $257Mb$ |

- Two-step recurrent detection-followed-by-detection;
- Two-step recurrent detection-followed-by-regression.

The mean fitting errors and failure rates are reported in Table 3.6. First, the results show that the two-step recurrent learning can instantly decrease the fitting error and failure rate, compared with either the single-step detection or regression. The improvement is more significant in challenging settings with large pose variations. Second, though landmark detection is more robust in challenging settings (low failure rate), it lacks the ability to predict precise locations (small fitting error) compared to landmark regression. This fact proves the effectiveness of the proposed recurrent detection-followed-by-regression.

**Validation of recurrent learning.** We also conducted comparisons between the proposed spatial recurrent learning and the cascade models that are widely used in former approaches [128, 153]. For a fair comparison, we implemented a two-step cascade variant to perform detection-followed-by-regression. Each network in the cascade has exactly the same architecture as the recurrent version. But there is no weight sharing among cascades. We fully trained the cascade networks using the same training set and validated the performance in challenging settings.

Table 3.8: Validation of temporal recurrent learning on 300-VW [113]. $f_{trn}$ helps to improve the tracking robustness (smaller std and lower failure rate), as well as the tracking accuracy (smaller mean error). The improvement is more significant in challenging settings of large pose and partial occlusion as demonstrated in Figure 3.7.

| | Common | | | Challenging | | |
|---|---|---|---|---|---|---|
| | Mean (%) | Std (%) | Fail (%) | Mean (%) | Std (%) | Failure (%) |
| w/o $f_{trn}$ | 4.52 | 2.24 | 3.48 | 6.27 | 5.33 | 13.3 |
| $f_{trn}$ | 4.21 | 1.85 | 1.71 | 5.64 | 3.28 | 5.40 |

The comparison is shown in Table 3.7. Unsurprisingly, the spatial recurrent learning can improve the fitting accuracy. The underlying reason is taht the recurrent network learns the step-by-step fitting strategy jointly, while the cascade networks learn each step independently. It can better handle the challenging settings where the initial guess is usually far away from the ground truth. Moreover, the recurrent network with shared weights can instantly reduce the memory usage to one-half of the cascaded model.

### 3.5.4 Validation of Temporal Recurrent Learning

We validate the proposed temporal recurrent learning on the validation set of 300-VW [121]. To better study the performance under different settings, we split the validation set into two groups: **(1)** 9 videos in common settings that roughly match "Scenario 1"; and **(2)** 15 videos in challenging settings that roughly match "Scenario 2" and "Scenario 3". The common, challenging and full sets were used for evaluation.

We implemented a variant of our approach that turns off the temporal recurrent learning $f_{trn}$. It was also pre-trained on the image training set and fine-tuned on the video training set. Since there was no temporal recurrent learning, we used frames instead of clips to conduct the fine-tuning which was performed for the same 50 epochs. We showed the result with and without temporal recurrent learning in Table 3.8.

For videos in common settings, the temporal recurrent learning achieves 6.8% and 17.4% improvement in terms of mean error and standard deviation respectively, while the failure rate is remarkably reduced by 50.8%. Temporal modeling produces better prediction

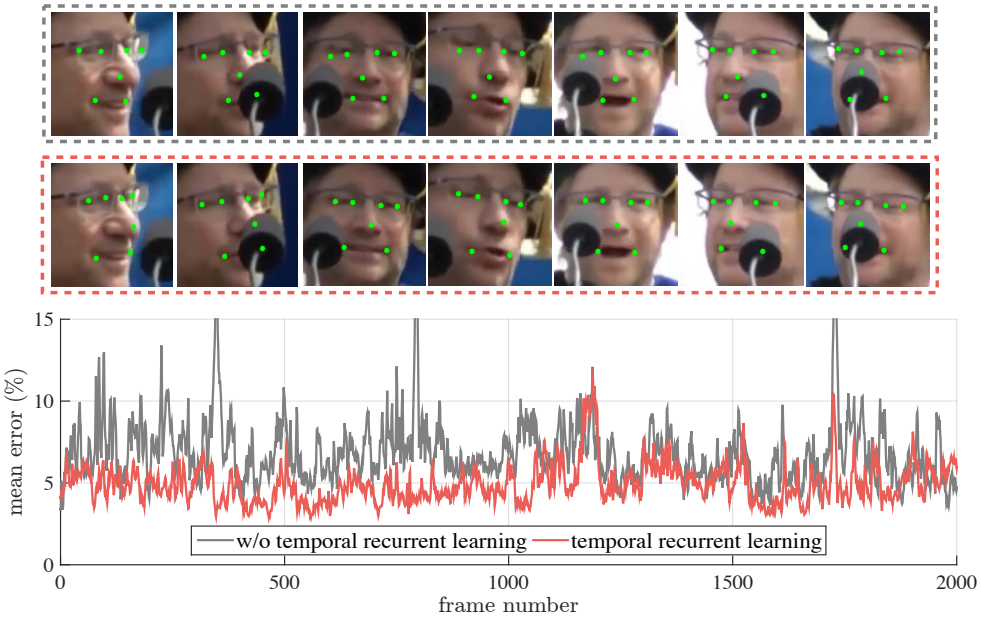



Figure 3.7: Examples of temporal recurrent learning on 300-VW [113]. The tracked subject undergoes intensive pose and expression variations as well as severe partial occlusions. $f_{trn}$ substantially improves the tracking robustness (less variance) and fitting accuracy (low error), especially for landmarks on the nose tip and mouth corners.

by taking consideration of history observations. It may implicitly learn to model the motion dynamics in the hidden units from the training clips.

For videos in challenging settings, the temporal recurrent learning won with even bigger margin. Without $f_{trn}$, it is hard to capture the drastic motion or changes in consecutive frames, which inevitably results in higher mean error, std and failure rate. Figure 3.7 shows an example where the subject exhibits intensive pose and expression variations as well as severe partial occlusions. The curve showed our recurrent model obviously reduced landmark errors, especially for landmarks on nose tip and mouth corners. The less oscillating error also suggests that $f_{trn}$ improves the prediction stability over frames.

### 3.5.5 Benefits of Supervised Identity Disentangling

The supervised identity disentangling is proposed to better decouple the temporal-variant and -invariant factors in the bottleneck of the encoder-decoder. This facilitates the temporal recurrent training, yielding better generalization and more accurate fittings at test time.

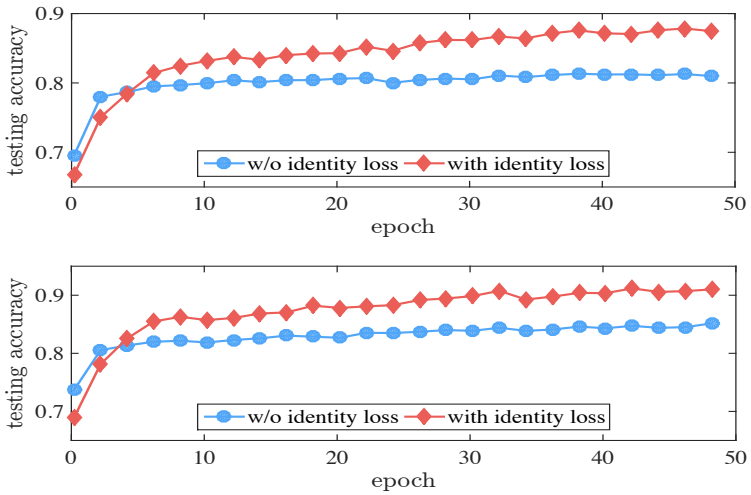



Figure 3.8: Fitting accuracy of different facial components with respect to the number of training epochs on 300-VW [121]. The proposed supervised identity disentangling helps to achieve a more complete factor decoupling in the bottleneck of the encoder-decoder, which yields better generalization capability and more accurate fitting results.

To study the effectiveness of the identity constraint, we removed $f_{cls}$ and follow the exact training steps. The testing accuracy comparison on the 300-VW [113] is shown in Figure 3.8. The accuracy was calculated as the ratio of pixels that were correctly classified in the corresponding channel(s) of the response map.

The validation results of different facial components show similar trends: **(1)** The network demonstrates better generalization capability by using additional identity cues, which results in a more efficient training. For instance, after only 10 training epochs, the validation accuracy for landmarks located at the left eye reaches 0.84 with identity loss compared to 0.8 without identity loss. **(2)** The supervised identity information can substantially boost the testing accuracy. There is an approximately 9% improvement by using the additional identity loss. It worth mentioning that, at the very beginning of the training (< 5 epochs), the network has inferior testing accuracy with supervised identity disentangling. It is because the suddenly added identity loss perturbs the backpropagation process. However, the testing accuracy with identity loss increases rapidly and outperforms the one without identity loss after only a few more training epochs.

Table 3.9: Mean error comparison with state-of-the-arts on video-based validation sets: TF, FM, and 300-VW [113]. The top performance in each dataset is highlighted. Our approach achieves the best fitting accuracy on both controlled and unconstrained datasets.

| 7 lmks | TF [28] | FM [102] | 300-VW [121] |
|---|---|---|---|
| DRMF [3] | 4.43 | 8.53 | 9.16 |
| ESR [15] | 3.81 | 7.58 | 7.83 |
| SDM [145] | 4.01 | 7.49 | 7.65 |
| IFA [2] | 3.45 | 6.39 | 6.78 |
| DCNC [128] | 3.67 | 6.16 | 6.43 |
| RED-Net (Ours) | **2.89** | **5.14** | **5.29** |

.

| 68 lmks | TF [28] | FM [102] | 300VW [121] |
|---|---|---|---|
| DRMF [3] | 3.49 | 6.74 | 7.09 |
| ESR [15] | 3.80 | 7.38 | 7.25 |
| SDM [145] | 3.31 | 6.47 | 6.64 |
| IFA [2] | 3.45 | 6.92 | 7.59 |
| DCNC [128] | 3.04 | 5.67 | 6.13 |
| RED-Net (Ours) | **2.77** | **4.93** | **5.15** |

### 3.5.6 General Comparison with the State of the art

We compared our framework with both traditional approaches and deep learning based approaches. The methods with hand-crafted features include: **(1)** DRMF [3], **(2)** ESR [15], **(3)** SDM [145], **(4)** IFA [2], and **(5)** PIEFA [102]. The deep learning based methods include: **(1)** DCNC [128], **(2)** CFAN [153], and **(3)** TCDCN [154]. All these methods were recently proposed and reported state-of-the-art performance. For fair comparison, we evaluated these methods in a tracking protocol: fitting result of current frame was used as the initial shape (DRMF, SDM and IFA) or the bounding box (ESR and PIEFA) in the next frame. The comparison was performed on both controlled, *e.g.* Talking Face (TF) [28], and in-the-wild datasets, *e.g.* Face Movie (FM) [102] and 300-VW [121].

We report the evaluation results for both 7 and 68 landmark setups in Table 3.9. Our approach achieves state-of-the-art performance under both settings. It outperforms others with a substantial margin on all datasets under both 7-landmark and 68-landmark protocols. The performance gain is more significant on the challenging datasets (FM and 300-VW) than controlled dataset (TF). Our alignment model runs fairly fast, it takes around 40ms to

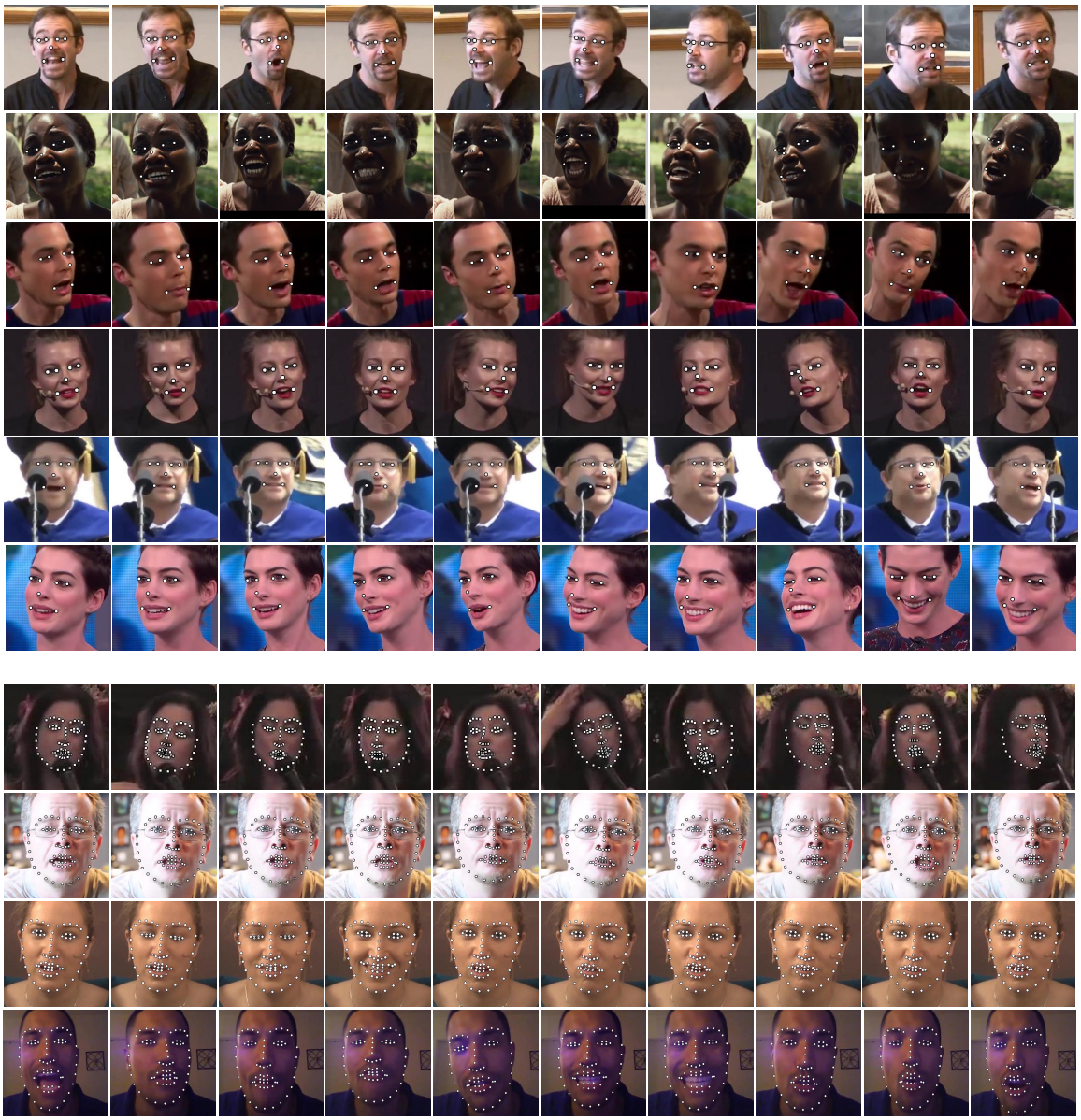


Figure 3.9: Examples of 7-landmark (**Row 1-6**) and 68-landmark (**Row 7-10**) fitting results on FM [**?**] and 300-VW [121]. The proposed approach achieves robust and accurate fittings when the tracked subjects suffer from large pose/expression changes (**Row 1, 3, 4, 6, 10**), illumination variations (**Row 2, 8**) and partial occlusions (**Row 5, 7**).

process an image using a Tesla K40 GPU accelerator. Please refer to Figure 3.9 for fitting results of our approach on FM [102] and 300-VW [121], which demonstrate the robust and accurate performance in wild conditions.

## 3.6 Discussion

In this study, we proposed a novel recurrent encoder-decoder network for real-time sequential face alignment. It utilizes spatial recurrent learning to train an end-to-end optimized coarse to fine landmark detection model. It decouples temporal-invariant and temporal-variant factors in the bottleneck of the network, and exploits recurrent learning at both spatial and temporal dimensions. Extensive experiments demonstrated the effectiveness of our framework and its superior performance. The proposed method provides a general framework that can be further applied to other localization-sensitive tasks, such as human pose estimation, object detection, scene classification, and others.

# Chapter 4

# Large-pose Face Recognition

Deep neural networks (DNNs) trained on large-scale datasets have recently achieved impressive improvements in face recognition. But a persistent challenge remains to develop methods capable of handling large pose variations that are relatively under-represented in training data. This paper presents a method for learning a feature representation that is invariant to pose, without requiring extensive pose coverage in training data. We first propose to generate non-frontal views from a single frontal face, in order to increase the diversity of training data while preserving accurate facial details that are critical for identity discrimination. Our next contribution is to seek a rich embedding that encodes identity features, as well as non-identity ones such as pose and landmark locations. Finally, we propose a new feature reconstruction metric learning to explicitly disentangle identity and pose, by demanding alignment between the feature reconstructions through various combinations of identity and pose features, which is obtained from two images of the same subject. Experiments on both controlled and in-the-wild face datasets, such as MultiPIE, 300WLP and the profile view database CFP, show that our method consistently outperforms the state-of-the-art, especially on images with large head pose variations. [1]

[1]Detail results and resource are referred to: https://sites.google.com/site/xipengcshomepage/iccv2017.

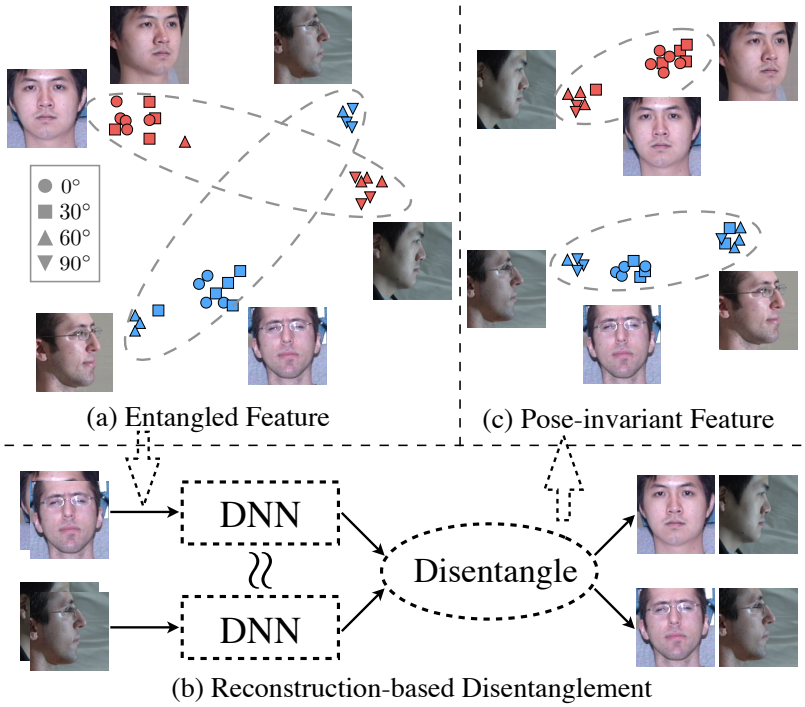



Figure 4.1: (a) Generic data-driven features for face recognition might confound images of the same identity under large poses with other identities, as shown two subjects (in different colors) from MultiPIE are mapped into the learned feature space of VGGFace [90]. (b) We propose a feature reconstruction metric learning to disentangle identity and pose information in the latent feature space. (c) The disentangled feature space encourages identity features of the same subject to be clustered together despite of the pose variation.

## 4.1 Introduction

The human visual system is commendable at recognition across variations in pose, for which two theoretical constructs are preferred. The first postulates invariance based on familiarity where separate view-specific visual representations or templates are learned [23, 104]. The second suggests that structural descriptions are learned from images that specify relations among viewpoint-invariant primitives [49]. Analogously, pose-invariance for face recognition in computer vision also falls into two such categories.

The use of powerful deep neural networks (DNNs) [61] has led to dramatic improvements in recognition accuracy. However, for objects such as faces where minute discrimination is required among a large number of identities, a straightforward implementation is still ineffective when faced with factors of variation such as pose changes [97]. Consider the feature space of the VGGFace [90] evaluated on MultiPIE [34] shown in Figure 4.1, where

examples from the same identity class that differ in pose are mapped to distant regions of the feature space.

An avenue to address this is by increasing the pose variation in training data. For instance, 4.4 million face images are used to train DeepFace [131] and 200 million labelled faces for FaceNet [118]. Another approach is to learn a mapping from different view-specific feature spaces to a common feature space through methods such as Canonical Correlation Analysis (CCA) [37]. Yet another direction is to ensemble over view-specific recognition modules that approximate the non-linear pose manifold with locally linear intervals [77, 55].

There are several drawbacks for the above class of approaches. First, conventional datasets including those sourced from the Internet have long-tailed pose distributions [76]. Thus, it is expensive to collect and label data that provides good coverage for all subjects. Second, there are applications for recognition across pose changes where the dataset does not contain such variations, for instance, recognizing an individual in surveillance videos against a dataset of photographs from identification documents. Third, the learned feature space does not provide insights since factors of variation such as identity and pose might still be entangled. Besides the above limitations, view-specific or multi-view methods require extra pose information or images under multiple poses at test time, which may not be available.

In contrast, we propose to learn a novel reconstruction based feature representation that is invariant to pose and does not require extensive pose coverage in training data. A challenge with pose-invariant representations is that discrimination power of the learned feature is harder to preserve, which we overcome with our holistic approach. First, inspired by [160], Section 4.3.1 proposes to enhance the diversity of training data with images under various poses (along with pose labels), at no additional labeling expense, by designing a face generation network. But unlike [160] which frontalizes non-frontal faces, we *generate rich pose variations* from frontal examples, which leads to advantages in better preservation of details and enrichment rather than normalization of within-subject variations. Next, to

achieve a rich feature embedding with good discrimination power, Section 4.3.2 presents a joint learning framework for identification, pose estimation and landmark localization. By jointly optimizing those three tasks, a *rich feature embedding* including both identity and non-identity information is learned. But this learned feature is still not guaranteed to be pose-invariant.

To achieve pose invariance, Section 4.3.3 proposes a feature reconstruction-based structure to explicitly *disentangle identity and non-identity* components of the learned feature. The network accepts a reference face image in frontal pose and another image under pose variation and extracts features corresponding to the rich embedding learned above. Then, it minimizes the error between two types of reconstructions in feature space. The first is *self-reconstruction*, where the reference sample's identity feature is combined with its non-identity feature and the second is *cross-reconstruction*, where the reference sample's non-identity feature is combined with the pose-variant sample's identity feature. This encourages the network to regularize the pose-variant sample's identity feature to be close to that of the reference sample. Thus, non-identity information is distilled away, leaving a disentangled identity representation for recognition at test.

Section 4.5 demonstrates the significant advantages of our approach on both controlled datasets and uncontrolled ones for recognition in-the-wild, especially on 90° cases. In particular, we achieve strong improvements over state-of-the-art methods on 300-WLP, MultiPIE, and CFP datasets. These improvements become increasingly significant as we consider performance under larger pose variations. We also present ablative studies to demonstrate the utility of each component in our framework, namely pose-variant face generation, rich feature embedding and disentanglement by feature reconstruction. To summarize, our key contributions are:

- To the best of our knowledge, we are the first to propose a novel reconstruction-based feature learning that disentangles factors of variation such as identity and pose.

- A comprehensively designed framework cascading rich feature embedding with the feature reconstruction, achieving pose-invariance in face recognition.
- A generation approach to enrich the diversity of training data, without incurring the expense of labeling large datasets spanning pose variations.
- Strong performance on both controlled and uncontrolled datasets, especially for large pose variations up to 90°.

## 4.2 Related Work

Face recognition is a popular topic involving large amount of research works. We only focus on the most relevant ones in this part. Literally we cover the three major components discussed in our method, namely face synthesization, deep face recognition, pose-invariant face recognition, and disentangle factors of variation.

**Face synthesization** Blanz and Vetter pioneered 3D morphable models (3DMM) for high quality face reconstruction [11] and recently, blend shape-based techniques have achieved real-time rates [14]. For face recognition, such techniques are introduced in DeepFace [131], where face frontalization is used for enhancing face recognition performance. As an independent application, specific frontalization techniques have also been proposed [38]. Another line of work pertains to 3D face reconstruction from photo collections [111, 70, 141] or a single image [76, 160, 134], where the latter have been successfully used for face normalization prior to recognition. While most of the methods apply the framework of aligning 3DMM with the 2D face landmarks [103] and conduct further refinement. In contrast, our use of 3DMM for face synthesis is geared towards enriching the diversity of training data.

**Deep face recognition** Several frameworks have recently been proposed that use DNNs to achieve impressive performances [90, 118, 127, 129, 131, 142, 148]. DeepFace [131] achieved verification rates comparable to human labeling on large test datasets, with further improvements from works such as DeepID [129]. Collecting face images from the Internet, FaceNet [118] trains on 200 million images from 8 million subjects. The very deep network can only be well stimulated by the huge volume of training data. The VGGNet [90] achieves competitive performance on the VGG face database and the recent feature learning approach of [142] reports top performance on the large scale MegaFace challenge [82].

We also use DNNs, but adopt the contrasting approach of learning pose-invariant features, since large-scale datasets with pose variations are expensive to collect, or do not exist in several applications such as surveillance.

**Pose-invariant face recognition** Early works use Canonical Correlation Analysis (CCA) to analyze the commonality among different pose subspaces [37, 86]. Further works consider generalization across multiple viewpoints [120] and multiview inter and intra discriminant analysis [54]. With the introduction of DNNs, prior works aim to transfer information from pose variant inputs to a frontalized appearance [135, 150], which is then used for face recognition [161]. The frontal appearance reconstruction usually relies on large amount of training data and the pairing across poses is too strict to be practical. Stacked progressive autoencoders (SPAE) [53] map face appearances under larger non-frontal poses to those under smaller ones in a continuous way by setting up hidden layers. The regression based mapping highly depends on training data and may lack generalization ability. Hierarchical-PEP [68] employs probabilistic elastic part (PEP) model to match facial parts from different yaw angles for unconstrained face recognition scenarios. The 3D face reconstruction method [160] synthesizes missing appearance due to large view points, which may introduce noise. Rather than compensating the missing information caused by severe pose variations at appearance level, we target learning a pose-invariant representation at feature level which preserves discrimination power through deep training.

**Disentangle factors of variation** Contractive discriminative analysis [110] learns disentangled representations in semi-supervised framework by regularizing representations to be orthogonal to each other. Disentangling Boltzmann machine [108] regularizes representations to be specific to each target task via manifold interaction. These methods involve non-trivial training procedure, and the pose variation is limited to half-profile views ($\pm 45^{\circ}$). Inverse graphics network [62] learns an interpretable representation by learning and decoding graphics codes, each of which encodes different factors of variation, but has been demonstrated only on the database generated from 3D CAD models. Multi-View Perceptron [163] disentangles pose and identity factors by cross-reconstruction of images synthesized from deterministic identity neurons and random hidden neurons. But it does not account for factors such as illumination or expression that are also needed for image-level reconstruction. In contrast, we use carefully designed embeddings as reconstruction targets instead of pixel-level images, which reduces the burden of reconstructing irrelevant factors of variation.

## 4.3 Method

We propose a novel pose-invariant feature learning method for large pose face recognition. Figure 4.2 provides an overview of our approach. *Pose-variant face generation* utilizes a 3D facial model to augment the training data with faces of novel viewpoints, besides generating ground-truth pose and facial landmark annotations. *Rich feature embedding* is then achieved by jointly learning the identity and non-identity features using multi-source supervision. Finally, *disentanglement by feature reconstruction* is performed to distill the identity feature from the non-identity one for better discrimination ability and pose-invariance.

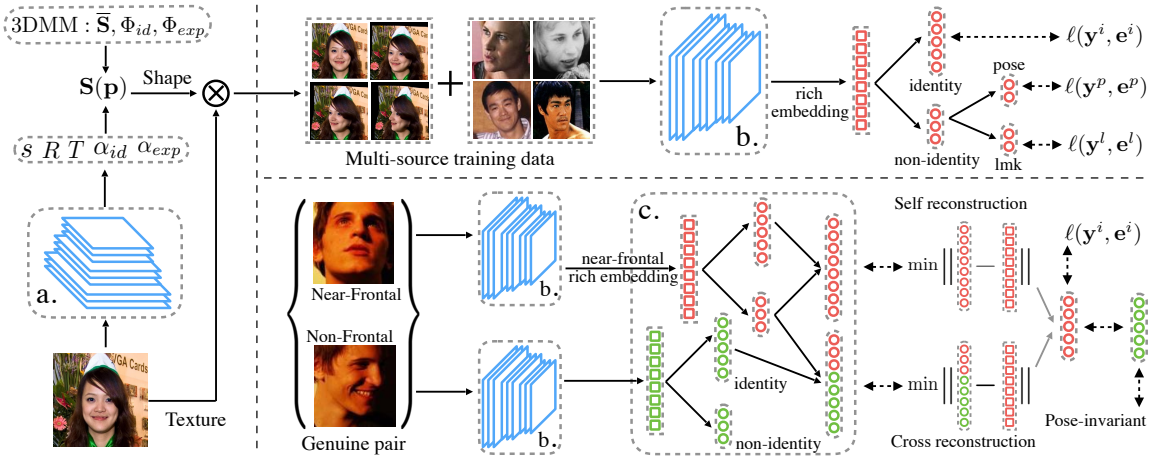



Figure 4.2: An overview of the proposed approach. (a) *Pose-variant face generation* utilizes a 3D facial model to synthesize new viewpoints from near-frontal faces. (b) *Rich feature embedding* is then achieved by jointly learning the identity and non-identity features using multi-source supervisions. (c) Finally, *Disentangling by reconstruction* is applied to distill the identity feature from the non-identity one for robust and pose-invariant representation.

### 4.3.1 Pose-variant Face Generation

The goal is to generate a series of pose-variant faces from a near-frontal image. This choice of generation approach is deliberate, since it can avoid hallucinating missing textures due to self-occlusion, which is a common problem with former approaches [38, 22] that rotate non-frontal faces to a normalized frontal view. More importantly, enriching instead of reducing intra-subject variations provides important training examples in learning pose-invariant features.

We reconstruct the 3D shape from a near-frontal face to generate new face images. Let $\chi$ be the set of frontal face images. A straightforward solution is to learn a nonlinear mapping $f(\cdot;\theta^s) : \chi \rightarrow \mathbb{R}^{3N}$ that maps an image $\mathbf{x} \in \chi$ to the $N$ coordinates of a 3D mesh. However, it is non-trivial to do so for a large number of vertices (15k), as required for a high-fidelity reconstruction.

Instead, we employ the 3D Morphable Model (3DMM) [11] to learn a nonlinear mapping $f(\cdot;\theta^s) : \chi \rightarrow \mathbb{R}^{235}$ that embeds $\mathbf{x}$ to a low-dimensional parameter space. The 3DMM parameters $\mathbf{p}$ control the rigid affine transformation and non-rigid deformation from a 3D

mean shape $\overline{\mathbf{S}}$ to the instance shape $\mathbf{S}$. Please refer to Figure 4.2 for an illustration:

$$\mathbf{S}(\mathbf{p}) = sR(\overline{\mathbf{S}} + \mathbf{\Phi_{id}}\alpha_{id} + \mathbf{\Phi_{exp}}\alpha_{exp}) + T, \tag{4.1}$$

where $\mathbf{p} = \{s, R, T, \alpha_{id}, \alpha_{exp}\}$ including scale $s$, rotation $R$, translation $T$, identity coefficient $\alpha_{id}$ and expression coefficient $\alpha_{exp}$. The eigenbases $\mathbf{\Phi_{id}}$ and $\mathbf{\Phi_{exp}}$ are learned offline using 3D face scans to model the identity [92] and expression [14] subspaces, respectively.

Once the 3D shape is recovered, we rotate the near-frontal face by evenly manipulating the yaw angle in the range of $[-90°, 90°]$. We follow [160] to use a z-buffer for collecting texture information and render the background for high-quality recovery. The rendered face is then projected to 2D to generate new face images from novel viewpoints.

### 4.3.2 Rich Feature Embedding

Most existing face recognition algorithms [76, 77, 118, 142] learn face representation using only identity supervision. An underlying assumption of their success is that deep networks can "implicitly" learn to suppress non-identity factors after seeing a large volume of images with identity labels [118, 131].

However, this assumption does not always hold when extensive non-identity variations exist. As shown in Figure 4.1 (a), the face representation and pose changes still present substantial correlations, even though this representation is learned throught a very deep neural network (VGGFace [90]) with large-scale training data (2.6M).

This indicates that using only identity supervision might not suffice to achieve an invariant representation. Motivated by this observation, we propose to utilize multi-source supervision to learn a rich feature embedding $\mathbf{e}^r$, which can be "explicitly" branched into an identity feature $\mathbf{e}^i$ and a non-identity feature $\mathbf{e}^n$, respectively. As we will show in the next section, the two features can collaborate to effectively achieve an invariant representation.

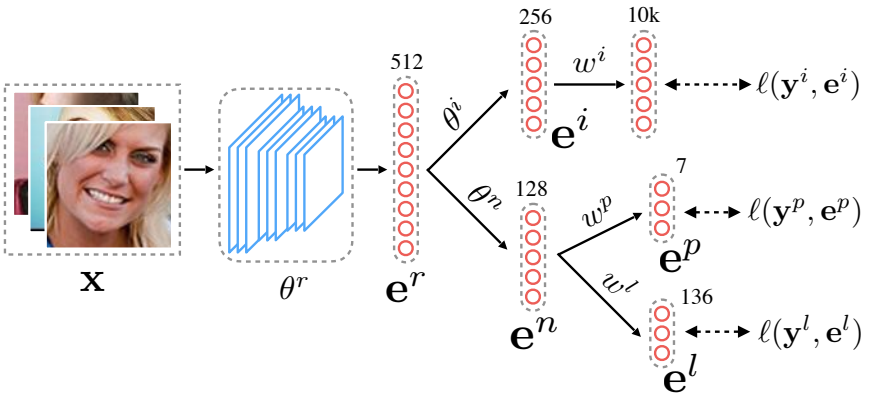



Figure 4.3: Pose-variant faces are used to fine-tune an off-the-shell recognition network $\theta^r$ to learn the rich feature embedding $\mathbf{e}^r$, which is explicitly branched into the identity feature $\mathbf{e}^i$ and the non-identity feature $\mathbf{e}^n$. Multi-source supervisions, such as identity, pose and landmark, are applied for joint optimization.

More specifically, as illustrated in Figure 4.3, $\mathbf{e}^n$ can be further branched as $\mathbf{e}^p$ and $\mathbf{e}^l$ to represent pose and landmark cues. For our multi-source training data that are not generated, we apply the CASIA-WebFace database [148] and provide the supervision from an off-the-shelf pose estimator. Therefore, we have:

$$\begin{aligned}
\mathbf{e}^i &= f(\mathbf{x}; \theta^r, \theta^i), \ \mathbf{e}^n = f(\mathbf{x}; \theta^r, \theta^n), \\
\mathbf{e}^p &= h(\mathbf{e}^n; w^p) = f(\mathbf{x}; \theta^r, \theta^n, w^p), \\
\mathbf{e}^l &= h(\mathbf{e}^n; w^l) = f(\mathbf{x}; \theta^r, \theta^n, w^l),
\end{aligned}$$

where mapping $f(\cdot; \theta/w) : \chi \rightarrow \mathbb{R}^d$ takes $\mathbf{x}$ and generates an embedding vector $f(\mathbf{x})$ and $\theta/w$ denotes the mapping parameters. Here, $\theta^r$ can be any off-the-shelf recognition network. $h(\cdot; \theta)$ is used to bridge two embedding vectors. We jointly learn all embeddings by optimizing:

$$\begin{aligned}
\underset{\theta^{r,i,n}, w^{i,p,l}}{\operatorname{argmin}} \sum_{image} & -\lambda^i \left[ \mathbf{y}^i \log softmax(w^{i^T} \mathbf{e}^i) \right] \\
& + \lambda^p \|\mathbf{y}^p - \mathbf{e}^p\|_2^2 + \lambda^l \|\mathbf{y}^l - \mathbf{e}^l\|_2^2,
\end{aligned} \tag{4.2}$$

where $\mathbf{y}^i$, $\mathbf{y}^p$ and $\mathbf{y}^l$ are identity, pose and landmark annotations and $\lambda^i$, $\lambda^p$ and $\lambda^l$ balance the weights between cross-entropy and $l_2$ loss.

By resorting to multi-source supervision, we can learn the rich feature embedding that "explicitly" encodes both identity and non-identity cues in $\mathbf{e}^i$ and $\mathbf{e}^n$, respectively. The remaining challenge is to distill $\mathbf{e}^i$ by disentangling from $\mathbf{e}^n$ to achieve identity-only representation.

### 4.3.3 Disentanglement by Feature Reconstruction

The identity and non-identity features above are jointly learned under different supervision. However, there is no guarantee that the identity factor has been fully disentangled from the non-identity one since there is no supervision applied on the decoupling process. This fact motivates us to propose a novel reconstruction-based framework for effective identity and non-identity disentanglement.

Recall that we have generated a series of pose-variant faces for each training subject in Section 4.3.1. These images share the same identity but have different viewpoints. We categorize these images into two groups according to their absolute yaw angles: near-frontal faces ($\leq 5°$) and non-frontal faces ($> 5°$). The two groups are used to sample image pairs that follow a specially designed configuration: a reference image randomly selected from the near-frontal group and a peer image randomly picked from the non-frontal group.

The next step is to obtain the identity and non-identity embeddings of two faces that have the same identity but different viewpoints. As shown in Figure 4.4, a pair of images $\{\mathbf{x}_k : k = 1, 2\}$ are fed into the network to output the identity and non-identity features:

$$\mathbf{e}_k^i = f(\mathbf{e}_k^r; \theta^i) = f(\mathbf{x}_k; \theta^r, \theta^i),$$
$$\mathbf{e}_k^n = f(\mathbf{e}_k^r; \theta^n) = f(\mathbf{x}_k; \theta^r, \theta^n).$$

Note that $\theta$ is not indexed by $k$ as the network shares weights to process the image pair.

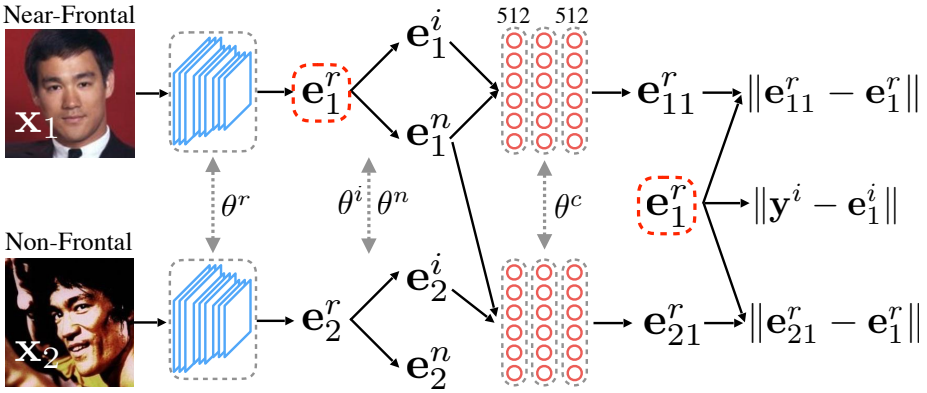



Figure 4.4: A genuine pair $\{\mathbf{x}_1, \mathbf{x}_2\}$ that share the same identity but different pose is fed into the recognition network $\theta^r$ to obtain the rich embedding $\mathbf{e}_1^r$ and $\mathbf{e}_2^r$. By regularizing the self and cross reconstruction, $\mathbf{e}_{11}^r$ and $\mathbf{e}_{21}^r$, the identity and non-identity features are eventually disentangled to make the non-frontal peer $\mathbf{e}_2^i$ to be similar to its near-frontal reference $\mathbf{e}_1^i$.

Our goal is to eventually push $\mathbf{e}_1^i$ and $\mathbf{e}_2^i$ close to each other to achieve a pose-invariant representation. A simple solution is to directly minimize the $l_2$ distance between the two features in the embedding subspace. However, this constraint only considers the identity branch, which might be entangled with non-identity, but completely ignores the non-identity factor, which provides strong supervision to purify the identity. Our experiments also indicate that a hard constraint would suffer from limited performance in large-pose conditions.

To address this issue, we propose to relax the constraint under a reconstruction-based framework. More specifically, we firstly introduce two reconstruction tasks:

$$\mathbf{e}_{11}^r = g(\mathbf{e}_1^i, \mathbf{e}_1^n; \theta^c), \ \mathbf{e}_{21}^r = g(\mathbf{e}_2^i, \mathbf{e}_1^n; \theta^c),$$

where $\mathbf{e}_{11}^r$ denotes the *self reconstruction* of the near-frontal rich embedding; while $\mathbf{e}_{21}^r$ denotes the *cross reconstruction* of the non-frontal rich embedding. Here, $g(\cdot, \cdot; \theta^c)$ is the reconstruction mapping with parameter $\theta^c$.

The identity and non-identity features can be rebalanced from the rich feature embedding by minimizing the self and cross reconstruction loss under the cross-entropy constraint:

$$\underset{\theta^i,\theta^n,\theta^c}{\operatorname{argmin}} \sum_{pair} -\gamma^i \left[\mathbf{y}_1^i \log softmax(w^{iT}\mathbf{e}_1^i)\right] + \gamma^s \|\mathbf{e}_{11}^r - \mathbf{e}_1^r\|_2^2 + \gamma^c \|\mathbf{e}_{21}^r - \mathbf{e}_1^r\|_2^2, \tag{4.3}$$

where $\gamma^i$, $\gamma^s$ and $\gamma^c$ weigh different constraints. Note that compared to (4.2), here we only finetune $\{\theta^i, \theta^n\}$ (as well as $\theta^c$) to rebalance the identity and non-identity features while keeping $\theta^r$ fixed, which is an important strategy to maintain the previously learned rich embedding.

In (4.3), we regularize both self and cross reconstructions to be close to the near-frontal rich embedding $\mathbf{e}_1^r$. Thus, portions of $\mathbf{e}_2^r$ to $\mathbf{e}_2^i$ and $\mathbf{e}_2^n$ are dynamically rebalanced to make the non-frontal peer $\mathbf{e}_2^i$ to be similar to the near-frontal reference $\mathbf{e}_1^i$. In other words, we encourage the network to learn a normalized feature representation across pose variations, thereby disentangling pose information from identity.

The proposed feature-level reconstruction is significantly different from former methods [118, 38] that attempt to frontalize faces at the image level. It can be directly optimized for pose invariance without suffering from artifacts that are common issues in face frontalization. Besides, our approach is an end-to-end solution that does not rely on extensive preprocessing usually required for image-level face normalization.

Our approach is also distinct from existing methods [77, 76] that synthesize pose-variant faces for data augmentation. Instead of feeding the network with a large number of augmented faces and letting it automatically learn pose-invariant or pose-specific features, we utilize the reconstruction loss to supervise the feature decoupling procedure. Moreover, factors of variation other than pose are also present in training, even though we only use pose as the driver for disentanglement. The cross-entropy loss in (4.3) plays an important role in preserving the discriminative power of identity features across various factors.

## 4.4 Implementation Details

In this section, we introduce the details of our designs for the three proposed functions: pose-variant face generation, rich feature embedding, and disentanglement by reconstruction.

**Pose-variant face generation.** A deep network is employed to predict 3DMM parameters of a near-frontal face as shown in Figure 4.2 (a). The network has a similar architecture as VGG16 [123]. We use pre-trained weights learned from ImageNet [61] to initialize the network instead of training from scratch. To further improve the performance, we make two important changes: (1) we use stride-2 convolution instead of max pooling to preserve the structure information when halving the feature maps; (2) the dimension of 3DMM parameters is changed to 66-$d$ (30 identity, 29 expression and 7 pose) instead of 235-$d$ used in [157]. To guarantee fast convergence and less overfitting, we apply principal component analysis to cut down the dimension of identity parameters from 199-$d$ to 30-$d$, which preserves 90% of the variance and significantly reduces the complexity. We evenly sample new viewpoints in every 5° from near-frontal faces to left/right profiles to cover the full range of pose variations. We apply Z-Buffer algorithm [160] to prevent the ambiguous pixel intensity due to same image plane position but different depths.

We use the same number of convolutional layers as VGG16 but replacing all max pooling layers with stride-2 convolutional operations. The fully connected (fc) layers are also different: we first use two fc layers, each of which has 1024 neurons, to connect with the convolutional modules; then, a fc layer of 30 neurons is used for identity parameters, a fc layer of 29 neurons is used for expression parameters, and a fc layer of 7 neurons is used for pose parameters. Different from [160] uses 199 parameters to represent the identity coefficients, we truncate the number of identity eigenvectors to 30 which preserves 90% of variations. This truncation leads to fast convergence and less overfitting. For texture, we only generate non-frontal faces from frontal ones, which significantly mitigate the hallucinating texture issue caused by self occlusion and guarantee high-fidelity reconstruction. We apply

the Z-Buffer algorithm used in [160] to prevent ambiguous pixel intensities due to same image plane position but different depths.

**Rich feature embedding.** The network is designed based on CASIA-net [148] with some improvements. As illustrated in Figure 4.3, we change the last fully connected layer to 512-$d$ for the rich feature embedding, which is then branched into 256-$d$ neurons for the identity feature and 128-$d$ neurons for the non-identity feature. To utilize multi-source supervision, the non-identity feature is further forked into 7-$d$ neurons for the pose embedding and 136-$d$ neurons for the landmark coordinates. Three different datasets are used to train the network: CASIA-WebFace, 300WLP and MultiPIE. We use Adam [58] stochastic optimizer with an initial learning rate of 0.0003, which drops by a factor of 0.25 every 5 epochs until convergence. Note that we train the network from scratch on purpose, since a pre-trained recognition model usually has limited ability to re-encode non-identity features.

During training, CASIA+MultiPIE or CASIA+300WLP are used. As shown in Figure 3 of the main submission, after the convolutional layers of CASIA-net, we use a 512-$d$ FC for the rich feature embedding, which is further branched into a 256-$d$ identity feature and a 128-$d$ non-identity feature. The 128-$d$ non-identity feature is further connected with a 136-d landmark prediction and a 7-$d$ pose prediction. Notice that in the face generation network, the number of pose parameters is 7 instead of 3 because we need to uniquely depict the projection matrix from the 3D model and the 2D face shape in image domain, which includes scale, pitch, yaw, roll, x translation, y translation, and z translations.

**Disentanglement by reconstruction.** Once $\{\theta^r, \theta^i, \theta^n\}$ are learned in the rich feature embedding, we freeze $\theta^r$ and finetune $\theta^i$ and $\theta^n$ to rebalance the identity and non-identity features as explained in Figure 4.4 and Equation (4.3). The network takes the concatenation (384-$d$) of $\mathbf{e}^i$ and $\mathbf{e}^n$ and outputs the reconstructed embedding (512-$d$). The mapping is achieved by rolling though two fully connected layers and each of them has 512-$d$ neurons. We have tried different network configurations but get similar performance. The initial

learning rate is set to 0.0001 and the hyper-parameters $\gamma^{i,s,c}$ are determined via 5-fold cross-validation. We also find that it is import to do early stopping for effective reconstruction-based regularization. In Equation (4.2) and (4.3), we use the cross-entropy loss to preserve the discriminative power of the identity feature. Other identity regularizations, *e.g.* triplet loss [118], can be easily applied in a plug-and-play manner.

To disentangle the identity and pose factors, we concatenate the identity and non-identity features and roll though two 512-$d$ fully connected layers to output a reconstructed rich embedding depicted by 512 neurons. Both self and cross reconstruction loss are designed to eventually push the two identity features close to each other. At the same time, a cross-entropy loss is applied on the near-frontal identity feature to maintain the discriminative power of the learned representation. The disentanglement of the identity and pose is finally achieved by the proposed feature reconstruction based metric learning.

The training of recognition network involves two steps. First, the multi-source multi-task training is applied with reconstruction objectives in Equation (4.3). We train from scratch using Adam [58] stochastic optimization with learning rate 0.0003 until convergence. Second, based on the same network structure, freezing all the convolution layers and the first fully connected layer, we specifically train the Siamese reconstruction network to optimize the identity related and unrelated network parameters, as shown in Figure 4.4. Identity loss is also applied for the reconstruction task. We find that it is important to do early stopping for the second step training. Relevant hyper-parameters, such as $\gamma$ or maximum number of training iterations, are determined via 5-fold cross-validation.

## 4.5 Experiments

We evaluate our feature learning method on three main pose-variant databases, Multi-PIE [34], 300WLP [157] and CFP [119]. We also compare with two top general face recognition frameworks, VGGFace [90] and N-pair loss face recognition [124], and three

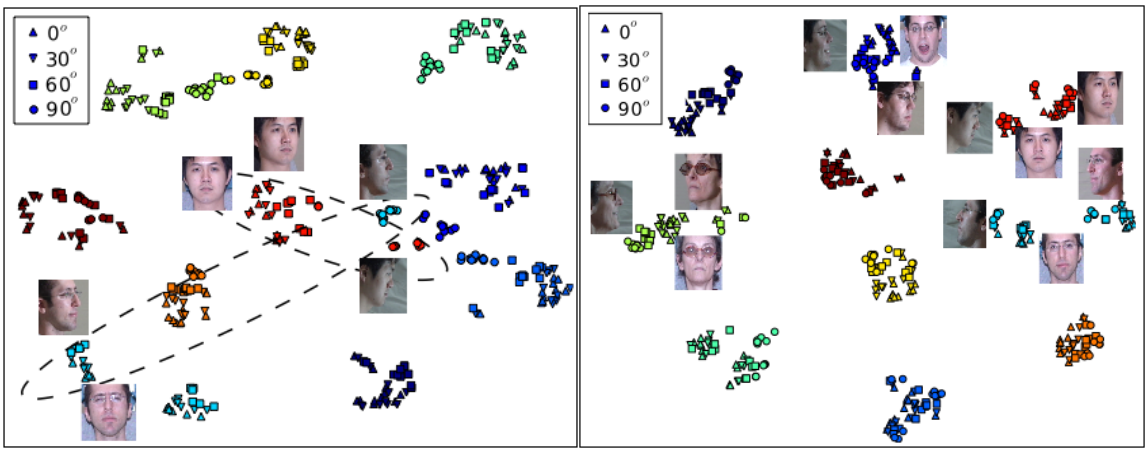
| Method | 15° | 30° | 45° | 60° | 75° | 90° | Avg |
|---|---|---|---|---|---|---|---|
| VGGFace [90] | 0.972 | 0.961 | 0.926 | 0.847 | 0.628 | 0.342 | 0.780 |
| N-pair [124] | 0.990 | 0.983 | **0.971** | **0.944** | 0.811 | 0.468 | 0.861 |
| MvDA [54]† | **1.000** | 0.979 | 0.909 | 0.855 | 0.718 | 0.564 | 0.837 |
| GMA [120]† | **1.000** | **1.000** | 0.904 | 0.852 | 0.725 | 0.550 | 0.838 |
| MvDN [55]† | **1.000** | 0.991 | 0.921 | 0.897 | 0.810 | 0.706 | 0.887 |
| Ours (P1) | 0.972 | 0.966 | 0.956 | 0.927 | **0.857** | **0.749** | **0.905** |
| Ours (P2) | 1.000 | 1.000 | 0.995 | 0.982 | 0.931 | 0.817 | 0.954 |

Table 4.1: Rank-1 recognition accuracy on MultiPIE at different yaw angles. The numbers in the entry with † are obtained from [55]. We evaluate our method using gallery set composed of 2 frontal face images per subject (P1) as well as entire frontal face images (P2).

state-of-the-art pose-invariant face recognition methods, namely, MvDA [54], GMA [120] and MvDN [55]. Further, we present an ablation study to emphasize the significance of each module that we carefully designed and a cross-database validation demonstrates the good generalization ability of our method.

### 4.5.1 Evaluation on MultiPIE

MultiPIE [34] is composed of 754,200 images of 337 subjects with different factors of variation such as pose, illumination, and expression. There are 15 different head poses set up, where we only use images of 13 head poses with yaw angle changes from $-90°$ to $90°$, with $15°$ difference every consecutive pose bin in this experiment. We split the data into train and test by subjects, of which the first 229 subjects are used for training and the remaining 108 are used for testing. This is similar to the experimental setting in [55], but we use entire data including both illumination and expression variations for training while excluding only those images taken with top-down views. Rank-1 recognition accuracy of non-frontal face images is reported. We take $\pm 15°$ to $\pm 90°$ as query and the frontal faces ($0°$) as gallery, while restricting illumination condition to be neutral. To be consistent with the experimental setting of [55], we form a gallery set by randomly selecting 2 frontal face images per subject, of which there are a total of 216 images. We evaluate the recognition accuracy for all query examples, of which there are 619 images per pose. The procedure is done with 10 random selections of gallery sets and mean accuracy is reported.




(a) VGGFace Feature Space (b) Disentangled Feature Space

Figure 4.5: t-SNE visualization of VGGFace [90] feature space (left) and the proposed reconstruction-based disentangling feature space (right), with 10 subjects from MultiPIE [34]. The same marker color indicates the same subject. Different marker shapes indicate different head poses. Our approach shows better results in disentangling pose factors from identity representations.

Evaluation is shown in Table 4.1. The recognition accuracy at every 15° interval of yaw angle is reported while averaging its symmetric counterpart with respect to the 0-yaw axis. For the two general face recognition algorithms, VGGFace [90] and N-pair loss [124], we clearly observe more than 30% accuracy drop when the head pose approaches 90° from 75°. Our method significantly reduces the drop by more than 20%. The general methods are trained with very large databases leveraging across different poses, but our method has the additional benefit of explicitly aiming for a pose invariant feature representation.

The pose-invariant methods, GMA, MvDA, and MvDN demonstrate good performance within 30° yaw angles, but again the performance starts to degrade significantly when yaw angle is larger than 30°. When comparing the accuracy on extreme poses from 45° to 90°, our method achieves accuracy 3 $\sim$ 4% better than the best reported. Besides the improved performance, our method has an advantage over MvDN, since it does not require pose information at test time. On the other hand, MvDN is composed of multiple sub-networks, each of which is specific to a certain pose variation and therefore requires additional information on head pose for recognition.

Figure 4.5 shows t-SNE visualization [137] of VGGFace [90] feature space and the proposed reconstruction-based disentangling feature space of MultiPIE [34]. For visual-

| Method | 15° | 30° | 45° | 60° | 75° | 90° | Avg |
|---|---|---|---|---|---|---|---|
| VGGFace [90] | 0.994 | 0.998 | **0.996** | 0.956 | 0.804 | 0.486 | 0.838 |
| N-Pair [124] | **1.000** | 0.996 | 0.993 | 0.962 | 0.845 | 0.542 | 0.859 |
| Ours | **1.000** | **0.999** | 0.995 | **0.994** | **0.978** | **0.940** | **0.980** |

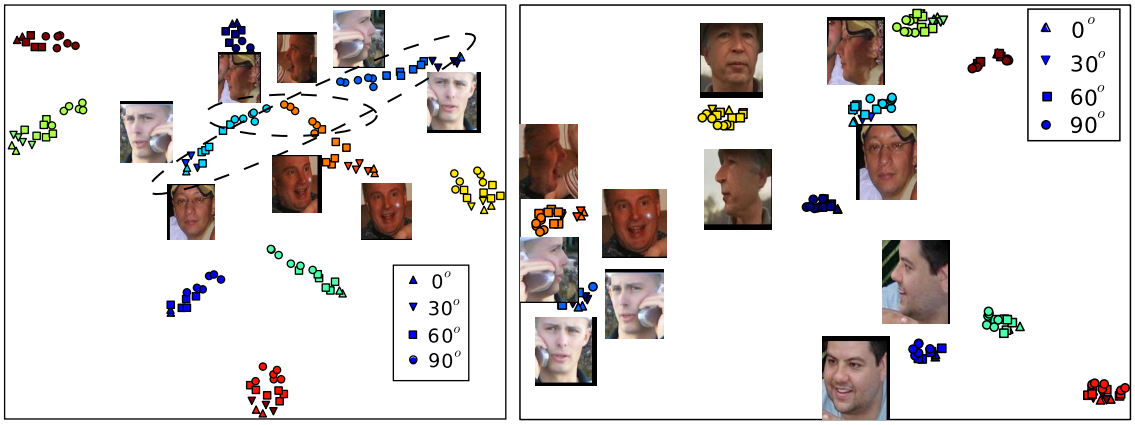
Table 4.2: Recognition performance on 300WLP, the proposed method with two general state-of-the-art face recognition frameworks, i.e. VGG Face Recognition Network (VGGFace) and N-pair loss face recognition (N-pair).

ization clarity, we only visualize 10 randomly selected subjects from the test set with 0°, 30°, 60°, and 90° yaw angles. Figure 4.5 (a) shows that samples from VGGFace feature embedding have large overlap among different subjects. In contrast, Figure 4.5 (b) shows that our approach can tightly cluster samples of the same subject together which leads to little overlap of different subjects, since identity features have been disentangled from pose in this case.

### 4.5.2 Evaluation on 300WLP

We further evaluate on a face-in-the-wild database, 300 Wild Large Pose [157] (300WLP). It is generated from 300W [114] face database by 3DDFA [157], in which it establishes a 3D morphable model and reconstruct the face appearance with varying head poses. It consists of overall 122,430 images from 3,837 subjects. Compared to MultiPIE, the overall volume is smaller, but the number of subjects is significantly larger. For each subject, images are with uniformly distributed continuously varying head poses in contrast to MultiPIE's strictly controlled 15° head pose intervals. The lighting conditions as well as the background are almost identical. Thus, it is an ideal dataset to evaluate algorithms for pose variation.

We randomly split 500 subjects of 8014 images as testing data and the rest 3337 subjects of 106,402 images as the training data. Among the testing data, two 0° head pose images per subject form the gallery and the rest 7014 images serves as the probe. Table 4.2 shows the comparison with two state-of-the-art general face recognition methods, i.e. VGGFace [90] and N-pair loss face recognition [124]. To the best of our knowledge, we are the first to apply




(a) VGGFace Feature Space (b) Disentangled Feature Space

Figure 4.6: t-SNE visualization of VGGFace [90] feature space (left) and the proposed reconstruction-based disentangling feature space (right), with 10 subjects from 300WLP [157]. The same marker color indicates the same subject. Different marker shapes indicate different head poses. Our approach shows better results in disentangling pose factors from identity representations.

our pose-invariant face recognition framework on this dataset. Thus, we only compare our method with the two general face recognition frameworks.

Since head poses in 300WLP continuously vary, we group the test samples into 6 pose intervals, $(0, 15°)$, $(15°, 30°)$, $(30°, 45°)$, $(45°, 60°)$, $(60°, 75°)$ and $(75°, 90°)$. For short annotation, we mark each interval with the end point, e.g., $30°$ denotes the pose interval $(15°, 30°)$. From Table 4.2, our method achieves consistently better accuracy especially when pose angle approaches $90°$, which is clearly contributed by our feature reconstruction based disentanglement.

We also conduct experiments on the Labeled Faces-in-the-Wild (LFW) dataset [45]. We note that the dataset is particularly biased towards frontal poses. Among 6,000 verification image pairs, only 358 pairs are with large pose variations, of which at lease one image is with pose greater than $30°$. The numbers are only to illustrate the potential of our framework for recognition in-the-wild under large poses. On the above subset of non-frontal images, we achieve 96.37% verification accuracy using the MSMT baseline model, which improves to 96.93% using our reconstruction-based disentanglement. This demonstrates that our method also applies to unconstrained scenarios. The VGGFace model achieves 97.49%

| Method | Frontal-Frontal | Frontal-Profile |
|---|---|---|
| Sengupta et al. [119] | 96.40 | 84.91 |
| Sankarana et al. [116] | 96.93 | 89.17 |
| Chen et al. [16] | **98.67** | 91.97 |
| DR-GAN [135] | 97.84 | 93.41 |
| Human | 96.24 | 94.57 |
| Ours | **98.67** | **93.76** |

Table 4.3: Verification accuracy comparison on CFP dataset.

accuracy on the same subset. However, it is trained on a dataset much larger than our combination of CASIA and MultiPIE.

Figure 4.6 shows t-SNE visualization [137] of VGGFace [90] feature space and the proposed reconstruction-based disentangling feature space, with 10 subjects from 300WLP [157]. Similar to the results of MultiPIE [34], the VGGFace feature embedding space shows entanglement between identity and the pose, i.e., the man with the phone in 45° view is overlapped with the frontal view image of other persons. In contrast, feature embeddings of our method are largely separated from one to another, while embeddings of the same subject are clustered together even there are extensive pose variations.

### 4.5.3 Evaluation on CFP

The Celebrities in Frontal-Profile (CFP) database [119] focuses on extreme head pose face verification. It consists of 500 subjects, with 10 frontal images and 4 profile images for each, in a wild setting. The evaluation is conducted by averaging the performance of 10 randomly selected splits with 350 identical and 350 non-identical pairs. Our MSMT+SR finetuned on MultiPIE with N-pair loss is the model evaluated in this experiment. The reported human performance is 94.57% accuracy on the frontal-profile protocol and 96.24% on the frontal-frontal protocol, which shows the challenge of recognizing profile views.

Results in Table 4.3 suggest that our method achieves consistently better performance compared to state-of-the-art. We reach the same Frontal-Frontal accuracy as Chen et al. [16]

while being significantly better on Frontal-Profile by 1.8%. We are slightly better than DR-GAN [135] on extreme pose evaluation and 0.8% better on frontal cases. DR-GAN is a recent generative method that seeks the identity preservation at the image level, which is not a direct optimization on the features. Our feature reconstruction method preserves identity even when presented with profile view faces. In particular, as opposed to prior methods, ours is the only one that obtains very high accuracy on both the evaluation protocols.

### 4.5.4 Control Experiments

We extensively evaluate recognition performance on various baselines to study the effectiveness of each module in our proposed framework. Specifically, we evaluate and compare the following models:

- SS: trained on a single source (e.g., CASIA-WebFace) using softmax loss only.
- SS-FT: fine-tuned on a target dataset (MultiPIE or 300WLP) using softmax loss only.
- MSMT: trained on multiple data sources (CASIA + MultiPIE or 300WLP) using softmax loss for identity and $L_2$ loss for pose.
- MSMT+L2: fine-tuned on MSMT models using softmax loss and Euclidean loss.
- MSMT+SR: fine-tuned on MSMT models using softmax loss and Siamese reconstruction loss.
- MSMT$^{\dagger}$: trained on the same multiple data sources as MSMT, using N-pair [124] metric loss for identity and $L_2$ loss for pose.
- MSMT$^{\dagger}$+SR: finetuned on MSMT$^{\dagger}$ models with N-pair loss and reconstruction loss.

The SS model serves as the weakest baseline. From Table 4.4 we observe that simultaneously training the network on multiple sources of CASIA and MultiPIE (or 300WLP) using multi-task objective (i.e., identification loss, pose or landmark estimation loss) is more

| Method | MultiPIE | | | | | | |
|---|---|---|---|---|---|---|---|
| | 15° | 30° | 45° | 60° | 75° | 90° | Avg |
| SS | 0.908 | 0.899 | 0.864 | 0.778 | 0.487 | 0.207 | 0.690 |
| SS-FT | 0.941 | 0.936 | 0.919 | 0.883 | 0.799 | 0.681 | 0.860 |
| MSMT | 0.965 | 0.955 | 0.945 | 0.914 | 0.827 | 0.689 | 0.882 |
| MSMT+L2 | 0.972 | 0.965 | 0.954 | 0.923 | 0.849 | 0.739 | 0.900 |
| MSMT+SR | 0.972 | 0.966 | 0.956 | 0.927 | 0.857 | **0.749** | 0.905 |
| MSMT$^{\dagger}$ | 0.993 | 0.989 | 0.982 | 0.959 | 0.903 | 0.734 | 0.927 |
| MSMT$^{\dagger}$+SR | **0.994** | **0.990** | **0.982** | **0.960** | **0.906** | 0.745 | **0.929** |
| Method | 300WLP | | | | | | |
| | 15° | 30° | 45° | 60° | 75° | 90° | Avg |
| SS | 0.945 | 0.934 | 0.884 | 0.753 | 0.567 | 0.330 | 0.679 |
| SS-FT | **1.000** | 0.999 | 0.992 | 0.973 | 0.934 | 0.839 | 0.944 |
| MSMT | **1.000** | 0.993 | 0.993 | 0.986 | 0.968 | 0.922 | 0.971 |
| MSMT+L2 | **1.000** | 0.997 | 0.996 | 0.991 | 0.973 | 0.933 | 0.977 |
| MSMT+SR | **1.000** | **0.999** | 0.995 | 0.994 | 0.978 | 0.940 | 0.980 |
| MSMT$^{\dagger}$ | **1.000** | 0.998 | 0.997 | 0.994 | 0.981 | 0.922 | 0.977 |
| MSMT$^{\dagger}$+SR | **1.000** | 0.998 | **0.999** | **0.997** | **0.988** | **0.953** | **0.986** |

Table 4.4: Recognition performance of several baseline models, i.e., single source trained model on CASIA database (SS), single source model fine-tuned on the target database (SS-FT), multi-source multi-task models (MSMT), MSMT with direct identity feature $\ell_2$ distance regularization (MSMT+L2), the proposed MSMT with Siamese reconstruction regularization models (MSMT+SR), MSMT with N-pair loss instead of cross entropy loss (MSMT$^{\dagger}$) and MSMT$^{\dagger}$ with SR, evaluated on MultiPIE (P1) and 300WLP.

effective than single-source training followed by fine-tuning. We believe that our MSMT learning can be viewed as a form of curriculum learning [8] since multiple objectives introduced by multi-source and multi-task learning are at different levels of difficulty (e.g., pose and landmark estimation or identification on MultiPIE and 300WLP are relatively easier than identification on CASIA-WebFace) and easier objectives allow to train faster and converge to better solution. As an alternative to reconstruction regularization, one may consider reducing the distance between the identity-related features of the same subject under different pose directly (MSMT+L2). Learning to reduce the distance improves the performance over the MSMT model, but is not as effective as our proposed reconstruction regularization method, especially on face images with large pose variations.

Further, we observe that employing the N-pair loss [124] within our framework also boosts performance, which is shown by the improvements from MSMT to MSMT$^{\dagger}$ and MSMT+SR to MSMT$^{\dagger}$+SR. We note that the MSMT$^{\dagger}$ baseline is not explored in prior

| Method | MultiPIE | | | | | | |
|---|---|---|---|---|---|---|---|
| | 15° | 30° | 45° | 60° | 75° | 90° | Avg |
| SS | 0.908 | 0.899 | 0.864 | 0.778 | 0.487 | 0.207 | 0.690 |
| SS-FT | 0.941 | 0.936 | 0.919 | 0.883 | 0.799 | 0.681 | 0.860 |
| MSMT | 0.965 | 0.955 | 0.945 | 0.914 | 0.827 | 0.689 | 0.882 |
| MSMT+L2 | 0.972 | 0.965 | 0.954 | 0.923 | 0.849 | 0.739 | 0.900 |
| MSMT+SR (ours) | **0.972** | **0.966** | **0.955** | **0.927** | **0.857** | **0.749** | **0.905** |
| Method | MultiPIE | | | | | | |
| | 15° | 30° | 45° | 60° | 75° | 90° | Avg |
| SS | 1.00 | 0.998 | 0.985 | 0.892 | 0.563 | 0.250 | 0.781 |
| SS-FT | 0.999 | 0.993 | 0.981 | 0.951 | 0.874 | 0.753 | 0.925 |
| MSMT | 1.00 | 1.00 | 0.993 | 0.982 | 0.908 | 0.753 | 0.939 |
| MSMT+L2 | 1.00 | 999 | 0.990 | 0.978 | 0.911 | 0.800 | 0.946 |
| MSMT+SR (ours) | **1.00** | 0.999 | **0.995** | **0.982** | **0.931** | **0.817** | **0.954** |

Table 4.5: Rank-1 recognition accuracy comparisons under P1 (top) and P2 (bottom) testing protocol on MultiPIE [34] dataset.

works on pose-invariant face recognition. It provides a different way to achieve similar goals as the proposed reconstruction method. Indeed, a collateral observation through the relative performances of MSMT and MSMT$^{\dagger}$ is that the softmax loss is not good at disentangling pose from identity, while metric learning excels at it. Indeed, our feature reconstruction metric might be seen as achieving a similar goal, thus, improvements over MSMT$^{\dagger}$ are marginal, while those over MSMT are large.

In Table 4.5, we report the standard deviation of our method as a more complete comparison. From the results, the standard deviation of our method is also very small, which suggests that the performance is consistent across all the trials. We also compare the cross database evaluation on both mean accuracy and standard deviation in Table 4.6. We show the models trained on 300WLP and tested on MultiPIE with both P1 and P2 protocol. Please note that with P2 protocol, our method still achieves better performance on MultiPIE than MvDN [55] with 0.7% gap. Further, across different testing protocols, the proposed method consistently outperforms the baseline method MSMT, which clearly shows the effectiveness of our proposed Siamese reconstruction based regularization for pose-invariant feature representation.

| Method | | MultiPIE | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | 15° | 30° | 45° | 60° | 75° | 90° | Avg |
| MultiPIE | MSMT | 0.965 | 0.955 | 0.945 | 0.914 | 0.827 | 0.689 | 0.882 |
| | Ours | 0.972 | 0.966 | 0.956 | 0.927 | 0.857 | 0.749 | 0.905 |
| 300WLP | MSMT | 0.941 | 0.927 | 0.898 | 0.837 | 0.695 | 0.432 | 0.788 |
| | Ours | 0.945 | 0.933 | 0.910 | 0.862 | 0.736 | 0.459 | 0.808 |
| Method | | 300WLP | | | | | | |
| | | 15° | 30° | 45° | 60° | 75° | 90° | Avg |
| MultiPIE | MSMT | 1.000 | 0.996 | 0.988 | 0.953 | 0.889 | 0.720 | 0.904 |
| | Ours | 0.994 | 0.995 | 0.992 | 0.958 | 0.901 | 0.733 | 0.910 |
| 300WLP | MSMT | 1.000 | 0.993 | 0.993 | 0.986 | 0.968 | 0.922 | 0.971 |
| | Ours | 1.000 | 0.999 | 0.995 | 0.994 | 0.978 | 0.940 | 0.980 |

Table 4.6: Cross database evaluation on MultiPIE and 300WLP. The top two rows show the model of MSMT and our method trained on CASIA and MultiPIE, while tested on both MultiPIE and 300WLP. The bottom two rows show the model of MSMT and our method trained on CASIA and 300WLP, while tested on both MultiPIE and 300WLP.

The P2 testing protocol utilizes all the 0° images as the gallery. The performance is expected to be better than that reported on P1 protocol in the main submission since more images are used for reference. There is no standard deviation in this experiment as the gallery is fixed by using all the frontal images. The results are shown in Table 4.5, which confirms the conclusion that the proposed feature reconstruction based regularization is effective in obtaining pose-invariant and highly discriminative feature representations for face recognition.

### 4.5.5 Cross Database Evaluation

We evaluate our models, which are trained on CASIA with MultiPIE or 300WLP, on the cross test set 300WLP or MultiPIE, respectively. Results are shown in Table 4.6 to validate the generalization ability. There are obvious accuracy drops on both databases, for instance, a 7% drop on 300WLP and 10% drop on MultiPIE. However, such performance drops are expected since there exists a large gap in the distribution between MultiPIE and 300WLP.

Interestingly, we observe significant improvements when compared to VGGFace. These are fair comparisons since neither networks is trained on the training set of the target dataset. When evaluated on MultiPIE, our MSMT model trained on 300WLP and CASIA


Figure 4.7: The gallery and probe samples adopted in the testing from MultiPIE [34] and 300WLP [160]. (a) The gallery samples of MultiPIE. (b) The probe samples of MultiPIE. (c) The gallery samples of 300WLP. (d) The probe samples of 300WLP.

database improves 0.8% over VGGFace and the model with reconstruction regularization demonstrates stronger performance, showing 2.8% improvement over VGGFace. Similarly, we observe 6.6% and 7.2% improvements for MultiPIE and CASIA trained MSMT models and our proposed MSMT+SR, respectively, over VGGFace when evaluated on the 300WLP test set. This partially confirms that our performance is not an artifact of overfitting to a specific dataset, but is generalizable across different datasets of unseen images.

### 4.5.6 Probe and Gallery Examples

In Figure 4.7, we show examples of gallery and probe images that are used in testing. Figure 4.7 (a) shows the gallery images in 0° from MultiPIE. Each subject only has one frontal image for reference. Figure 4.7 (b) shows probe images of various pose and expression from MultiPIE. Each subject presents all possible poses and expressions such as neutral, happy, surprise, etc. The illumination is controlled with plain front lighting. Figure 4.7 (c) shows the gallery images from 300WLP, with two near-frontal images of each subject randomly selected. Figure 4.7 (d) shows all poses of the same subject from 300WLP. The pose angle

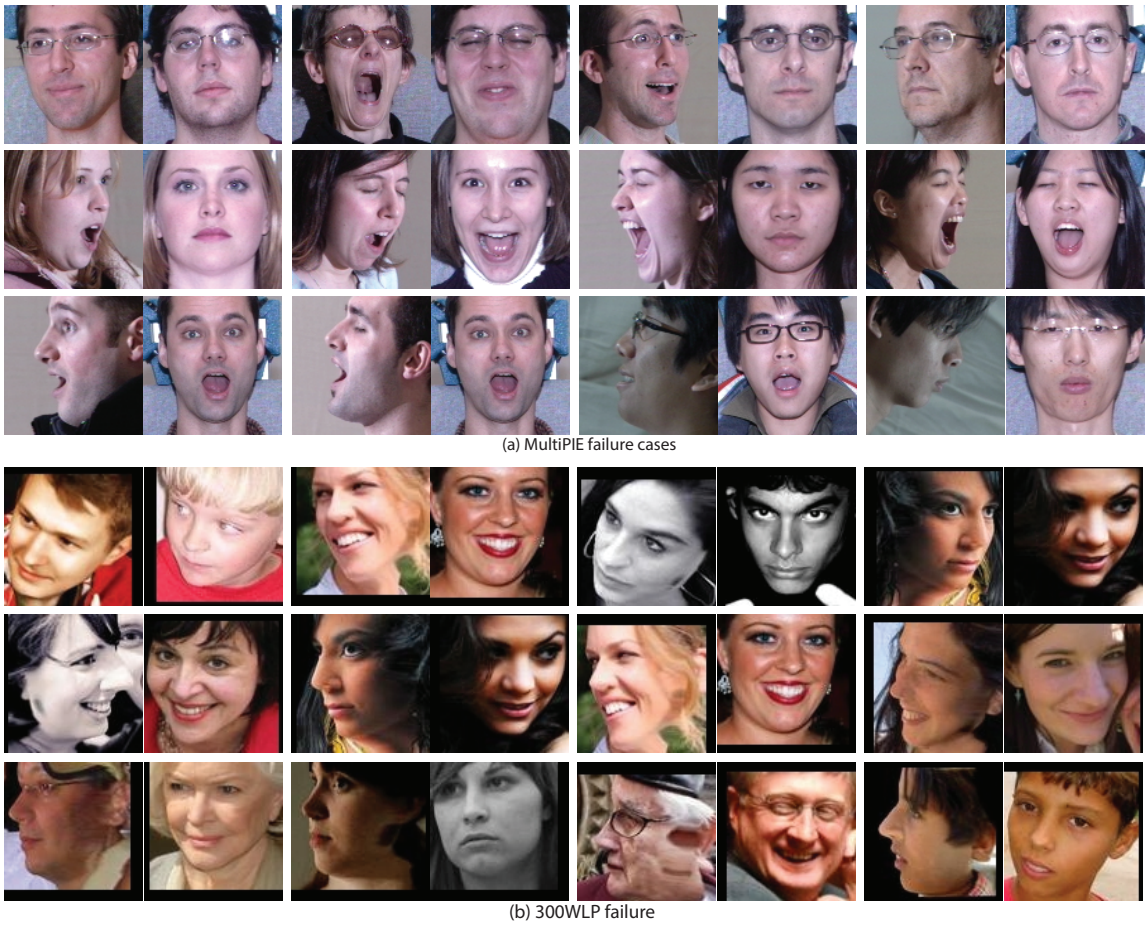



Figure 4.8: Some failure cases in MultiPIE [34] and 300WLP [160]. Each case consists of a pair of images. The gallery image is on the left and the probe image is on the right. In both (a) and (b), the first row shows cases of 15° and 30°, the second row shows cases of 45° and 60°, and the third row shows cases of 75° and 90°. (b) follows the same layout as (a). In MultiPIE, most failures result from extensive expressions. In 300WLP, most failures results from the large pose and illumination changes. Images in most failure pairs are visually similar.

variations are continuous as they are generated by 3DMM [11], which could generate face images under arbitrary head poses.

In Figure 4.8, we show the typical failure cases generated by the proposed method on both MultiPIE and 300WLP. For MultiPIE, the most challenging cases come from exaggerated expression variations, e.g. Figure 4.8 (a), the second row. For 300WLP, the challenge mostly come from head pose variations and illumination variations. However, images in most failure pairs are visually similar.

## 4.6 Discussion

In the work, we propose a new reconstruction loss to regularize identity feature learning for face recognition. Such regularization method is portable to other deep face recognition frameworks without extra effort. We also introduce a data synthesization strategy to enrich the diversity of pose, requiring no additional training data. Rich embedding has already shown promising effects revealed by our control experiments, which is interpreted as curriculum learning. To construct the Siamese reconstruction, a multi-source multi-task network is set up for both preparing the identity and non-identity features and improving the feature discrimination ability. The self and cross reconstruction regularization achieves successful disentanglement of identity and pose, to show significant improvements on both MultiPIE, 300WLP and CFP with 2% to 12% gaps. Cross-database evaluation further verifies that our model generalizes well across databases. Future work will focus on closing the systematic gap among databases and further improve the generalization ability.

# Chapter 5

# Conclusion and Future Work

In this study, learning reliable and interpretable representations is one of the fundamental challenges in machine learning and computer vision. Specially, we investigate why and how to factorize the latent factors and decouple embeddings to achieve reliable and robust representations in a supervised or weekly supervised manner. We also investigate learning disentangled representations and its applications in deep visual analysis, which guarantees state-of-the-art performance on multiple tasks including viewpoint estimation, landmark localization, and large-pose recognition. In the future, we plan to further explore the factorization and disentanglement of representation learning in machine learning and deep learning domain to design vision-based perception systems with higher-level intelligence.

## 5.1 Conclusion

Representations learned by deep models do not always manifest consistent meaning along variations: many latent factors are highly entangled. As a result, tremendous data annotations and sophisticated training skills are required, even though flawed representations with undesirable characteristics are still produced from time to time.

Our objective is to decouple the latent factors in a representation space, where factorizable structures are obtained and consistent semantics are associated with different variables. The disentanglement can be learned in an either supervised or self-supervised manner. Especially, we investigate three different visual analysis tasks: head pose estimation, facial landmark tracking, and large-pose pose recognition.

By factorizing and disentangling latent factors of face embeddings, such as Pose (P), Identity (I), Expression (E), and illumination, we can learn various reliable representations with respect to target perspectives: head pose estimation (pose/non-pose modeling), facial landmark tracking (shape-dependent/-independent factorization), and deep face recognition (identity-related/-unrelated disentanglement).

**Head Pose Estimation.** Three-dimensional head pose estimation from a single 2D image is a challenging task with extensive applications. Different from existing approaches that lack the capability to deal with multiple pose-related and -unrelated factors in a uniform way. To address these problems, We propose a coarse-to-fine manifold embedding framework to model pose and non-pose factors for robust head pose estimation [97]. Specially, we define a unit circle and 3-sphere to model the manifold topology on the coarse and fine layer respectively. It can uniformly factorize multiple factors in the proposed instance parametric subspace, where novel inputs can be synthesized in a generative manner. Moreover, our approach can effectively avoid the manifold degradation problem when 3D pose estimation is performed. The results on both experimental and in-the-wild databases demonstrate the strong performance of our approach.

**Facial Landmark Tracking.** Face alignment, especially on real-time or large-scale sequential images, is a challenging task with broad applications. Both generic and joint alignment approaches have been proposed with varying degrees of success. However, many generic methods have limited performance on sequential images with extensive variations. To address these limitations, one the one hand, we propose to exploit sparse coding based incremental learning for personalized ensemble alignment [102]. We sample multiple initial shapes to achieve image congealing within one frame, which enables us to incrementally conduct ensemble alignment by group-sparse regularized rank minimization. At the same time, personalized modeling is obtained by subspace adaptation under the same incremental framework, while correction strategy is used to alleviate model drifting.

On the other hand, we propose a novel recurrent encoder-decoder network model for real-time video-based face alignment [94]. Our proposed model predicts 2D facial point maps regularized by a regression loss, while uniquely exploiting recurrent learning at both spatial and temporal dimensions. At the spatial level, we add a feedback loop connection between the combined output response map and the input, in order to enable iterative coarse-to-fine face alignment using a single network model. At the temporal level, we first decouple the features in the bottleneck of the network into temporal-variant factors, such as pose and expression, and temporal- invariant factors, such as identity information. Temporal recurrent learning is then applied to the decoupled temporal-variant features, yielding better generalization and significantly more accurate results at test time.

**Large-Pose Face Recognition.** Deep neural networks (DNNs) trained on large-scale datasets have recently achieved impressive improvements in face recognition. But a persistent challenge remains to develop methods capable of handling large pose variations that are relatively under-represented in training data. We presents a method for learning a feature representation that is invariant to pose, without requiring extensive pose coverage in training data [101]. We first propose to generate non-frontal views from a single frontal face, in order to increase the diversity of training data while preserving accurate facial details that are

critical for identity discrimination. Our next contribution is to seek a rich embedding that encodes identity features, as well as non-identity ones such as pose and landmark locations. Finally, we propose a new feature reconstruction metric learning to explicitly disentangle identity and pose, by demanding alignment between the feature reconstructions through various combinations of identity and pose features, which is obtained from two images of the same subject. Experiments on both controlled and in-the-wild face datasets show that our method consistently outperforms the state-of-the-art, especially on images with large head pose variations.

The experiments in all the three investigations indicate that, by learning disentangled representations, deep models are efficient to train and robust to variations, achieving state-of-the-art performance in challenging conditions.

## 5.2 Future Work

Following the current work, we plan to further extend the ongoing investigation to generative modeling in 3D structure recovering. Understanding 3D structure from a monocular view is a challenging task due to the ill-posted nature of 2D-to-3D mapping. How to incorporate prior knowledge to complement the missing information and eventually reduce the uncertainty becomes a crucial shortcut to learn the mapping successfully.

**Weekly Supervised Factor Manipulation.** By investigating the relation of latent factors, we aim to achieve the disentanglement in a weekly supervised or even unsupervised manner. The advantage is obvious. We can exploit real-world data, which are nearly unlimited in volume but extremely expensive in labeling, to learn better representations with disentangled nature. More importantly, by utilizing the learned representations, we can manipulate latent factors for attribute editing and novel data generation to reveal deep understandings of perceptions.

**Adversarial Hard Sample Mining.** Given the fact that 3D annotated data are usually collected in laboratory environment with limited volume, an encouraging way to improve the 2D-to-3D mapping is to deeply exploit the training data by mining hard samples. Specially, we extend adversarial learning from the image distribution domain to variation parametric domain, and therefore we can generate not only novel samples but also knowledge about variations to provide hard samples for network training.

# Bibliography

[1] Jania Aghajanian and Simon J.D. Prince. Face pose estimation in uncontrolled environments. In *British Machine Vision Conference*, 2009.

[2] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic. Incremental face alignment in the wild. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[3] Akshay Asthana, Stefanos Zafeiriou, Shiyang Cheng, and Maja Pantic. Robust discriminative response map fitting with constrained local models. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 3444–3451, 2013.

[4] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *CoRR*, 2015.

[5] V.N. Balasubramanian, Jieping Ye, and S. Panchanathan. Biased manifold embedding: A framework for person-independent head pose estimation. In *Computer Vision and Pattern Recognition*, pages 1–7, 2007.

[6] Peter N. Belhumeur, David W. Jacobs, David J. Kriegman, and Neeraj Kumar. Localizing parts of faces using a consensus of exemplars. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[7] Chiraz BenAbdelkader. Robust head pose estimation using supervised manifold learning. In *European Conference on Computer Vision*, volume 6316, pages 518–531, 2010.

[8] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the International Conference on Machine Learning*, ICML '09, pages 41–48, 2009.

[9] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[10] M. Black and Y. Yacoob. Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 374–381, 1995.

[11] Volker Blanz and Thomas Vetter. Face recognition based on fitting a 3D morphable model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1063–1074, 2003.

[12] M.D. Breitenstein, D. Kuettel, T. Weise, L. Van Gool, and H. Pfister. Real-time face pose estimation from single range images. In *Computer Vision and Pattern Recognition*, pages 1–8, June 2008.

[13] Adrian Bulat and Georgios Tzimiropoulos. *Human Pose Estimation via Convolutional Part Heatmap Regression*, pages 717–732. Springer International Publishing, Cham, 2016.

[14] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. FaceWarehouse: a 3D facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):413–425, March 2014.

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

[16] J.-C. Chen, J. Zheng, V.M. Patel, and R. Chellappa. Fisher vector encoded deep convolutional features for unconstrained face verification. In *ICIP*, 2016.

[17] KyungHyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *CoRR*, abs/1409.1259, 2014.

[18] Grigoris G. Chrysos, Epameinondas Antonakos, Stefanos Zafeiriou, and Patrick Snape. Offline deformable face tracking in arbitrary videos. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 954–962, 2015.

[19] T. F. Cootes and C. J. Taylor. Active shape models - smart snakes. In *British Machine Vision Conference*, 1992.

[20] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, January 1995.

[21] D. Decarlo and D. Metaxas. Optical flow constraints on deformable models with applications to face tracking. *International Journal of Computer Vision*, 38(2):99–127, 2000.

[22] Chi Nhan Duong, K Luu, Kha Gia Quach, and T D Bui. Beyond principal components: Deep boltzmann machines for face modeling. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[23] S. Edelman and H. H. Bülthoff. Orientation dependence in the recognition of familiar and novel views of three-dimensional objects. *Vision Research*, 32(12):2385–2400, 1992.

[24] Ahmed Elgammal and Chan-Su Lee. Separating style and content on a nonlinear manifold. In *Computer Vision and Pattern Recognition*, volume 1, pages 478–485, 2004.

[25] Ahmed Elgammal and Chan-Su Lee. Homeomorphic manifold analysis (hma): Generalized separation of style and content on manifolds. *Image and Vision Computing*, 31(4):291–310, 2013.

[26] Gabriele Fanelli, Juergen Gall, and Luc Van Gool. Real time head pose estimation with random regression forests. In *Computer Vision and Pattern Recognition*, pages 617–624, June 2011.

[27] V.F. Ferrario, C. Sforza, G. Serrao, G. Grassi, and E. Mossi. Active range of motion of the head and cervical spine: A three-dimensional investigation in healthy young adults. *J. Orthopaedic Research*, 20(1):122–129, 2002.

[28] FGNet. Talking face video. Technical report, Online, 2004.

[29] Yun Fu and T.S. Huang. Graph embedded analysis for head pose estimation. In *International Conference on Automatic Face and Gesture Recognition*, pages 6–8, 2006.

[30] X. Gao, Y. Su, X. Li, and D. Tao. A review of active appearance models. *IEEE Transactions on Systems, Man, and Cybernetics*, 40(2):145–158, March 2010.

[31] Ross Girshick. Fast r-cnn. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 1440–1448, Washington, DC, USA, 2015. IEEE Computer Society.

[32] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[33] Stephen Gould, Paul Baumstarck, Morgan Quigley, Andrew Y. Ng, and Daphne Koller. Integrating visual and range data for robotic object detection. In *ECCV Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications*, 2008.

[34] R. Gross, I. Matthew, J.F. Cohn, T. Kanade, and S. Baker. Multipie. *Image and Vision Computing*, 2009.

[35] Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and Simon Baker. Multi-pie. *Image and Vision Computing*, 28:807–813, 2010.

[36] M.A. Haj, J. Gonzalez, and L.S. Davis. On partial least squares in head pose estimation: How to simultaneously deal with misalignment. In *Computer Vision and Pattern Recognition*, pages 2602–2609, June 2012.

[37] D. Hardoon, S. Szedmak, and J. Shawe-Taylor. Cannonical correlation analysis: an overview with application to learning methods. *Neural Comput.*, 16, 2004.

[38] Tal Hassner, Shai Harel, Eran Paz, and Roee Enbar. Effective face frontalization in unconstrained image. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[39] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition, 2016 IEEE Conference on*, 2016.

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

[41] Seunghoon Hong, Hyeonwoo Noh, and Bohyung Han. Decoupled deep neural network for semi-supervised semantic segmentation. *CoRR*, abs/1506.04924, 2015.

[42] N. Hu, W. Huang, and S. Ranganath. Head pose estimation by non-linear embedding and mapping. In *International Conference on Image Processing*, volume 2, pages II–342–5, 2005.

[43] Qiong Hu, Xi Peng, Peng Yang, Fei Yang, and Dimitris N Metaxas. Robust multi-pose facial expression recognition. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 1782–1787. IEEE, 2014.

[44] Dong Huang, M. Storer, F. De la Torre, and H. Bischof. Supervised local subspace learning for continuous head pose estimation. In *Computer Vision and Pattern Recognition*, pages 2921–2928, 2011.

[45] G.B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, 2007.

[46] J. Huang, Xuhui Shao, and H. Wechsler. Face pose discrimination using support vector machines (svm). In *International Conference on Pattern Recognition*, volume 1, pages 154–156, 1998.

[47] Junzhou Huang, Xiaolei Huang, Wang Yan, and Dimitris N. Metaxas. Learning with dynamic group sparsity. In *International Conference on Computer Vision*, 2009.

[48] Junzhou Huang, Shaoting Zhang, Hongsheng Li, and Dimitris Metaxas. Composite splitting algorithms for convex optimization. *Computer Vision and Image Understanding*, 115(12):1610–1622, 2011.

[49] John E. Hummel and Irv Biederman. Dynamic binding in a neural network for shape recognition. *Psychological Review*, 99(3):480–517, 1992.

[50] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015.

[51] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, pages 675–678, 2014.

[52] A. Jourabloo and X. Liu. Large-pose face alignment via cnn-based dense 3d model fitting. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[53] M. Kan, S. Shan, H. Chang, and X. Chen. Stacked progressive auto-encoders (spae) for face recognition across poses. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[54] M. Kan, S. Shan, H. Zhang, S. Lao, and X. Chen. Multi-view discriminant analysis. In *European Conference on Computer Vision*, 2012.

[55] Meina Kan, Shiguang Shan, and Xilin Chen. Multi-view deep network for cross-view classification. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[56] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *The IEEE Conference on Computer Vision and Pattern Recognition*, June 2015.

[57] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *CoRR*, abs/1511.02680, 2015.

[58] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR: 1412.6980*, 2014.

[59] Martin Koestinger, Paul Wohlhart, Peter M. Roth, and Horst Bischof. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *Workshop on Benchmarking Facial Image Analysis Technologies*, 2011.

[60] Iasonas Kokkinos. Rapid deformable object detection using dual-tree branch-and-bound. In *The Conference on Neural Information Processing Systems*, 2011.

[61] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2012.

[62] Tejas D Kulkarni, William F. Whitney, Pushmeet Kohli, and Josh Tenenbaum. Deep convolutional inverse graphics network. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2015.

[63] H. Lai, S. Xiao, Z. Cui, Y. Pan, C. Xu, and S. Yan. Deep cascaded regression for face alignment. In *CoRR:1510.09083v2*, 2015.

[64] Lieven De Lathauwer, Bart de Moor, and Joos Vandewalle. A multilinear sigular value decomposition. *SIAM Journal On Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[65] Vuong Le, Jonathan Brandt, Zhe Lin, Lubomir Bourdev, and Thomas S. Huang. Interactive facial feature localization. In *European Conference on Computer Vision*, pages 679–692, 2012.

[66] Gary B. Huang Erik Learned-Miller. Labeled faces in the wild: Updates and new reporting procedures. Technical Report UM-CS-2014-003, University of Massachusetts, Amherst, 2014.

[67] Yann Lecun, LÃľon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[68] Haoxiang Li and Gang Hua. Hierarchical-pep model for real-world face recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[69] S.Z. Li, QingDong Fu, Lie Gu, Bernhard Scholkopf, Yimin Cheng, and Hongjiag Zhang. Kernel machine based learning for multi-view face detection and pose estimation. In *International Conference on Computer Vision*, volume 2, pages 674–679 vol.2, 2001.

[70] S. Liang, L.G. Shapiro, and I. Kemelmacher-Shlizerman. Head reconstruction from internet photos. In *European Conference on Computer Vision*, 2016.

[71] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3730–3738, 2015.

[72] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *CoRR*, abs/1411.4038, 2014.

[73] Jonathan L Long, Ning Zhang, and Trevor Darrell. Do convnets learn correspondence? In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 1601–1609. 2014.

[74] Liang Lu, Xingxing Zhang, KyungHyun Cho, and Steve Renals. A study of the recurrent neural network encoder-decoder for lar ge vocabulary speech recognition. In *INTERSPEECH*, 2015.

[75] Y. Ma, Yoshinori Konishi, K. Kinoshita, Shihong Lao, and M. Kawade. Sparse bayesian regression for head pose estimation. In *International Conference on Pattern Recognition*, volume 3, pages 507–510, 2006.

[76] Iacopo Masi, Anh Tu an Trãn, Tal Hassner, Jatuporn Toy Leksut, and Gérard Medioni. Do we really need to collect millions of faces for effective face recognition? In *European Conference on Computer Vision*, 2016.

[77] Iacopo Masi, Stephen Rawls, Gerard Medioni, and Prem Natarajan. Pose-aware face recognition in the wild. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[78] Dimitris Metaxas and Shaoting Zhang. A review of motion analysis methods for human nonverbal communication computing. *Image and Vision Computing*, 31(6âĂŞ7):421–433, 2013.

[79] Tomas Mikolov, Armand Joulin, Sumit Chopra, Michaël Mathieu, and Marc'Aurelio Ranzato. Learning longer memory in recurrent neural networks. *CoRR*, abs/1412.7753, 2014.

[80] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In *INTERSPEECH*, 2010.

[81] Stephen Milborrow and Fred Nicolls. Locating facial features with an extended active shape model. In *European Conference on Computer Vision*, pages 504–513, 2008.

[82] D. Miller, I. Kemelmacher-Shlizerman, and S.M. Seitz. Megaface: A million faces for recognition at scale. In *CoRR*, volume 1505.02108, 2015.

[83] E. Murphy-Chutorian and M.M. Trivedi. Head pose estimation in computer vision: A survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(4):607–626, 2009.

[84] Erik Murphy-chutorian, Anup Doshi, and Mohan Manubhai Trivedi. Head pose estimation for driver assistance systems: A robust algorithm and experimental evaluation. In *Proc. 10th Int'l IEEE Conf. Intelligent Transportation Systems*, pages 709–714, 2007.

[85] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *CoRR*, pages 807–814, 2010.

[86] A.A. Nielson. Multiset canonical correlations analysis and multispectral, truly multitemporal remote sensing data. *IEEE Trans. on Image Processing*, 11(3), 2002.

[87] Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard L Lewis, and Satinder Singh. Action-conditional video prediction using deep networks in atari games. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 2845–2853, 2015.

[88] N. Oliver, A. Pentland, and F. Berard. Lafter: Lips and face real time tracker. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 123–129, 1997.

[89] Sharathchandra U Pankanti, Xi Peng, and Nalini K Ratha. Visual object recognition, April 4 2016. US Patent 20170286809A1.

[90] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *British Machine Vision Conference*, 2015.

[91] I. Patras and M. Pantic. Particle filtering with factorized likelihoodsfor tracking facial features. In *Automatic Face and Gesture Recognition*, pages 97–102, 2004.

[92] P Paysan, R Knothe, B Amberg, S Romdhani, and T Vetter. A 3D face model for pose and illumination invariant face recognition. In *IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2009.

[93] X. Peng, N. Ratha, and S. Pankanti. Learning face recognition from limited training data using deep neural networks. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 1442–1447, Dec 2016.

[94] Xi Peng, Rogerio S Feris, Xiaoyu Wang, and Dimitris N Metaxas. A recurrent encoder-decoder network for sequential face alignment. In *European Conference on Computer Vision*, pages 38–56. Springer International Publishing, 2016.

[95] Xi Peng, Rogerio S Feris, Xiaoyu Wang, and Dimitris N Metaxas. Red-net: A recurrent encoder-decoder network for video-based face alignment. *International Journal of Computer Vision*, 2018.

[96] Xi Peng, Qiong Hu, Junzhou Huang, and Dimitris N Metaxas. Track facial points in unconstrained videos. *British Machine Vision Conference*, 2016.

[97] Xi Peng, Junzhou Huang, Qiong Hu, Shaoting Zhang, Ahmed Elgammal, and Dimitris Metaxas. From circle to 3-sphere: Head pose estimation by instance parameterization. *Computer Vision and Image Understanding*, 136:92–102, 2015.

[98] Xi Peng, Junzhou Huang, Qiong Hu, Shaoting Zhang, and Dimitris N Metaxas. Head pose estimation by instance parameterization. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 1800–1805. IEEE, 2014.

[99] Xi Peng, Junzhou Huang, Qiong Hu, Shaoting Zhang, and Dimitris N Metaxas. Three-dimensional head pose estimation in-the-wild. In *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, volume 1, pages 1–6. IEEE, 2015.

[100] Xi Peng, Junzhou Huang, and Dimitris N Metaxas. Sequential face alignment via person-specific modeling in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 107–116, 2016.

[101] Xi Peng, Xiang Yu, Kihyuk Sohn, Dimitris N Metaxas, and Manmohan Chandraker. Reconstruction-based disentanglement for pose-invariant face recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[102] Xi Peng, Shaoting Zhang, Yu Yang, and Dimitris N Metaxas. Piefa: Personalized incremental and ensemble face alignment. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 3880–3888, 2015.

[103] Xi Peng, Shaoting Zhang, Yang Yu, and Dimitris N Metaxas. Toward personalized modeling: Incremental and ensemble alignment for sequential faces in the wild. *International Journal of Computer Vision*, pages 1–14, 2017.

[104] T. Poggio and S. Edelman. A network that learns to recognize 3-dimensional objects. *Nature*, 343(6255):263–266, 1990.

[105] T. Poggio and Girosi F. Network for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990.

[106] R. Rae and H. Ritter. Fecognition of human head orientation based on artificial neural networks. *IEEE Trans. on Neural Networks*, 9(2):160–165, 1998.

[107] Ananth Ranganathan and Ming-Hsuan Yang. Online sparse matrix gaussian process regression and vision applications. In *European Conference on Computer Vision*, volume 5302, pages 468–482, 2008.

[108] Scott Reed, Kihyuk Sohn, Yuting Zhang, and Honglak Lee. Learning to disentangle factors of variation with manifold interaction. In *Proceedings of the International Conference on Machine Learning*, 2014.

[109] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc., 2015.

[110] Salah Rifai, Yoshua Bengio, Aaron Courville, Pascal Vincent, and Mehdi Mirza. Disentangling factors of variation for facial expression recognition. In *European Conference on Computer Vision*, 2012.

[111] J. Roth, Y. Tong, and X. Liu. Unconstrained 3d face reconstruction. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[112] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2 edition, 2003.

[113] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2013.

[114] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2013.

[115] Christos Sagonas, Epameinondas Antonakos, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 300 faces in-the-wild challenge: database and results. *Image and Vision Computing*, 47:3 – 18, 2016.

[116] S. Sankaranarayanan, A. Alavi, C. Castillo, and R. Chellappa. Triplet probabilistic embedding for face verification and clustering. In *CoRR*, volume 1605.05396, 2016.

[117] JasonM. Saragih, Simon Lucey, and JeffreyF. Cohn. Deformable model fitting by regularized landmark mean-shift. *International Journal of Computer Vision*, 91(2):200–215, 2011.

[118] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

[119] S. Sengupta, J.-C. Chen, C. Castillo, V.M. Patel, R. Chellappa, and D.W. Jacobs. Frontal to profile face vefirication in the wild. In *WACV*, 2016.

[120] A. Sharma, A. Kumar, H. Daume III, and D.W. Jacobs. Generalized multiview analysis: A discriminative latent space. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[121] J. Shen, S. Zafeiriou, G. Chrysos, J. Kossaifi, G. Tzimiropoulos, and M. Pantic. The first facial landmark tracking in-the-wild challenge: Benchmark and results. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015.

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

[123] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *CoRR*, 2014.

[124] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2016.

[125] Richard Souvenir and Robert Pless. Image distance functions for manifold learning. *Image and Vision Computing*, 25(3):365–373, 2007.

[126] Chan su Lee and Ahmed Elgammal. Homeomorphic manifold analysis: Learning decomposable generative models for human motion analysis. In *Workshop on Dynamical Vision*, 2005.

[127] Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang. Proceedings of the International Conference on Neural Information Processing Systems. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 1988–1996. 2014.

[128] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep convolutional network cascade for facial point detection. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 3476–3483, 2013.

[129] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation from predicting 10,000 classes. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[130] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deeply learned face representations are sparse, selective, and robust. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 2892–2900, 2015.

[131] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[132] Ming Tang and Xi Peng. Robust tracking with discriminative ranking lists. *IEEE Transactions on Image Processing*, 21(7):3273–3281, 2012.

[133] Ming Tang, Xi Peng, and Duowen Chen. Robust tracking with discriminative ranking lists. In *Asian Conference on Computer Vision*, pages 283–295. Springer, 2010.

[134] Anh Tuan Tran, Tal Hassner, Iacopo Masi, and Gérard G. Medioni. Regressing robust and discriminative 3d morphable models with a very deep neural network. *CoRR*, abs/1612.04904, 2016.

[135] L. Tran, X. Yin, and X. Liu. Disentangled representation learning gan for pose-invariant face recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[136] G. Tzimiropoulos. Project-out cascaded regression with an application to face alignment. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 3659–3667, 2015.

[137] L.J.P. van der Maaten. Accelerating t-sne using tree-based algorithms. *Journal of Machine Learning Research*, 2014.

[138] Vivek Veeriah, Naifan Zhuang, and Guo-Jun Qi. Differential recurrent neural networks for action recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, December 2015.

[139] Jing Wang, Yu Cheng, and Rogerio Schmidt Feris. Walk and learn: Facial attribute representation learning from egocentric video and contextual data. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[140] X. Wang, M. Yang, S. Zhu, and Y. Lin. Regionlets for generic object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(10):2071–2084, Oct 2015.

[141] Xiaolong Wang, Guodong Guo, Michele Merler, Noel CF Codella, MV Rohith, John R Smith, and Chandra Kambhamettu. Leveraging multiple cues for recognizing family photos. *Image and Vision Computing*, 58:61–75, 2017.

[142] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *European Conference on Computer Vision*, 2016.

[143] Junwen Wu and Mohan M. Trivedi. A two-stage head pose estimation framework and evaluation. *Pattern Recognition*, 41(3):1138–1158, 2008.

[144] Yue Wu and Qiang Ji. Constrained joint cascade regression framework for simultaneous facial action unit recognition and facial landmark detection. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[145] Xuehan-Xiong and Fernando De la Torre. Supervised descent method and its application to face alignment. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[146] Jimei Yang, Scott Reed, Ming-Hsuan Yang, and Honglak Lee. Weakly-supervised disentangling with recurrent transformations for 3d view synthesis. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2015.

[147] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. Describing videos by exploiting temporal structure. In *Proceedings of the IEEE International Conference on Computer Vision*, December 2015.

[148] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Learning face representation from scratch. In *CoRR*, 2014.

[149] Lijun Yin, Xiaochen Chen, Yi Sun, T. Worm, and M. Reale. A high-resolution 3d dynamic facial expression database. In *International Conference on Automatic Face and Gesture Recognition*, pages 1–6, 2008.

[150] Xi Yin, Xiang Yu, Kihyuk Sohn, Xiaoming Liu, and Manmohan Chandraker. Towards large-pose face frontalization in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[151] AlanL. Yuille, PeterW. Hallinan, and DavidS. Cohen. Feature extraction from faces using deformable templates. *International Journal of Computer Vision*, 8(2):99–111, 1992.

[152] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pages 818–833, 2014.

[153] Jie Zhang, Shiguang Shan, Meina Kan, and Xilin Chen. Coarse-to-fine auto-encoder networks (CFAN) for real-time face alignment. In *European Conference on Computer Vision*, pages 1–16, 2014.

[154] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Facial landmark detection by deep multi-task learning. In *European Conference on Computer Vision*, pages 94–108, 2014.

[155] Yi Zhou, Lie Gu, and Hong-Jiang Zhang. Bayesian tangent shape model: Estimating shape and pose parameters via bayesian inference. In *Computer Vision and Pattern Recognition*, pages 109–116, 2003.

[156] Shizhan Zhu, Cheng Li, Chen Change Loy, and Xiaoou Tang. Face alignment by coarse-to-fine shape searching. In *The IEEE Conference on Computer Vision and Pattern Recognition*, June 2015.

[157] X. Zhu, Z. Lei, X. Liu, H. Shi, and S.Z. Li. Face alignment across large poses: A 3d solution. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[158] Xiangxin Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Computer Vision and Pattern Recognition*, pages 2879–2886, 2012.

[159] Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Z. Li. Face alignment across large poses: A 3d solution. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[160] Xiangyu Zhu, Zhen Lei, Junjie Yan, Dong Yi, and Stan Z. Li. High-fidelity pose and expression normalization for face recognition in the wild. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[161] Z. Zhu, P. Luo, X. Wang, and X. Tang. Deep learning identity-preserving face space. In *Proceedings of the IEEE International Conference on Computer Vision*, 2013.

[162] Zhenyao Zhu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Multi-view perceptron: a deep model for learning face identity and view representations. In *Proceedings of the International Conference on Neural Information Processing Systems*. 2014.

[163] Zhenyao Zhu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Multi-view perceptron: a deep model for learning face identity and view representations. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2014.