ABSTRACT

Title of dissertation: ROBUST FACIAL LANDMARKS LOCALIZATION WITH APPLICATIONS IN FACIAL BIOMETRICS

Amit Kumar
Doctor of Philosophy, 2019

Dissertation directed by: Professor Rama Chellappa
Department of Electrical and Computer Engineering

Localization of regions of interest on images and videos is a well studied problem in computer vision community. Usually localization tasks imply localization of objects in a given image, such as detection and segmentation of objects in images. However, the regions of interests can be limited to a single pixel as in the task of facial landmark localization or human pose estimation. This dissertation studies robust facial landmark detection algorithms for faces in the wild using learning methods based on Convolution Neural Networks.

Detection of specific keypoints on face images is an integral pre-processing step in facial biometrics and numerous other applications including face verification and identification. Detecting keypoints allows to align face images to a canonical coordinate system using geometric transforms such as similarity or affine transformations mitigating the adverse affects of rotation and scaling. This challenging problem has become more attractive in recent years as a result of advances in deep learning and release of more unconstrained datasets. The research community is pushing bound-

aries to achieve better and better performance on unconstrained images, where the images are diverse in pose, expression and lightning conditions.

Over the years, researchers have developed various hand crafted techniques to extract meaningful features from features, most of them being appearance and geometry-based features. However, these features do not perform well for data collected in unconstrained settings due to large variations in appearance and other nuisance factors. Convolution Neural Networks (CNNs) have become prominent because of their ability to extract discriminating features. Unlike the hand crafted features, DCNNs perform feature extraction and feature classification from the data itself in an end-to-end fashion. This enables the DCNNs to be robust to variations present in the data and at the same time improve their discriminative ability.

In this dissertation, we discuss three different methods for facial keypoint detection based on Convolution Neural Networks. The methods are generic and can be extended to a related problem of keypoint detection for human pose estimation. The first method called Cascaded Local Deep Descriptor Regression uses deep features extracted around local points to learn linear regressors for incrementally correcting the initial estimate of the keypoints. In the second method, called KEPLER, we develop efficient Heatmap CNNs to directly learn the non-linear mapping between the input and target spaces. We also apply different regularization techniques to tackle the effects of imbalanced data and vanishing gradients. In the third method, we model the spatial correlation between different keypoints using Pose Conditioned Convolution Deconvolution Networks (PCD-CNN) while at the same time making it pose agnostic by disentangling pose from the face image. Next, we show an application

of facial landmark localization used to align the face images for the task of apparent age estimation of humans from unconstrained images.

In the fourth part of this dissertation we discuss the impact of good quality landmarks on the task of face verification. Previously proposed methods perform with reasonable accuracy on high resolution and good quality images, but fail when the input image suffers from degradation. To this end, we propose a semi-supervised method which aims at predicting landmarks in the low quality images. This method learns to predict landmarks in low resolution images by learning to model the learning process of high resolution images. In this algorithm, we use Generative Adversarial Networks, which first learn to model the distribution of real low resolution images after which another CNN learns to model the distribution of heatmaps on the images. Additionally, we also propose another high quality facial landmark detection method, which is currently state of the art.

Finally, we also discuss the extension of ideas developed for facial keypoint localization for the task of human pose estimation, which is one of the important cues for Human Activity Recognition. As in PCD-CNN, the parts of human body can also be modelled in a tree structure, where the relationship between these parts are learnt through convolutions while being conditioned on the 3D pose and orientation. Another interesting avenue for research is extending facial landmark localization to naturally degraded images.

ROBUST FACIAL LANDMARK LOCALIZATION WITH APPLICATIONS IN FACIAL BIOMETRICS

by

Amit Kumar

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2019

Advisory Committee:
Professor Rama Chellappa, Chair/Advisor
Professor Behtash Babadi
Professor Larry Davis
Professor Vishal Patel
Professor Ramani Duraiswami

© Copyright by
Amit Kumar
2019

Dedication

Dedicated to my parents, who have always provided unconditional support throughout my life.

Acknowledgments

While the rest of the dissertation is meant to convey the technical work done, this is the only place to take the liberty to express my personal gratitude, I owe to all the people who have made this thesis possible and because of whom my graduate experience has been one that I will cherish forever.

First and foremost I'd like to thank my advisor, Professor Rama Chellappa for giving me an invaluable opportunity to work on challenging and extremely interesting projects over the past four years. He has always been supportive and has given me freedom to pursue research in many directions. He has always made himself available for help and advice and there has never been an occasion when I've knocked on his door and he hasn't given me time. It has been a pleasure to work with and learn from such an extraordinary individual.

It is an honor to have Professor Larry Davis, Professor Behtash Babadi, Professor Abhinav Shrivastava and Professor Vishal Patel in my dissertation committee. I am thankful to them for serving in my committee and providing insightful and diverse suggestions to improve this dissertation.

I am thankful to Professor Vishal Patel and Dr. Jun-Cheng Chen, and Dr. Swami Sankaranarayanan, and other UMIACS graduate students for intense and fruitful research discussions that led to a good number of publications.

I owe my deepest thanks to my family - my mother and father who have always stood by me and guided me through my career, and have pulled me through against impossible odds at times. Words cannot express the gratitude I owe them.

This research is based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA R&D Contract No. 2014-14071600012 ,2019-022600002 and D17PC00345. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

Table of Contents

Bibliography 141

List of Tables

List of Figures

Chapter 1: Introduction

Interpretation and analysis of faces are fundamental functions of the human vision system and it improves social interaction. Recently, with the increase in the use of portable image and video recording devices, the trend has been shifting towards automatic face analysis in uncontrolled scenarios. To achieve a fully automatic face analysis system, a face detector and a robust facial landmark detector is crucial.

More generally, localization in images refers to detecting or segmenting objects in a given image. However, regions of interests can be limited to a single pixel. One such task is facial landmark localization which refers to automatically detecting important keypoints in a face such as eye corners, nose tip. Localizing regions of interest is extremely challenging and has been researched quite extensively in the literature. Objects vary in appearance and appear in variety of shapes and scale. Humans appear in different poses and are usually occluded. Face images can be captured under extreme pose, occlusion or resolution. This dissertation studies robust facial landmark detection algorithms for faces in the wild using learning methods based on Convolution Neural Networks.

In general, an automatic face analysis system comprises four main steps: face detection, face association, facial landmark localization and face alignment, facial

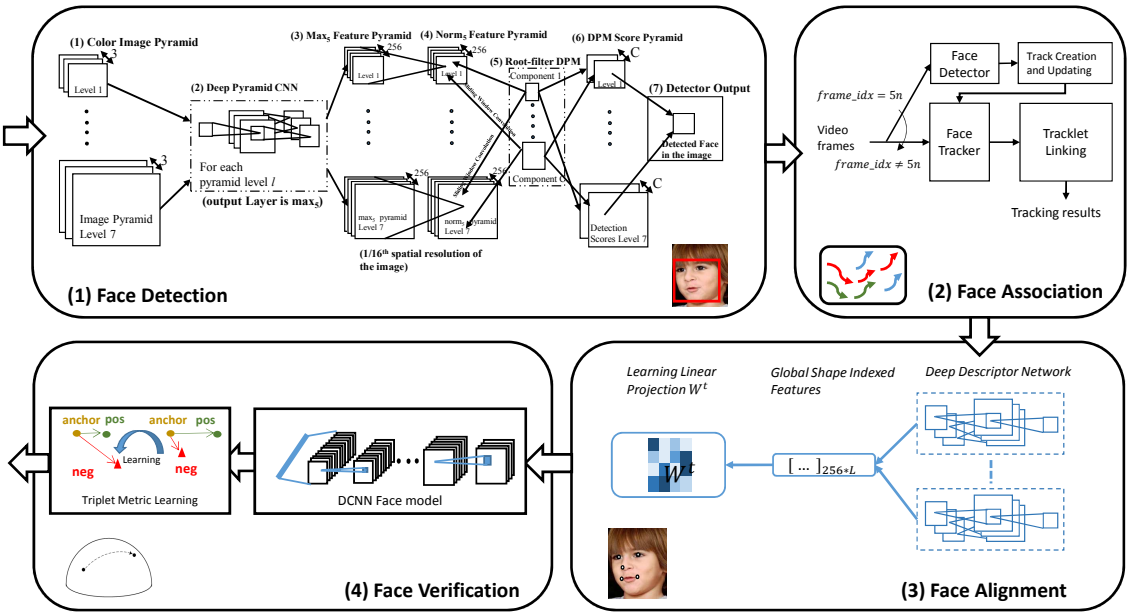


Figure 1.1: Face alignment in a face analysis system

feature extraction and face analysis as illustrated in Figure 1.1. Facial landmark localization is an integral component in almost every facial biometric task such as face identification, face synthesis, 3D modeling of faces. These landmarks are used to align faces which mitigates the effects of in-plane rotation and scaling. Facial landmarks are used both directly and indirectly. Typical direct applications include facial expression analysis [147] where landmarks are used to decode specific set of emotions or non-verbal message and marker-less motion capture [139] where landmarks assist in computer generated imagery. To the category of the indirect applications of facial landmark detection belong all applications where the facial landmarks are used for some pre-processing, for example: face verification [30, 80]; 3D face reconstruction [120], where, for instance, the landmarks are used to aid the structure from motion algorithm; head-pose orientation [27] where a 3D face model is fitted to estimated 2D landmark positions; face tracking; other face processing

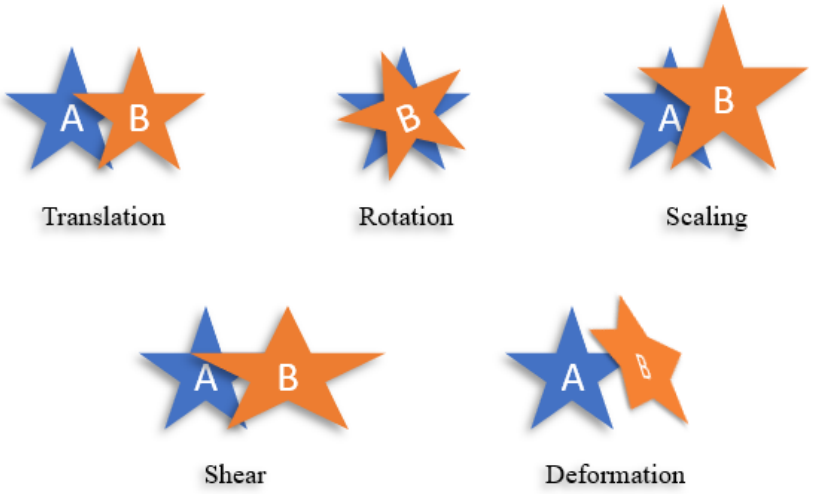


Figure 1.2: Rigid image transformations: translation, rotation, scale, shear; Nonrigid image transformations and out of plane rotation: deformation. The face alignment poses all these five transformations.

tasks like prediction of gender, age, expression, or other facial attributes [46].

Detection of facial landmarks in uncontrolled environments is a non-trivial problem for several reasons. The key factor is a large intra-class variability of the input image due to the change of position, scale, and rotation of the face, lighting conditions, background clutter, facial expression, occlusions, and self-occlusions, hair style, make-up, race, aging, modality (webcam, camera, scanned image) and so on. Figure 1.2 illustrates different transformations in an image and shows due to the deformable nature of human face, the problem of landmark detection is extremely challenging.

With the advent of Deep Convolutional Neural Networks facial biometrics problems such as facial landmark detection has received a great deal of attention from the computer vision community. DCNNs have been shown to be very effective for several computer vision tasks like image classification [64, 122, 137], and object

detection [55, 117]. Deep CNNs (DCNNs) are highly non-linear regressors because of the presence of hierarchical convolutional layers with non-linear activations. Not only this, deep networks have shown to improve the performance of face landmark detection by a large margin [89, 171, 176]. Existing methods for facial key-points localization task have focused primarily on detecting essential landmarks for frontal faces (pose yaw angles in between -60° and 60°). Most of these methods fail to correctly localize key-points for off-frontal or profile faces which occur frequently in images collected in unconstrained settings. Moreover, manually annotating facial key-points locations is a tedious task and hence it is very difficult to collect large number of training samples to train a DCNN for this task.

1.0.1 Proposed Methods

In the second part of this dissertation, we discuss a deep learning-based method called Local Deep Descriptor Regression addressing the task of facial keypoint localization. The proposed method consists of several stages of feature extraction followed by linear regression. It is worth noting that networks trained for the task of face detection/face verification have abstract information about the structure of face. Hence, such a network is used for feature extraction, which are then used to design linear regressors. The spatial resolution of the areas used for feature extraction is reduced in a step-wise manner to achieve better localization over the image space.

Chapter 3 discusses another cascade regression based method called, KE-

PLER. This method shows an application of multi-tasking in Convolution Neural Networks, where a single network is used to jointly estimate the facial keypoints and their visibility and 3D head pose. Information is pooled from shallow as well as deeper layers of the network to achieve better localization. Some of the practical issues, such as vanishing gradients are tackled by designing improved loss functions and using smart training policies such as hard sample mining and local error correction.

We propose a Convolution-Deconvolution network, where we decouple the tasks of facial keypoint localization and 3D head pose estimation by learning them in two different networks. This makes the network agnostic to facial pose. We also model the spatial correlation between different keypoints in a tree-structure the weights of which are learned through convolutions. The proposed network, called Pose-Conditioned-Dendritic CNN is able to precisely estimate the keypoints in a single step which makes it fast and easy to deploy in real life scenarios.

Chapter 5 discusses an application of the facial keypoint localization in context of apparent age estimation from unconstrained images. The detected faces are first aligned using Local Deep Descriptor Regression after which the aligned faces are used to train and age group classifier and age regression networks. We also develop an error correction strategy after observing the fact that classifiers makes mistakes between the boundary of age groups.

In chapter 6 of this dissertation we discuss the impact of good quality landmarks on the task of face verification. Previously proposed methods perform with reasonable accuracy on high resolution good quality images, but fails when the in-

put image suffers from degradation. To this end, we also propose a semi-supervised method which aims at predicting landmarks on the low quality images. This method learns to predict landmarks on low resolution images by learning to model the learning process of high resolution images. In this algorithm, we use Generative Adversarial Networks, which first learn to model the distribution of real low resolution images after which another CNN learns to model the distribution of heatmaps on the images. Additionally, we also propose another high quality facial landmark detection method, which is currently state of the art.

We also discuss some ongoing work and future plans of localizing facial landmarks in naturally degraded images such as turbulent images. We also plan to extend the ideas developed for facial keypoint localization to other tasks such as human pose estimation and action recognition from human poses.

Organization: Chapter 2 discusses in detail the proposed Local Deep Descriptor Regression, followed by the discussion of KEPLER in chapter 3. In chapter 4 we discuss Pose-Conditioned Dendritic CNN proposed for one step and faster facial alignment. In chapter 5 we discuss, method to address the problem of apparent age estimation from unconstrained images. Chapter 6 discusses the strategy of landmark localization in Low resolution images. Finally in chapter 7 we conclude the discussion by presenting future plans of extension and open issues in landmark localization.

Chapter 2: Local Deep Descriptor Regression

2.1 Introduction

Most of the recent methods use discriminative shape regression approach to estimate the face landmark positions. With their ability to utilize large amount of training data, and enforce shape constraints adaptively, regression-based methods have achieved state-of-the-art performance on various unconstrained face alignment datasets. However, the success of these methods is limited by the strength of the features they use. In previous works, the features used are either hand crafted ; for example SIFT was used as features in [158], or learned from a limited set of training samples [25, 116].

In recent years, features obtained using deep CNNs have yielded impressive results for various computer vision applications. They significantly outperform methods proposed earlier for the tasks of face detection and recognition. It has been shown in [84] that a deep CNN pre-trained with a large generic dataset such as Imagenet [122], can be used as a meaningful feature extractor. Although these features are effective for reliable classification, they are global in nature. Hence, this approach may not be effective for problems such as face alignment where local features are desirable. To overcome this problem, Overfeat [130] uses predicted detection bound-

Figure 2.1: We present a deep descriptor-based regression approach for fiducial point extraction. This figure shows fiducial points extracted on all the detected faces on an image from the IJB-A [81] dataset using the proposed method.

aries, but lacks the needed pixel-based localization feature. [138] and [48] propose pixel-based localization, the former based on the Restricted Boltzmann machine while the latter processes the image to determine a key-point descriptor.

In this chapter, we address the localization problem in existing deep CNNs by constructing a deep convolutional key-point descriptor model. We build a network which takes a small local image patch around a pixel as an input and produces a feature vector as the output. We claim that the proposed deep descriptor network can be used as a substitute for SIFT [100] descriptors in most vision problems. To support our claim, we apply the descriptor model for facial landmark detection. Local features calculated for a small rectangular patch around each estimated landmark position are used by a linear regressor to learn the shape increment during training, and predict the landmark positions at test time. Figure ?? shows several faces where our method is able to locate fiducial points on all the detected faces. Overall, this chapter makes the following contributions:

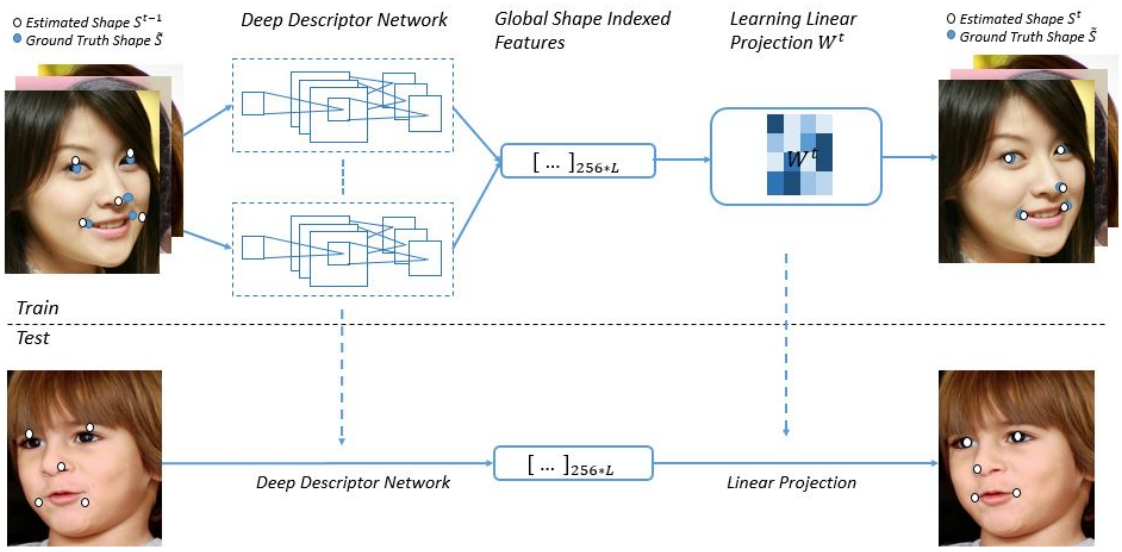


Figure 2.2: Overview of the proposed method. During training, we extract *deep descriptors* for each landmark and concatenate them to form a shape-indexed feature vector. Given these features and target shape increments ΔS_i^t, we learn the linear regression weights W^t. During testing, deep descriptors are extracted around each point of the initialized mean shape. Intermediate shape is predicted using the regressor weights W^t. This process is iterated to reach the final estimated shape.

1. We construct a novel deep descriptor network to evaluate the local features for a given key-point.

2. We perform face alignment by applying linear regression to the deep descriptors evaluated for facial landmarks.

This chapter is organized as follows. Section 2.2 reviews a few related works. Details of our deep descriptor-based face alignment method are given in Section 2.3. Section 6.4.3 provides the landmark localization results on five challenging datasets. Finally, Section 2.5 concludes the chapter with a brief summary and discussion.

2.2 Previous Work

The task of face alignment can be classified broadly into three categories depending on the approach.

2.2.1 Model-based Approaches

Model-based approaches learn a shape model during training and use it to fit new faces during testing. The pioneering works of Cootes *et al.* such as Active Appearance Models (AAM) [36] and Active Shape Models (ASM) [35] were built using PCA constraints on appearance and shape. In recent years many improvements over these models have been proposed in [57, 58, 95, 105, 128, 141]. In [37], Cristinacce and Cootes generalised the ASM model to a Constrained Local Model (CLM), in which every landmark has a shape constrained descriptor to capture the appearance. In [127], a more sophisticated local model and mean shift was used to obtain good results. However, these methods depend upon the goodness of the error cost function and how well it is optimised. For example, AAM estimates the shape by minimizing the texture residual. Recently, Antonakos *et al.* [7] proposed a method along similar lines by modeling the appearance of the object using multiple graph-based pairwise normal distributions (Gaussian Markov Random Field) between the patches extracted from the regions. However, the learned models lack the power to capture complex face image variations in pose, expression and illumination. Also, they are sensitive to initialization due to gradient descent optimization, a critical step.

2.2.2 Regression-based Approaches

Since face alignment is naturally a regression problem there has been a plethora of regression-based approaches in recent years. These methods learn a regression model that directly maps image appearance to target output. But the performance of these methods depends on the robustness of local descriptors. Sun *et al.* [135] proposed a cascade of carefully designed CNNs in which at each level outputs of multiple networks are fused for landmark estimation. Our work is different from [135], in that we use a single CNN carefully designed to provide a unique key-point descriptor. Xiong *et al.* [158] predicts the increment in shape by applying linear regression on SIFT features. Burgos *et al.* [151] proposed a cascade of T-regressors to estimate the pose in image sequence using pose-indexed features. Cao *et al.* [25] sequentially learned a cascade of random fern regressors using pixel intensity difference as the feature and regresses the shape stage-wise over the learnt cascade. They performed regression on all parameters simultaneously, thus effectively exploiting the shape constraint. Following this, Sun *et al.* [116] proposed cascaded regression using fern regressors and local binary features. Subsequently, Burgos *et al.* [24] extended their work to face alignment with occlusion handling, enhanced shape indexed features and more robust initialization which they refer to as Robust cascaded pose regression (RCPR). Li *et al.* [159] combined multiple final shapes from multiple initializations in a cascade regression manner using weights matrices learnt to combine these hypotheses accurately. Recently, Lee *et al.* [93] proposed a Gaussian Process Regression face alignment method based on the responses of the Gaussian filters

around the patches extracted from the region adjacent to intermediate landmarks. Finally, Zhu *et al.* [174] proposed a hierarchical face alignment , starting from a coarse shape estimate and refining it to reach the target landmark. Also, Xiong et al. [157] proposed the global supervised descent method where they consider direct optimization over the landmarks independent of any shape model.

2.2.3 Part-based Deformable Models

Part-based deformable models perform alignment by maximizing the posterior likelihood of part locations given an input image I. The models vary in the optimization techniques or the shape priors used. In [126] Saragih *et al.* used a method similar to mean shift to optimize the posterior likelihood. Recently, Saragih [125] developed a sample specific prior which significantly improves over the original PCA prior in ASM , CLM and AAM. Zhu and Ramanan [177] used a part-based model for face detection, pose estimation and landmark localization assuming the face shape to be a tree structure. Asthana *et al.* [9] combined discriminative response map fitting with CLM, which learns a dictionary of probability response maps based on local features and adopts linear regression-based fitting in the CLM framework.

2.3 Regression of Deep Descriptors

The proposed method for facial landmark detection, called **Local Deep Descriptor Regression** (LDDR), consists of two modules. The first module generates local features for each estimated facial landmark points using the deep descriptor frame-

work. These features are concatenated together to form a global shape-indexed feature. The second module is a linear regressor which learns the relationship between the shape feature and the corresponding shape increment during training. The process is repeated stage-by-stage in a cascaded fashion. Figure 2.2 shows the overview of our method.

2.3.1 Deep Descriptor Construction

In order to construct a deep CNN descriptor, we start with the Alexnet [84] network. We use the publicly available network weights trained on the Imagenet [122] data using Caffe [72], that are distributed with RCNN [55] . However, this particular CNN cannot be used directly as a key-point descriptor because of the following limitations. Firstly, the CNN requires a fixed input image size of 224×224 pixels which is too large to be considered for the patch size around the key-point. Secondly, a single activation unit at the fifth convolutional layer ($conv_5$) has a highly overlapping receptive field of size 195×195 pixels, which makes localization difficult. As a result, two pixel points in close vicinity cannot be distinguished from one another.

On further analysis of the first problem, we found that a CNN requires fixed size input only because of its fully-connected layers. A convolutional layer can process any input as long as it is larger than the convolutional kernel. On the other hand, a fully connected layer needs a fixed size input as its output dimension is predetermined. To resolve this issue, we remove the last max pooling layer ($pool_5$) and all the subsequent fully-connected layers (fc_6, fc_7, fc_8, and softmax) from the

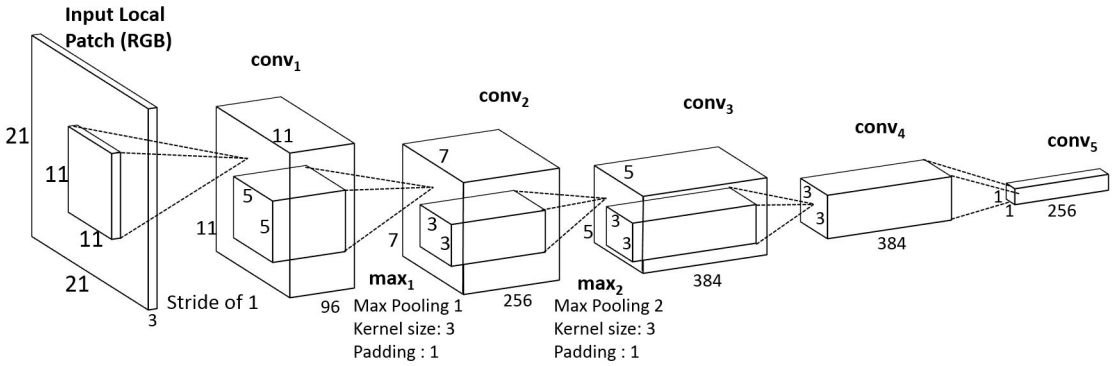


Figure 2.3: Architecture of the proposed Deep Descriptor Network. The height and width represents the dimensions of each feature map, whereas the depth denotes the number of features maps for a given layer. The number of strides for each layer is restricted to 1.

network. The CNN output is, therefore, computed by the $conv_5$ layer containing 256 feature channels. Analyzing the second problem, we find that a major contributor for the large size of receptive field is the inter-layer subsampling operation, which is implemented in the form of strides in the convolutional as well as max pooling layers. They are deployed mainly to reduce the number of parameters and feature computation time, which are not required for a key-point descriptor since the small patch input will drastically bring down the convolution time anyway. Hence, strides in all the existing layers are set to 1. Also, padding from all the convolutional layers are removed as they contribute very little to describing a key-point. Instead, we apply a single pixel padding in the max pooling layer to further reduce the size of the receptive field without altering the output. With these architectural changes, the receptive field size is reduced to 21×21 pixels which is good enough for the size of a local patch surrounding a key-point. The final network structure obtained for the deep descriptor is shown in Figure 2.3. With the input size as small as the

receptive field, single pixel feature maps are obtained at the $conv_5$ layer forming a 256 dimensional output vector.

The proposed deep descriptor satisfies the essential properties of being a key-point descriptor. It is position independent, as it depends only on the image patch relative to the point. It is robust to small geometric transformations because of the max pooling operation in CNN. The normalization operation after each convolutional layer makes it robust to illumination variations. Since the network weights are trained using fixed sized inputs, the descriptor works best when the input images are scaled to the same size prior to key-point extraction, thus reducing the dependency on scale. Hence it can be used as a generic keypoint descriptor in many computer vision applications. Additionally, for domain specific problems, the model weights can be fine-tuned before evaluating the features. For the application of face alignment, we fine-tune the model weights using face images from the FDDB [70] dataset. Fine-tuning was done for the face detection task, which classifies the input as face or non-face. The procedure adopted is similar to the method described in [55]. During fine-tuning, the network learns features specific to face parts which is a crucial part in our work. As a result, the activations at the fifth convolutional layer become more discriminatory to local face patches such as eyes, nose, lips, etc. The other advantage of fine-tuning is that the same network weights can be used for both face detection as well as face alignment. Once the network is fine-tuned, the test image just goes through a forward pass to generate CNN features, which are then fed to a simple linear regression method to generate incremental shapes.

2.3.2 Computing Shape Indexed Features

Given an initial mean shape containing L landmarks, we compute the 256 dimensional deep descriptor ϕ_l^t for each landmark $l \in 1, 2, ...L$ at a given stage t. A global shape indexed feature is composed by concatenating the set of deep descriptors, i.e., $\Phi^t = [\phi_1^t, \phi_1^t, ..., \phi_L^t]$, which is subsequently used to learn the ground truth shape increment, as explained in section 2.3.3.

We adopt a coarse to fine regression approach. It is important in face alignment that the features used to describe the landmark points are local. To predict the offset Δs of a single landmark, we extract the deep descriptors from a local region of size r. It has been shown in [116] that the optimal size is almost linear to the standard deviation of individual shape increment Δs. Since, we want Δs to decrease sharply at every stage, we need to choose the size of the local patch region around the landmark accordingly. Following [116], we keep the patch size for deep descriptor larger in the first stage and decrease it linearly in subsequent stages. With this modification, the deep descriptor is bound to generate higher dimensional output for the initial stages. Additional structural modification is needed for uniform output dimension, which limits us to consider only four stages of regression. The patch sizes normalized by face rectangle are taken to be $0.4, 0.3, 0.2, 0.1$ for respective stages. Since the face is resized to 224×224 pixels (the input face size used for fine-tuning), the actual patch sizes correspond approximately to $92, 68, 42, 21$. Moreover, variable amounts of strides are added to $conv_1$, max_1, $conv_2$ and max_2 layers for each stage as listed in Table 2.1. The network for the last stage remains unchanged as its input

patch size matches the requirement for our deep descriptor network. This ensures a consistent output dimension of 256 at each stage and for every landmark. In addition to just removing the fully connected layers, our network has reduced the amount of subsampling/stride for different regression stages as shown in Table 2.1.

Stage 1	Input Size (pixels)	conv1	max1	conv2	max2
Stage 1	92×92	4	2	1	1
Stage 2	68×68	3	2	1	1
Stage 3	42×42	2	1	1	2
Stage 4	21×21	1	1	1	1

Table 2.1: Input size and the number of strides in conv1, max1, conv2 and max2 layers for 4 stages of regression.

2.3.3 Learning the Global Regression

In this section, we introduce our basic shape regression methodology for the face alignment problem. Unlike [25] and [116] which have two level cascaded regression framework, we perform a single global regression at each stage. Given a face image I and initial shape S^0, the regressor computes the shape increment ΔS from the deep descriptors and updates the face shape using (2.1)

$$S^t = S^{t-1} + W^t \Phi^t(I, S^{t-1}) \tag{2.1}$$

After extracting the deep descriptors, we concatenate them to a form a global shape-indexed feature $\Phi^t = [\phi_1^t, \phi_1^t, ..., \phi_L^t]$. Our aim is to learn a global linear projection

W^t by minimizing the following objective function:

$$\min_{W^t} \sum_{i=1}^{N} \|\Delta \tilde{S}_i^t - W^t \Phi^t(I_i, S_i^{t-1})\|_2^2 + \lambda \|W^t\|_2^2, \tag{2.2}$$

where the first term is the regression target and the second term is a regularization of W^t in $L2$ sense. The parameter λ controls the strength of regularization. Regularization here plays a major role due to the high dimensionality of the shape-indexed feature. In the experiments, the dimensionality of features for 68 landmarks points could be as high as 17K+. Without regularization there could be substantial amount of over-fitting. For implementing regression, we use L2 regularized L2-loss support vector regression using the LIBLINEAR [47] package. Since the objective function is quadratic in W^t, we can always reach a global minimum.

2.3.4 Incorporating Shape Constraint

As mentioned in [25], the shape constraint is preserved by learning a vector regressor and explicitly minimizing the shape alignment error as in (2.2). Since each shape is updated in an additive manner, and each shape increment is a linear combination of certain training shapes, the final shape is modeled as a linear combination of the initial shape S^0 and all training shapes:

$$S = S^0 + \sum_{i=1}^{N} w_i \hat{S}_i. \tag{2.3}$$

Hence, as long as the initial shape satisfies the shape constraint, the regressed final shape is bound to lie in the linear subspace constructed by all the training shapes. As a matter of fact all the intermediate shapes also satisfy the shape constraint, since they are constructed in a similar fashion.

2.4 Experiments

There are several landmark annotated datasets publicly available. However, we choose the most recent and challenging ones. These are Helen [90], LFPW [12], AFW [177] and IBUG [124]. In addition to these, we evaluate the performance of our method on a recently introduced IARPA Janus Benchmark A (IJB-A) dataset [81]. These datasets present different variations in face shape, appearance and pose and are described in the following subsections. To maintain consistency in the experiments, we perform face alignment using Multi-PIE [59] 68 point markup format.

2.4.1 Datasets

LFPW [12] is one of the widely used datasets to benchmark the face alignment tasks. It consists of 811 training and 220 testing images. The dataset contains unconstrained images from the internet which have large variations in pose, illumination and expression. Since some of the image links mentioned in the dataset are invalid, we downloaded the LFPW images from the ibug [124] website which has accumulated all valid images and their 68 point annotations.

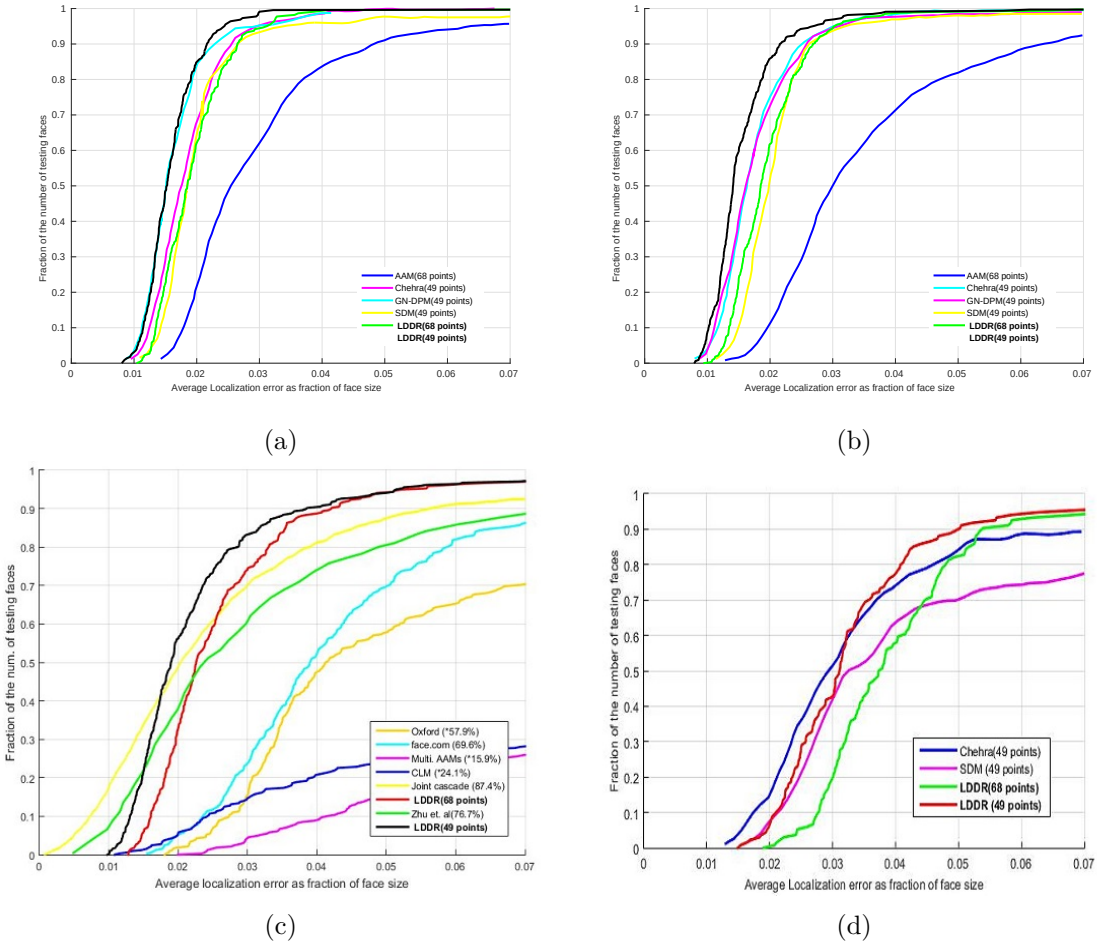


Figure 2.4: Average pt-pt error (normalized by face size) vs fraction of images in (a) LFPW, (b) Helen, (c) AFW and (d) iBUG.

Helen [90] dataset has 2300 high resolution web images, each one marked with 194 landmark points. To be consistent with the 68 point markup in our experiments, we downloaded this dataset from the ibug website which provides the 68 point annotations along with this dataset.

AFW has annotated faces in the wild dataset created by Zhu and Ramanan [177]. It consists of 205 in-the-wild-faces with varying illumination, pose, attributes and expressions. It was originally annotated with six landmark points. However, we perform our experiment on the AFW dataset provided on ibug website, as it contains 68 points annotated ground truth helping us to maintain consistency in the experiments.

IBUG is a challenging subset of 135 images taken from the *300-W* [124] dataset. *300-W* contains IBUG and images from existing datasets LFPW, Helen, AFW and XM2VTS [106]. It inherently follows the 68 point annotation format.

IJB-A [81] dataset is the recently released face verification dataset. The dataset is annotated with three key-points on the faces (two eyes and nose base). The dataset contains images and videos from 500 subjects collected from online media. In total, there are 67,183 faces of which 13,741 are from images and the remaining are from videos. The locations of all faces in the IJB-A dataset were manually annotated by human annotators. The subjects were captured so that the dataset contains wide geographic distribution. The faces in this dataset have significant variations in pose, illumination and resolution.

Training and testing: We evaluated the performance of our method on these challenging datasets. First, we performed training and testing on the LFPW and

Helen datasets taking only their own training and testing sets. Using this model we test on AFW dataset. In order to evaluate on the IBUG dataset, we generated our own cumulative training set consisting of 3148 images taken from the LFPW, Helen and AFW datasets. This is done since AFW has more pose variations compared to LFPW and Helen. To test on IJBA-A dataset we use the same model.

Evaluation Metric: Following the standards of [25], [12], we computed the average error for all landmarks in an image normalized by the inter-pupil distance. For each dataset, the mean error evaluated over all the images is reported. In the following sub-section, we compare our LDDR algorithm against existing state-of-the-art methods and validate our results. Since the IJB-A dataset has only three annotated points, the interoccular distance error was normalized by the distance between nose tip and the midpoint of the eye centers.

2.4.2 Comparison with state-of-the-art Methods

During training we augmented the data to improve the generalization ability. A single training sample is translated to multiple samples by flipping all the images and then randomly rotating them. Then initial shapes are also randomly assigned. Our method has only one fitting parameter *i.e.* number of stages of regression, which following the principles of [116], [25] has been set to 4 in our case. We compare our results with those reported in [25], [116], [24], [9], [144].

Tables 2.2, 2.3, 2.4 and Figure 2.4 provide the Normalized Mean Square Error and average pt-pt error (normalized by face size) vs fraction of images plots of

Method	*68-pts*	*49-pts*
Zhu *et al. [177]*	8.29	7.78
DRMF [9]	6.57	-
RCPR [24]	6.56	5.48
SDM [158]	5.67	4.47
GN-DPM [144]	5.92	4.43
CFAN [168]	5.44	-
CFSS [174]	4.87	3.78
LDDR	**4.67**	**2.38**

Table 2.2: Averaged error comparison of different methods on the LFPW dataset.

different methods, respectively. In Figure 2.6 we present the comparison of our algorithm with [177], [9] and [79]. Our deep descriptor-based global shape regression method outperforms the above mentioned state-of-the-art methods. The tables also show a comparison of our method with many other pioneering methods such as Gauss Newton based Deformable Part Models [144] and Robust Cascaded Pose Regression (RPCR) [24] and some recent methods like [174]. Figure 2.5 shows some landmark localization results on the five datasets. It can be seen from this figure that the proposed method is able to localize landmarks on near profile faces as well as faces of low resolution, partially visible and expression from the *IJB-A* dataset.

Randomly rotating and flipping doubles the amount of data and hence generalizes the data more while reducing the error by $\sim 2\%$. After the advent of deep learning, it was seen that the $conv_5$ features capture a lot of salient information. Our method depends on the generalization of the *deep descriptors* and hence the increase in the amount of data available for training favors the learning step. After training only on *Helen* and LFPW trainset, we get an error of 5.09% and 5.08%, respectively. However, after training on the cumulative data we achieve improved

Figure 2.5: Qualitative results of our landmark localization method. First row: LFPW, Second row: Helen, Third row: AFW and Fourth row: IBUG. Fifth row: IJB-A.

performance getting 4.76% on the former and 4.67% on the latter. Also, it can be seen from Tables 2.2 and 2.3, the error with 68 landmark points is higher than that with 49 points as the former includes the face contour points. It is evident from our experiments that the proposed method performs better than [177] and [158] where HOG and SIFT were used as their features. Table 2.4 shows the performance of our method on a challenging subset of 300-W ibug dataset. The error in the performance of CFSS [174] is lower than our method. This may be due to the fact that CFSS performs its initial search on the space of multiple mean shapes, whereas we initialize with only one mean shape at test time. We do this to reduce the time and space complexity during training. In our experiments we only flipped and rotated the shapes in contrast to conventional techniques where the shapes are flipped, rotated, translated and scaled. This also demonstrates the discriminatory quality of our Deep Descriptors and how better it can get given a large amount of diversified training data.

Method	*68-pts*	*49-pts*
Zhu *et al. [177]*	8.16	7.43
DRMF [9]	6.70	-
RCPR [24]	5.93	4.64
SDM [158]	5.50	4.25
GN-DPM [144]	5.69	4.06
CFAN [168]	5.53	-
CFSS [174]	4.63	3.47
LDDR	**4.76**	**2.36**

Table 2.3: Averaged error comparison of different methods on the Helen dataset.

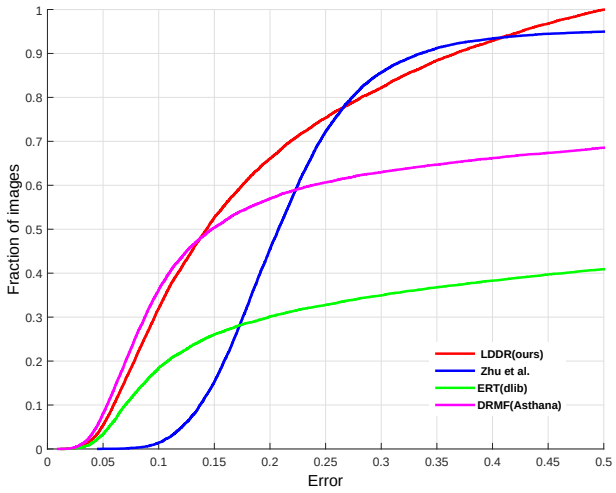


Figure 2.6: Average 3-pt error (normalized by eye-nose distance) vs fraction of images in the IJB-A dataset.

2.4.3 Runtime

All the experiments were performed using an NVIDIA TITAN-X GPU using cudnn library on a 2.3Ghz computer. Training on LFPW took 5.5 hours and on Helen took 9 hours. Training on cumulative data took around 15 hours. Due to different CNN being initialized in each stage, the testing was observed to be slow taking ~ 4 seconds given a face bounding box. *However in our implementation testing was close to real time performance taking only* ~ 0.8 *seconds per face, thereby reducing the testing time by* 80% . This includes the time taken for feature extraction and regression. The time consuming part for the landmark localization was the initialization of a different CNN in each stage. To counter this delay in testing, we merged the 4 CNN models in a single CNN model which is initialized only once. To reduce the performance time even more, the 68 patches extracted around the intermediate shape were passed in a batch.

Method	*68-pts*
Zhu *et al. [177]*	18.33
DRMF [9]	19.75
RCPR [24]	17.26
SDM [158]	15.40
GN-DPM [144]	-
CFAN [168]	-
ESR [25]	17.00
LBF [116]	11.98
LBF Fast [116]	15.50
CFSS [174]	9.98
LDDR	**11.49**

Table 2.4: Averaged error comparison of different methods on the iBUG challenging dataset.

2.5 Conclusions

In this work, we presented a deep descriptor-based method for face alignment using regression of local descriptors. The highly informative nature of *deep descriptor* makes it useful as SIFT, SURF and HOG features. This means *deep descriptors* have potential in many different kinds of applications in machine vision, such as pose estimation, activity recognition and human detection and many others. We also presented an effective way of reducing the testing time by combining four CNNs into one achieving real-time performance. Extensive experiments on five publicly available unconstrained face datasets demonstrate the effectiveness of our proposed image alignment approach.

Chapter 3: KEPLER: Keypoint and Pose Estimation of Unconstrained Faces by Learning Efficient H-CNN Regressors

3.1 Introduction

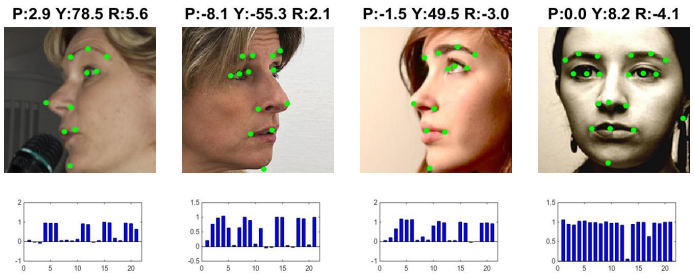


Figure 3.1: Sample results generated by the proposed method. The numbers in black are the predicted 3D pose P:Pitch Y:Yaw R:Roll. Green dots represent the predicted keypoints. The bar graphs show the visibility confidence of each of the 21 keypoints.

In the last five years, keypoint localization using DCNNs has received great attention from computer vision researchers. This is mainly due to the availability of large scale annotated unconstrained face datasets such as AFLW [82]. Recently, Bulat et al. [19] released even larger dataset with more that 200K annotated face images. Works such as [166] have hypothesized that as the network becomes deeper and deeper more semantic information such as identity, pose, attributes are retained while immediate local features are lost. However, various methods such as [135],

[168], and [174] directly used CNNs as regressors or used deeper features from CNNs to design regressors for predicting keypoints. Some of the methods used global features to regress for the keypoints, while others opted for local deep features and train in a coarse to fine manner.

On the other hand, an earlier method known as Explicit Shape Regression (ESR) [25] proposed by Cao et al. achieved superior results by introducing the important concept of non-parametric shape regression for facial keypoint localization. Many of its variants [92], [116], [79], [135], [89] were published later, using a variety of features producing incremental improvements over [25]. However, they are all limited by the fixed number of points on the face. In real life applications, there are more challenging datasets such as IJBA [81] and AFW [177], which do not always have 68 or 49 fixed points due to occlusion or pose variation. As alternatives, researchers moved towards more sophisticated techniques by incorporating 3D shape models [178], [74], [73], domain learning [176], recurrent autoencoder-decoder [1] and many others. The LS3D-W dataset by Bulat et al. [19] is annotated with 34 3D-coordinates. However, in applications such as face recognition, the images are aligned directly from the 2D images/coordinates skipping the 3D mapping step. Thus, unconstrained face alignment on 2D face images has received much attention as an emerging research topic in the recent past. With all the methods in recent years, one question still remains unanswered: Can cascaded shape regression be applied for an arbitrary face with no prior knowledge of its pose ?

The motivation for this work stems from a desire to adapt cascaded regression for predicting landmarks of arbitrary faces, while taking advantage of CNNs. We

transform the cascaded regression formulation into an iterative scheme for arbitrary faces. In each iteration the regressor predicts the increment for the next stage which progressively moves the initial estimate closer to ground truth. By jointly training for all the points, the inherent shape constraint is maintained implicitly. As by-products of KEPLER, we get the visibility confidence of each keypoint and 3D pose (pitch, yaw and roll) for the face image. Figure 3.1 shows a set of sample results from the proposed method, indicating the 3D pose, keypoint locations and their visibility confidences.

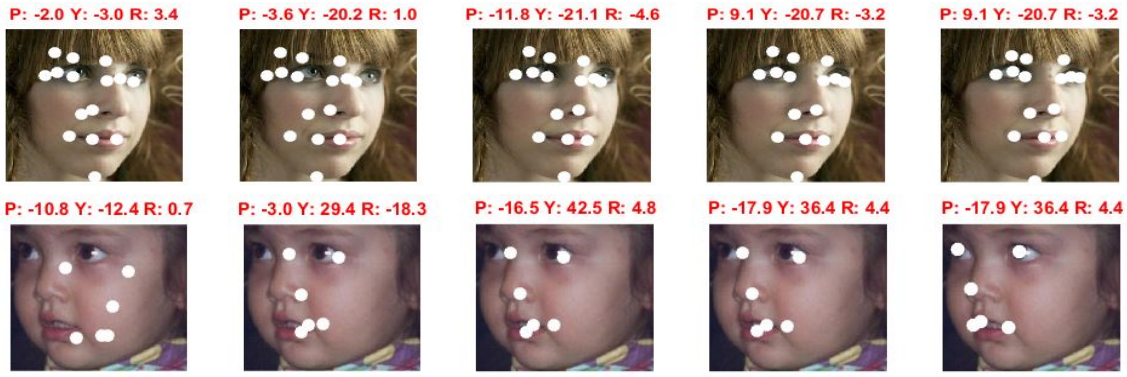


Figure 3.2: Sample results generated by the proposed method KEPLER. White dots represent the location of keypoints after each iteration. The first row shows an image from the AFLW dataset. The points move at subpixel level after fourth iteration. The second row is a sample image from the AFW dataset, which shows how the last stage of error correction can effectively mitigate the inconsistent bounding box across datasets. The numbers in red are the predicted 3D pose P:Pitch Y:Yaw R:Roll

The main contributions of this chapter are:

- We design a novel GoogLenet-based [137] architecture with a channel inception module which pools features from intermediate layers and concatenates them similar to the inception module. We call the proposed architecture *Channeled Inception* in the rest of the chapter. This network is used in all the stages of

KEPLER.

- Inspired by [26], we present an iterative method for estimating the face landmarks using the fixed point consolidation scheme. Fixed point consolidation refers to estimating the error correction in an iterative way by partitioning the total error correction into multiple steps. We observe that estimating landmarks on a face is more challenging than estimating keypoints on a human body. The overview of the pipeline is shown in Figure 3.3.

- After each stage, the error from ground-truth decreases, making the gradient smaller. This is because regression-based approaches use Euclidean loss, the gradient of which also depends on the error. Hence we employ different training policies in each stage for the efficient training of H-CNNs. Figure 5.1 shows how by correcting the estimates of landmark points locally, the issues of inconsistent bounding box can be handled.

- We evaluate the performance of the landmark estimation method on challenging benchmark datasets AFLW, AFW which include faces in diverse pose, expressions and occlusion. Different from many previous methods such as [158], [176], this work estimates variable number of points depending on the head pose. We also introduce a new protocol for evaluating the facial keypoint localization scheme on AFLW which is more challenging and usually left out while evaluating unconstrained face alignment methods.

This chapter builds upon KEPLER [86] by Kumar et. al by evaluating KEPLER on two other challenging datasets. To test the robustness of the proposed method

during deployment, we evaluate it on the datasets with images of qualities different from which KEPLER was trained on. Without retraining or finetuning we test the proposed method on COFW [23] dataset which is a standard benchmark dataset for evaluating face alignment schemes designed to work on images under heavy internal and external object occlusion. We show that it performs comparable to methods such as RCPR [23] which uses COFW training set to develop the method. We also evaluate the method on the IJB-A dataset which is one of the most challenging datasets publicly available for face verification. Without finetuning, we test the performance of the proposed method on IJB-A. We show in Figure 3.13 that earlier methods [9], [79] which yielded good performance on high resolution images, almost fails on IJB-A dataset. However, due to efficient training scheme of KEPLER, it is able to yield improved landmark estimates on images with lower resolution and extreme head-pose.

The rest of the chapter is organized as follows. Section II reviews closely related works. Section III presents the proposed method in detail. Section IV describes the experiments and comparisons, which are then followed by conclusions and suggestions for future works in section V.

3.2 Related Work

Following [25], we classify the previous works on face alignment into two basic categories.

Part-Based Deformable models: These methods perform alignment by maximizing the likelihood of part locations in the given input image. One of the major works in this category was done by Zhu and Ramanan [177], where they used a part-based model for face detection, pose estimation and landmark localization assuming the face shape to be a tree structure. Discriminative Response Map Fitting (DRMF) [9] by Asthana et al., learned a dictionary of probability response maps followed by linear regression in a Constrained Local Model (CLM) framework. However, it is widely acknowledged that the formulation based on CLMs is non-convex, and may converge to local minima. Hsu et al. [66] extended the mixture of tree model [177] in a coarse to fine manner to achieve improved accuracy and efficiency. However, their method again assumes face shape to be a tree structure, enforcing strong constraints specific to shape variations. However, formulating keypoint detection problem as a classification problem, Kumar et al. [87] attempted to capture the structural relationships between different keypoints through convolutional filter assuming the keypoints to be arranged in a tree structure.

Regression-based approaches: A multitude of regression-based approaches has been proposed in recent years by formulating the keypoint detection as a regression problem using local or global features. Methods reported in [95], [25], [174] are based on learning a regression model that directly maps image appearance to target outputs. Different low-level features such as Local Binary Patterns (LBP) [3], Histogram of Oriented Gradients (HOG) [39], Scale Invariant Feature Transform (SIFT) [101] have been used in a variety of regression methods such as Support Vector Regression and Random Forests. However, these methods along with methods

from [6], [142], [5], [8], [144] and [134] were mostly evaluated either in a lab setting or on face images where all the facial keypoints are visible. These methods depend highly on the bounding box annotation and hence the training data is augmented by jittering the images to accomodate for different bounding box annotation. However, when evaluated on challenging datasets such as IJB-A, these methods do not yield accurate results as we show in section 3.4 in Figure 3.13. Wu et al. [155] proposed an occlusion-robust cascaded regressor to handle occlusion by including two separate models for landmark localization and visibility estimation in an iterative way. Xiong et al. [157] pointed out that standard cascaded regression approaches such as Supervised Descent Method (SDM) [158] tend to average conflicting gradient directions resulting in reduced performance. Hence, Xiong et. al [157] suggested domain dependent descent maps. Inspired by this, Cascade Compositional Learning (CCL) [176] and Ensemble of Model Regression Trees (EMRT) [175] developed head pose-based and domain selective regressors respectively. [176] partitioned the optimization domain into multiple directions based on head pose and learned to combine the results of multiple domain regressors through composition estimator function. Similarly, [175] trained an ensemble of random forests to directly predict the locations of keypoints for a given face image, and face alignment is then achieved by aggregating the consensus of different models.

Recently, methods based on 3D models have been proposed for aligning faces. PIFA [73] by Jourabloo et al. proposed a 3D approach that employed cascaded regression to predict the coefficients of 3D to 2D projection matrix and the base shape coefficients. Another recent work from Jourabloo et al. [74] formulated the

face alignment problem as a dense 3D model fitting problem, where the camera projection matrix and 3D shape parameters were estimated by a cascade of CNN-based regressors. However, [176] suggests that optimizing the base shape coefficients and projection is indirect and sub-optimal since smaller parameter errors are not necessarily equivalent to smaller alignment errors. 3DDFA [178] by Zhu et al. fitted a dense 3D face model to the image via CNN, by modelling the depth data in a Z-Buffer. In [15, 98] authors used the 3D-morphable model to learn the 3D camera projection matrix parameters and warping parameters while simultaneously training for 2D face alignment. Although these methods provide 3D coordinates of the keypoints for a given image, they do not outperform the state of the art methods for 2D face alignment. This can be attributed to the fact that learning 3D points from 2D data is a complex problem where the groundtruth data itself is noisy. In contrast, KEPLER simultaneously learns the keypoints, visibility and pose directly from the 2D image, and hence is able to capture the inherent structural dependencies among them. We show that, even without finetuning, KEPLER performs comparable to the state of the art methods on the COFW dataset.

Our work falls in the category of regression-based approaches and addresses the issue of adapting the cascade shape regression to unconstrained settings. Different from all other previous works, it performs joint training on the three fundamental tasks simultaneously, namely, 3D pose, visibility of each keypoint and the location of keypoints. It also demonstrates that efficient joint training on the three tasks achieves superior performance. One of the closely related work is [172] where the authors used multi-tasking for many attributes, but did not leverage the intermedi-

ate features.

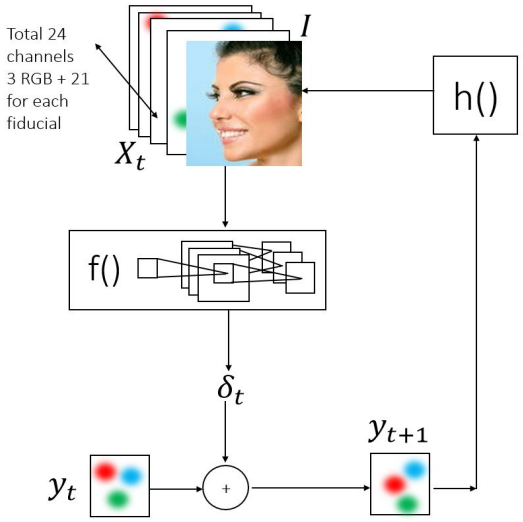


Figure 3.3: Overview of the architecture of KEPLER. The function *f()* predicts the visibility, pose and the corrections for the next stage. The representation function *h()* forms the input representation for the next iteration.

3.3 KEPLER

KEPLER is an iterative method which consists of three modules. Figure 3.3 illustrates the basic building blocks of KEPLER. The first module is a rendering module h which models the structure in an N-dimensional input space, with N being the maximum number of keypoints on a face image. The current locations of the keypoints are represented by the vector $\mathbf{y_t} = \{y_t^1 \dots y_t^N\}$. The output of the rendering module is concatenated to the raw RGB input image $\mathbf{I}$, along the channel dimension which is then fed to the function f.

The second module is the function f which calculates the correction to be made at the next stage. The function f is modeled by a convolution neural network whose architecture is described in section 3.3.1.

The third module is the correction stage which adds the increments, predicted by f, to the current locations. The output goes again into the rendering module h which prepares the rendered data for the next iteration. The rendering function is not learned in this work, but represented by a 2D Gaussian with a fixed variance and centered at current keypoint locations in each of the N channels. Finally, the Gaussian rendered images are stacked together with image $\mathbf{I}$. Therefore the overall method can be summarized by the following set of equations.

$$\boldsymbol{\delta_t} = f_t(\mathbf{X_t}, \Theta_t) \tag{3.1}$$

$$\mathbf{y_{t+1}} = \mathbf{y_t} + \boldsymbol{\delta_t} \tag{3.2}$$

$$\mathbf{X_{t+1}} = h(\mathbf{y_{t+1}}) \tag{3.3}$$

where f is a function with learned parameters Θ_t. The prediction function f is indexed by t as it is trained separately for every iteration. In the first iteration, the function h renders Gaussians at y_0, which is the mean shape. In this work we set $t = 5$ iterations. We perform the last iteration only to take into effect the improper bounding box across different datasets(see Figure 5.1). After four stages of global corrections, no significant improvement was observed on the validation set and hence we adopted local corrections as the last stage of KEPLER. The loss functions for each task is mentioned below.

Keypoint localization

Keypoint localization is the task of predicting the keypoints in a face. In this chapter, we consider predicting the locations of $N = 21$ keypoints on the face. With

each point is associated the visibility of that point. The loss function for this task is given by

$$L_1(\mathbf{y}, \mathbf{g}) = \sum_{i=1}^{N} v^i (y_t^i - g^i)^2, \tag{3.4}$$

where y_t^i and g^i are the predicted and the ground truth locations of the i^{th} keypoint resprectively at time t. v^i is the ground truth visibility associated with each keypoint. According to this formulation of the keypoint loss, since there is no penalty for invisible points, there is no gradient back-propagated for such points. We discuss this loss function and its variant in detail in section 3.3.3.

Pose Prediction

Pose prediction refers to the task of estimating the 3D pose of the face. We use the Euclidean loss function for pose prediction.

$$L_2(\mathbf{p}_p, \mathbf{g}_p) = (p_{yaw} - g_{yaw})^2 + (p_{pitch} - g_{pitch})^2 + (p_{roll} - g_{roll})^2 \tag{3.5}$$

where p stands for predicted and g for the ground-truth. In an alternate formulation this task can be constructed as a classification problem where the face images are to be classified into different classes. However, this will result in binning of pose into discrete bins. Since, we have access to accurate 3D pose, we use the Euclidean loss for this task.

Visibility

This task is associated with estimating the visibility of each keypoint.The number of keypoints visible on the face varies with pose. Hence, we use the Euclidean loss

to estimate the visibility confidence of each point.

$$L_3(\mathbf{v}_p, \mathbf{v}_g) = \sum_{i=1}^{N} (v_{p,i} - v_{g,i})^2, \tag{3.6}$$

Alternatively, one can also use multi target cross-entropy loss for this task.

Therefore the net loss in the network is the weighted linear combination of the above loss functions.

$$L(p, g) = \lambda L_1(\mathbf{y}, \mathbf{g}) + \mu L_2(\mathbf{p}_p, \mathbf{g}_p) + \nu L_3(\mathbf{v}_p, \mathbf{v}_g) \tag{3.7}$$

where λ, μ and ν are the weight parameters suitably chosen depending on the iteration.

3.3.1 Network Architecture

For the modeling function f we design a unique ConvNet architecture based on GoogLenet [137] by pruning the inception network after inception_4c. As PReLU has shown better performance in many vision tasks such as object recognition [63], in this pruned network we first replace the ReLU non-linearity with PReLU. We pool the intermediate features from the pruned GoogLenet. Then convolutions are performed from the output of each branch, and the output maps are concatenated similar to inception module. We call this module as the *Channeled Inception* module. Since the output maps after $conv_1$ are larger in size, we first perform $4X4$ convolution and then again a $4X4$ convolution, both with the stride of 3 to finally match the dimension

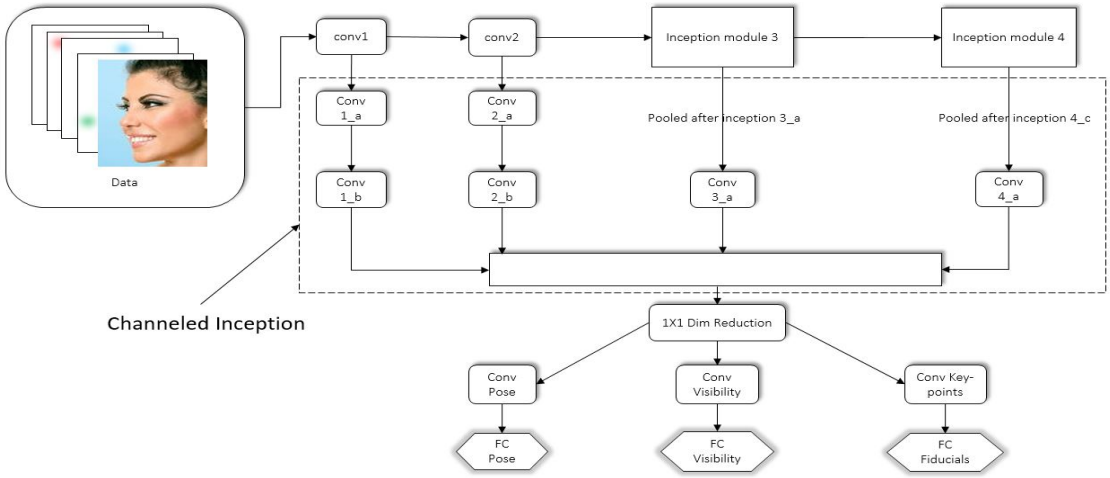


Figure 3.4: The KEPLER network architecture. The dotted line shows the channeled inception network. The intermediate features are convolved and the responses are concatenated in a similar fashion as inception module. Tasks such as pose are abstract and contained in deeper layers, however, the localization property is in the shallower layers.

of the output to $7X7$. Similarly, after $conv_2$ we first perform $4X4$ convolution and then $3X3$ convolution to match the output to $7X7$. The former uses a stride of 4 and the latter uses 2. The most näive way of combining features is by concatenation. However, the concatenated output blob can be very high dimensional and hence we perform $1X1$ convolution for dimensionality reduction. This lets the network decide the weights to effectively combine the pooled features into lower dimension. It has been shown in [166] that adjacent layers are correlated and therefore, we only pool features from alternate layers.

Next, the network is trained on three tasks namely, pose, visibilities and the bounded error using ground truth. The joint training is helpful since it models the inherent relationships among visible number of points, pose and the amount of correction needed for a keypoint in particular pose. Choosing an architecture like GoogLenet is appropriate as it has fewer parameters (as compared to VGG-

Net [131]) and the training of GoogLenet is faster when batch normalization is added after each convolution layer. In order to further speed up the process we only use convolution layers till the last layer where we use a fully connected layer to get the final output. Recently, Residual Networks [64] with skip connections have been proposed where the number of parameters is even fewer; furthermore these networks have achieved improved classification results on the Imagenet [40] classification task. In that case the backbone network in each stage of the whole pipeline of KEPLER can be replaced by a ResNet, while keeping the training process same. The architecture of the whole network is shown in Figure 3.4.

3.3.2 Iteration 1 and 2: Constrained Training

In this section, we explain the first stage training for keypoint estimation. The first stage is the most crucial one for face alignment. Since the network is trained from scratch, precautions have to be taken on what the network learns. Directly learning the locations of keypoints from a network is difficult not only because of highly non-linear mapping between input and target space, but also because when the network gets deeper it loses the localization capability. This is due to the fact that the outputs of the final convolution layers have a larger receptive field on the input image. We devise a strategy in which the corrections for the first two stages are bounded. Let us suppose the key-points are represented by their 2D coordinates $\mathbf{y} : \{y^i \in \Re^2, i \in [1, \ldots, N]\}$ where N is the number of keypoints and y^i denotes the

i^{th} keypoint. The bounded corrections were calculated using (3.8) given below.

$$\delta_t^i(g^i, y_t^i) = min(L, \|\mathbf{u}\|).\hat{\mathbf{u}} \tag{3.8}$$

where L denotes the bound of correction. $\mathbf{u} = \mathbf{g} - \mathbf{y_t}$ and $\hat{\mathbf{u}} = \frac{\mathbf{u}}{\|\mathbf{u}\|}$ represent the error vector and error unit vector respectively. In our experiments, we set the bound L to a maximum of 20 pixels. This simplifies the learning problem for the network for the first stage. According to this formulation, error correction for points for which the ground truth is far away, gets bounded by L. The interesting property of this formulation is that in the first and second stages the network only learns the direction in which the points have to shift. This can be thought of as learning the direction of the error unit vector, to which the magnitude will be added later. In addition to just having keypoint location we also have access to facial 3D pose and the visibility of each point. One-shot prediction of the location of keypoints is difficult since the input-output mapping is typically nonlinear. Also, learning small corrections should be easier, when the network is being trained for the first time. Hence, to impart the prior knowledge to the network we jointly learn the pose and visibility of each point. The loss functions used for the three tasks are described in equations (3.4, 3.5, 3.6) in the previous section3.3.

The function f for second iteration is trained in a similar fashion with the weights initialized from the first iteration.

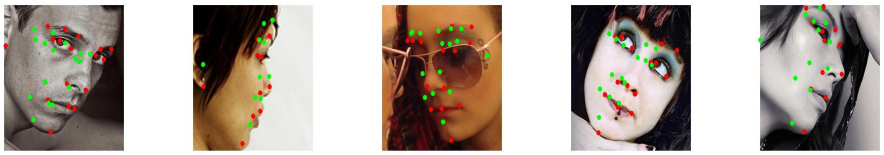


Figure 3.5: Qualitative results of KEPLER after second stage. The green dots represent the predicted points after second stage. Red dots represent the ground truth. It can be seen that the visible points have taken the shape of input face image.

3.3.3 Iteration 3: Variant of Euclidean loss

We show the outputs of the network after the second stage of training in Figure 3.5. Visual inspection of the outputs shows that for many of the faces, the network has already learned the magnitude and direction of the correction vector. However, there are misalignments in some images or in some keypoints in the images. But repeating the training methodology exactly as second iteration revealed that our architecture suffered from vanishing gradients. While back propagating the gradients, the loss is averaged over a batch and if there are few misalignments in a batch, there is very little gradient to be propagated. To maintain consistency we stick with the same architecture. Even though GoogLenet [137] claims to not have vanishing gradient problem, KEPLER faced it because of the absence of intermediate supervision which GoogLenet originally had.

This motivated us to design a loss function that satisfies both of these conditions: on the one hand, the loss function should minimize the error between prediction and the ground truth; on the other hand, it should have sufficient gradients to be propagated to make the learning process reach global minima. Towards this end,

we use the following loss function.

$$L_1(\mathbf{y}, \mathbf{g}) = \frac{1}{n}\left(\sum_{i=1}^{N} v_i(y_i - g_i)^2 + \gamma \sum_{i=1}^{N} v_i \mid y_i - g_i \mid\right) \tag{3.9}$$

$$\frac{\delta L_1(\mathbf{y}, \mathbf{g})}{\delta \mathbf{y}} = \frac{1}{n}\left(2\sum_{i=1}^{N} v_i(y_i - g_i) + \gamma \sum_{i=1}^{N} v_i \frac{\mid y_i - g_i \mid}{y_i - g_i}\right) \tag{3.10}$$

where γ is a parameter which controls the strength of the gradient and n is the number of samples in a batch. We would like to emphasize that the additional term is not a regularizer as it is added to the objective function and does not directly regularize the weights. However, this is able to provide substantial gradients for the training of ConvNet because depending on the sign of difference, second term is always +1 or -1 in equation 3.9(b).

The representation function h in this stage does not render any Gaussian in the channel for which the predicted visibility is below the threshold τ. In this work, we set this threshold τ to 0.03 and γ to 0.2 , which were determined experimentally. Now that the network has learned the unit vectors in first and second iteration, we do not constrain the amount of error corrections for the third stage training.

3.3.4 Iteration 4: Hard sample mining

Face alignement is a task which requires precise localization as error in alignment can propoagate to errors in verification/recognition or other tasks which depend on the aligned image. In our case, although after the third iteration, most of the images are aligned, they lack precision in local alignment. Recently, Kabkab et al. [77] suggested that by efficiently sampling the data one can make an optimal use of training data

while training ConvNets leading to obtain improved performance. [77] developed an online data sampling method based on a convex optimization formulation and showed how their formulation can make the classifier robust when class imbalance is present. Inspired by [77], we reuse the hard samples of the dataset to build a more robust keypoint localization system.

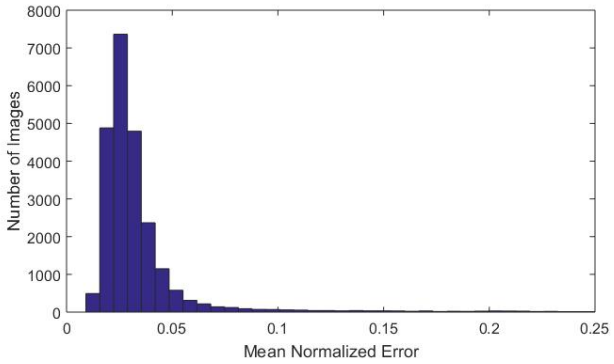


Figure 3.6: Error Histogram of training samples after stage 3

Using the keypoints predicted after the third iteration, we plot the histogram (Fig.3.6) of normalized mean error (NME), after calculating it for all the training samples. We denote the NME on the x-axis at which the maximum number of samples are centered, as C. In an ideal case, the value of C should be low, implying that the average alignment error is less. Therefore, the objective of this stage is to lower the value of C by hard sample mining. We select a threshold $C+\Delta$ (0.03 in our experiments), towards the right of C, after which at least $30-40\%$ of the samples lie, as the threshold for hard samples. Using $C + \Delta$, we partition the dataset into two groups of hard and easy samples. We first select equal number of samples from both groups to form a batch which is then presented to the ConvNet for training. This effectively results in reusing the hard samples since the number of samples in hard

group is much lower than in the easy group. Then, to counter the group imbalance we finetune the network with the whole dataset again with a lower learning rate. We use the loss function as in (3.9) with $\gamma = 0.1$ for this stage.

3.3.5 Iteration 5: Local Error Correction

There is a lot of inconsistency among the bounding boxes provided by different datasets. AFLW [82] provides larger bounding box annotations compared to AFW [177]. Regression-based alignment methods are dependent on the mean shape initialization, which is scaled to the bounding box size. Also it is impractical to come up with a heuristic which tries to determine compatible bounding boxes. Almost all the existing methods perform data augmentation by randomly perturbing the bounding boxes by some amount. However, it is not clear by how much the bounding boxes should be perturbed to obtain reasonably good bounding boxes during testing which is consistent with the dataset the network was trained on. We train our networks on a larger bounding box provided by AFLW. AFLW bounding boxes tend to be square and for almost all the images the nose tip appears at the center of the bounding box. This is a big limitation for the deployment of the system in real world scenarios. It is worth noting that the previous four stages are trained on full images and hence produce global corrections.

Our last stage of local correction is optional, which depends upon the test set and the bounding box annotations that it comes with. We train a similar network as before but only for the tasks of predicting the visibility and corrections in the local

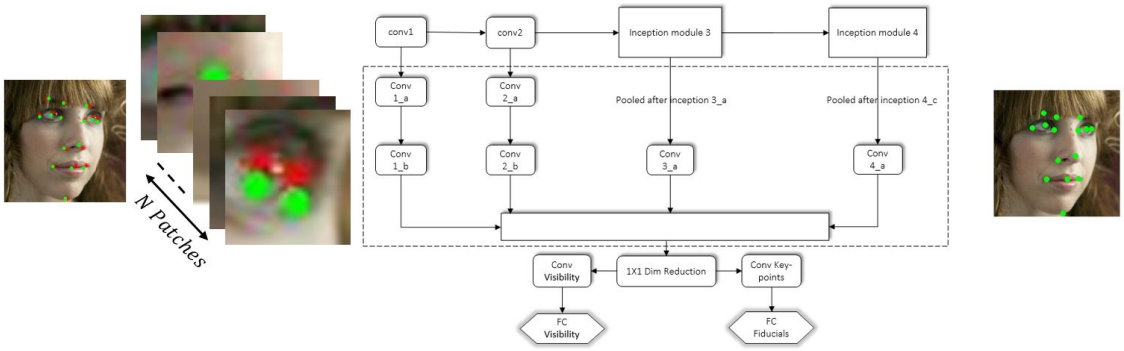


Figure 3.7: Red dots in the left image represent the ground truth while green dots represent the predicted points after the fourth iteration. Local patches centered around predicted points are extracted and fed to the network. The network shown in Fig 3.4 is trained on the task of local fiducial correction and visibility of fiducials inside the patch. The image on the right shows the predictions after local correction.

patches (see Fig 3.7). Predicting the pose with a local patch of say WXW pixels is difficult which can lead the network to learn improper weights. We choose all the N patches irrespective of the visibility factor. Learning visibility and corrections is important because we do not want the network to propagate any gradient if the point is invisible. We observe during experimentation that training the ConvNet on two tasks together achieves significantly better performance than when the network is trained only for the task of error correction. We again partition the dataset into easy and hard sample groups according to the strategy explained in section3.3.4. We finally finetune the network with the whole dataset with a lower learning rate.

3.4 Experiments and Comparison

3.4.1 Datasets

We select two challenging datasets with their most recent benchmarks.

In-the-wild datasets: To make the system robust for images in real life scenarios due to challenging shape variations and significant view changes, we select AFLW [82] for training and, AFLW and AFW [177] as the main test sets.

AFLW contains $24,386$ in-the-wild faces (obtained from *Flickr*) with head pose ranging from $0°$ to $120°$ for yaw and upto $90°$ for pitch and roll with extremely challenging shape variations and deformations. AFLW provides at most 21 points for each face. It excludes coordinates for invisible landmarks, which we consider to be the best, because there is no way of correctly knowing the exact location of those points. In many cases such invisible points are mostly hallucinated and annotated thereafter. Along with this, AFLW also demonstrates a limited amount of external-object occlusion.

COFW is a collection of 1007 face images out which of 507 images are partitioned as the test set. Caltech Occluded Faces in the Wild (COFW) dataset exhibits wide range of images in diverse pose and is mainly used for evaluation of face alignment methods designed to perform on images under extreme occlusion. In addition, one important point to note is that COFW also provides the annotations for the invisible landmarks while in the case of AFLW the invisble landmarks are unavailable.

IJB-A dataset is one of the most challenging face verification dataset. The face images in the dataset are annotated with three key-points ; two eyes and nose base. The dataset contains images and videos from 500 subjects collected from online media. In total, there are 67,183 faces of which 13,741 are from images and the remaining are from videos. The locations of all faces in the IJB-A dataset were manually annotated by human annotators. The images were captured so that the dataset contains wide geographic distribution. The challenge comes through the wide diversity in pose, illumination and resolution.

AFW is a popular benchmark for the evaluation of face alignment algorithms. AFW contains 468 in-the-wild faces (obtained from Flickr) with yaw degree up to 90° . The images are diverse in terms of pose, expression and illumination. The number of visible points also varies depending on the image, but the location of occluded points are to be predicted as well.

Testing Protocols:

(I)AFLW-PIFA: We follow the protocol used in PIFA [73]. We randomly select $23,386$ images for training and the remaining $1,000$ for testing. We divide the testing images in three groups as done in [73]: $[0^\circ, 30^\circ]$, $[30^\circ, 60^\circ]$ and $[60^\circ, 90^\circ]$ where the number of images in each group are taken to be equal.

(II)AFLW-Full: We also test on the full test set of AFLW of sample size $1,000$.

(III)AFLW-All variants: In the next experiment, to have more rigorous anal-

ysis, we perform the test on all variants of images from (I) above. To create all variants images, we first rotate the full images from (I) at angles of 15°, 30°, 45° and 60°. We do the same with the horizontally flipped version of these images. We then rotate the bounding box coordinates and the key-points also at the same angles and crop the faces. This is done for all the images following the AFLW-PIFA protocol. One important effect of this rotation is that some of the images have smaller faces compared to others due to rotated bounding box. This experiment tests the robustness of the algorithm on faces of different effective scales and orientations.

(IV)AFW: We only use AFW for testing purposes. We follow the protocol as stated in [177]. AFW provides 468 images in total, out of which 329 faces have height and width greater than 150 pixels. We only evaluate on those 329 images following the protocol of [177]. It is to be noted that methods such as PIFA [73] and CCL [176] also exclude images with pose greater than 75 degrees follwing the protocol of TCDCN [171].

(V) Occlusion: We use COFW dataset only for evaluation purposes without fine-tuning. This shows the efficacy of the proposed method on other datasets. COFW face images are annotated with 29 facial landmarks, however we only evaluate on 21 points as in AFLW. We show that even without retraining KEPLER performs comparable to Robust Cascaded Pose Regression(RCPR) [23] which is the baseline method. We show in Figure 3.8 the schema to convert 29 points to 21 points format.

(VI) Real Life Scenario: We use IJB-A dataset to evaluate on images and videos which are taken in challenging situations. We only evaluate against the three points which were manually annotated. The error between the two eye coordinates is normalized the the distance between the nose coordinate and the midpoint of two eye coordinates.

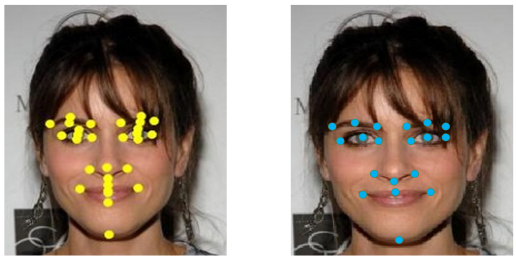


Figure 3.8: Schema to convert COFW 29 point format to AFLW 21 point format.

Evaluation metric: Following most previous works, we obtain the error for each test sample via averaging normalized errors for all annotated landmarks. We demonstrate our results with mean error over all samples, or via Cumulative Error Distribution (CED) curve. For pose, we evaluate on continuous pose predictions as well as their discretized versions rounded to nearest 15°. We report the continuous mean absolute error for the AFLW testset and plot the Cumulative Error Distribution curve for AFW dataset. For the COFW dataset we normalise by the inter-occular distance following the protocol from [23]. The Normalized Mean Error (NME), which is the average of the normalized estimation error of visible landmarks is calculated as follows.

$$NME = \frac{1}{N_t}\sum_{1}^{N_t}\left(\frac{1}{N_f|v_i|}\sum_{j}^{N} v_i^j||p_i(:,j) - g_i(:,j)||_2^2\right) \tag{3.11}$$

where N_f is the normalization factor, which for AFLW and AFW is the ground truth bounding box size calculated as $\sqrt{w_{box} \text{x} h_{box}}$ and for COFW is the inter-occular distance.

All the experiments including training and testing were performed using the Caffe [72] framework and two Nvidia TITAN-X GPUs. Our method can process upto 12-16 frames per second in batch mode.

	AFLW	***AFW***
Method	**NME**	**NME**
TSPM [177]	-	11.09
CDM [2]	12.44	9.13
RCPR [24]	7.85	-
ESR [25]	8.24	-
PIFA [73]	6.8	8.61
3DDFA [178]	5.32	-
LPFA-3D [74]	4.72	7.43
EMRT [175]	4.01	3.55
CCL [176]	5.85	2.45
Rec Enc-Dec [1]	>6	-
FA-3DFR [98]	4.49	-
Tree CNN [87]	3.93	3.28
3D STN [15]	4.23	-
KEPLER	**2.98**	3.01

Table 3.1: Comparison of KEPLER with other state of the art methods. NME stands for normalized mean error. For AFLW, the numbers for other methods are taken from respective papers following the PIFA protocol. For AFW, the numbers are taken from respective works published following the protocol of [177].

3.4.2 Results

Table 3.1 compares the performance of KEPLER compared to other existing methods. Table 3.3 summarises the performance of KEPLER under different protocols of AFLW testset. Table 3.4 shows the mean error in degrees, in estimating the 3D pose

Method	COFW
FPLL [177]	14.40
ESR [25]	11.20
FLD [155]	5.18
RCPR [23]	8.5
KEPLER	8.8

Table 3.2: Performance comparison of the proposed method on COFW dataset. It is to be noted that NME in FPLL, ESR, FLD and RCPR (trained on COFW) is calculated over 29 points, which is calculated for 21 points in KEPLER. It can be observed that the performance of KEPLER is comparable to RCPR without finetuning on the training set of COFW.

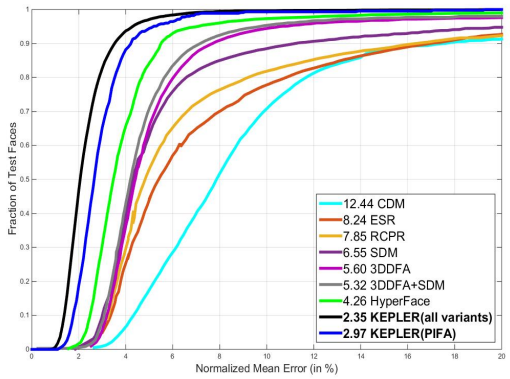


Figure 3.9: Cumulative error distribution curves for landmark localization on the AFLW dataset. The numbers in the legend are the average normalized mean error normalized by the face size.

of a face image. Table 3.2 compares the performance of KEPLER on COFW testset. It can be observed that even without finetuning KEPLER performs comparable to RCPR. Figures 3.9 and 3.10 show the cumulative error distribution in predicting keypoints on the AFLW and AFW test sets. Figure 3.11 shows the cumulative error distribution in pose estimation on AFW. Figures 3.12 and 3.13 shows the cumulative error distribution curves for the COFW and IJB-A datasets.

Comparison with CCL [176]: It is clear from the tables that KEPLER outperforms all state of the art methods on the AFLW dataset. It also outperforms all state of the art methods except CCL [176] on the AFW datatset. Visual inspec-

tion of our results suggests that KEPLER is a little farther from ground truth on invisible points. We note that CCL [176] manually annotates the AFLW dataset with 19 landmarks along with the invisible landmarks, leaving the earpoints. In our experiments we prefer to use the dataset as provided by AFLW [82], although we believe that CCL-kind of reannotation may boost the performance(since during AFW evaluation the location of occluded points also need to be predicted). In KEPLER there is no loss propagated for the invisible points. We believe that training KEPLER on the revised annotation by [176] would make the prediction of occluded points more precise.

Method	**AFLW-PIFA**	**AFLW-FULL**	**AFLW-Allvariants**	**AFW**
KEPLER	2.98	2.90	2.35	3.01

Table 3.3: Summary of performance on different protocols of AFLW and AFW by KEPLER.

Method	**AFLW**				**AFW**
	Yaw	**Pitch**	**Roll**	**MAE**	**Accuracy**($\leq$ **15°**)
Random Forest [146]	-	-	-	12.26°	83.54%
KEPLER	**6.45°**	**5.85°**	**8.75°**	**6.45°**	**96.67%**

Table 3.4: Comparison of Mean error in 3D pose estimation by KEPLER on AFLW testset. For AFLW [146] only compares mean average error in Yaw. For AFW we we compare the percentage of images for which error is less than 15°.

We also verify our claim that iteration 5 is optional and only required for transferring the algorithm to other datasets with different bounding box annotations. To support our claim we calculate the normalized mean error after iteration 4 for both datasets and compare with the error obtained after iteration 5. The error after iteration 4 for AFLW testset was 0.0369 (which is already lower than all existing

works) and after fifth iteration it was 0.0299, bringing the performance up by 18%. On the other hand the improvement in AFW (whose bounding box annotation is different from AFLW) was close to 60%. The error after iteration 4 on AFW dataset was 0.0757 which decreases to 0.0301 after fifth iteration.

We demonstrate some qualitative results from AFLW and AFW test sets in Figure 3.14 and from COFW and IJB-A datasets in Figures 3.15 and 3.16.

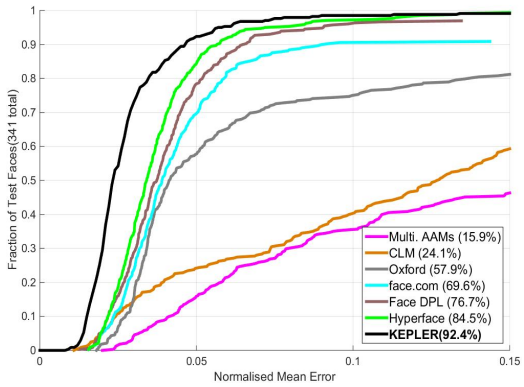


Figure 3.10: Cumulative error distribution curves for landmark localization on the AFW dataset. The numbers in the legend are the fraction of testing faces that have average error below (5%) of the face size.

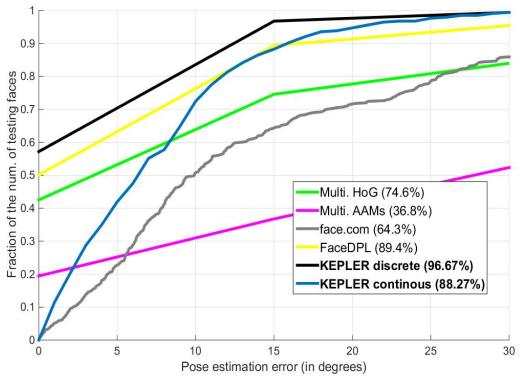


Figure 3.11: Cumulative error distribution curves for pose estimation on AFW dataset. The numbers in the legend are the percentage of faces that are labeled within $\pm 15^{\circ}$ error tolerance

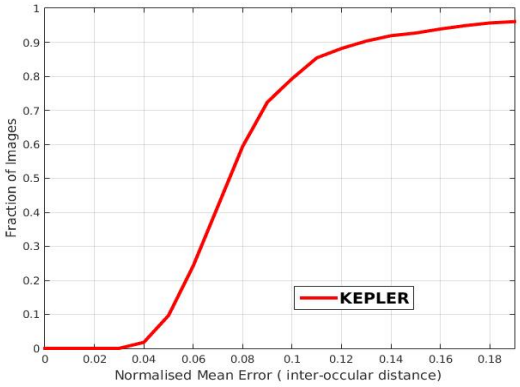


Figure 3.12: Cumulative error distribution curves for landmark localization on the COFW dataset. This is to be noted that the error is calculated over 21 points normalized by inter-occular distance.

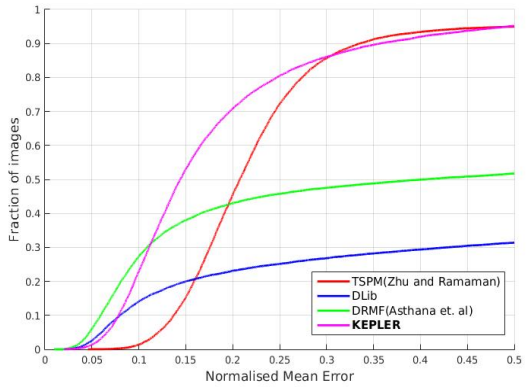


Figure 3.13: Cumulative error distribution curves for landmark localization on the IJBA dataset. The error is calculater for 3 points normalized by the distance between midpoint of eyes and the nose.

3.5 Conclusions

In this work, we showed that by efficiently capturing the structure of face through additional channels, we can obtain precise keypoint localization on unconstrained faces. We proposed a novel *Channeled Inception* deep network which pools features from intermediate layers and combines them in the same manner as the Inception module. We show how cascade regressors can outperform other recently developed works and designed to yield variable number of keypoints. As a byproduct of KEPLER, 3D pose information is also generated which can be used for other tasks such as pose dependent verification methods, 3D model generation and many others. In conclusion, KEPLER demonstrates that by improved initialization and multitask training, cascade regressors outperforms state of the art methods not only in predicting the keypoints but also for head pose estimation. One future avenue for extending this work, can be developing methods in which the gaussians are learned and estimated directly from the image.

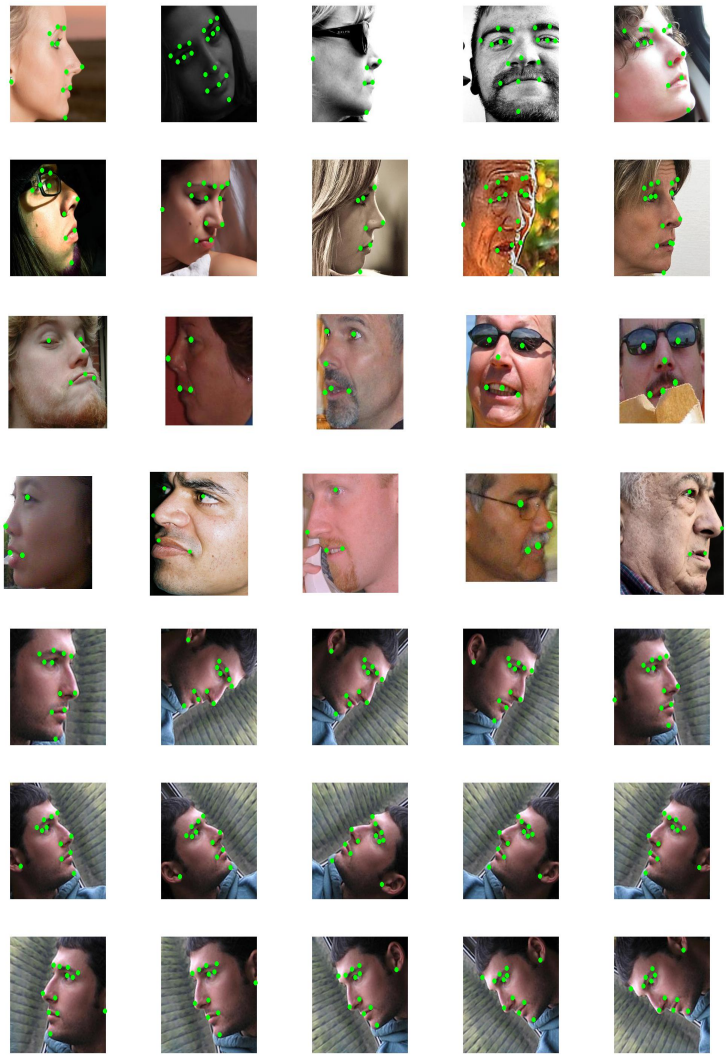


Figure 3.14: Qualitative results of KEPLER after last stage. The green dots represent the final predicted points after last stage. First row are the test samples from AFLW. Second row shows the samples from AFW dataset. The last two rows are the results of KEPLER after last stage from AFLW testset for all variants protocol. The green dots represent the final predicted points after second stage.

Figure 3.15: Qualitative results of KEPLER after last stage on COFW dataset. The green dots represent the final predicted points after last stage.

Figure 3.16: Qualitative results of KEPLER after last stage on IJBA dataset. The green dots represent the final predicted points after last stage.

Chapter 4: Disentangling 3D Pose in A Dendritic CNN for Unconstrained 2D Face Alignment

4.1 Introduction

As shown in [10], accurate face alignment improves the performance of a face verification system, as well as other applications such as 3D face modelling, face animation etc. Currently, face alignment is still dominated by regression-based approaches which yield a fixed number of points. Explicit Shape Regression (ESR) [25] and Supervised Descent Method (SDM) [158] have addressed the problem of face alignment for faces in medium pose. To achieve sub-pixel accuracy on such face images, coarse to fine approaches have also been proposed in the literature [89, 168, 174]. It is evident that such methods perform poorly on face images with extreme pose, expression and lighting mainly because they are dependent on bounding box and mean face shape intializations. On the other hand, Convolutional Neural Networks (CNNs) have achieved breakthroughs in many vision tasks including the task of keypoints estimation [109]. Lately, researchers have used heatmap regression extensively for the task of face alignment and pose estimation using an Encoder-Decoder architecture in the form of Convolution-Deconvolution Networks [32]. Most of the approaches in

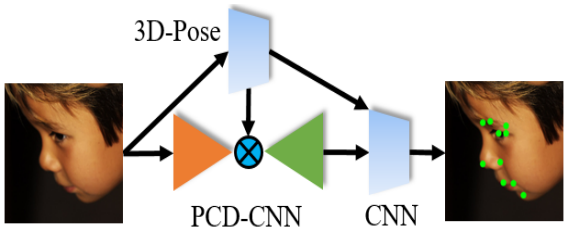


Figure 4.1: (a) A bird's eye view of the proposed method. Dendritic CNN is explicitly conditioned on 3D pose. A generic CNN is used for auxiliary tasks such as fine-grained localization or occlusion detection.

the literature perform heatmap classification followed by regression [11, 17, 18, 21]. In this work, we propose the Pose Conditioned Dendritic Convolution Neural Network (PCD-CNN); which models the dendritic structure of facial landmarks using a single CNN (see Figure 4.1).

Shape constraint: Methods such as ESR [25] and SDM [158] impose the shape constraint by jointly regressing over all the points. Such a shape constraint cannot be applied to a profile face as a consequence of extreme pose leading to a variable number of points. Tree structured part models (TSPM) [177] by Zhu et al. had two major limitations associated with it; namely pre-determined models and slower run-time. With an intent to solve these, we propose a tree structure model in a single Dendritic CNN (PCD-CNN), which is able to capture the shape constraint in a deep learning framework.

Pose: Works such as Hyperface [115] and TCDCN [172] have used 3D pose in a multitask framework and demonstrated that learning pose and keypoints jointly using a deep network improves the performance of both tasks. However, in contrast to multi-tasking approaches, we condition the landmark estimates on the head pose, following a Bayesian formulation and demonstrate the effectiveness of the proposed

approach through extensive experiments. We wish to point out that our primary goal is not to predict the head pose, instead, use 3D head pose to condition the landmark points. This makes our work different from multitask approaches.

Speed-vs-Accuracy: We observe that systems which process images at real time, such as [14,75] have higher error rate as opposed to cascade methods which are accurate but slow. Researchers have proposed many different network architectures like Hourglass [109], Binarized CNN (based on hourglass) [18] in order to achieve accuracy in keypoints estimation. Although, such methods are fully convolutional , they suffer from slower run time as a result of cascaded deep bottleneck modules which perform a large number of FLOPs during test time. The proposed PCD-CNN works at the same scale as the input image and thus reduces the extrapolation errors. PCD-CNN is fully convolutional with fewer parameters and is capable of processing images almost at real time speed (20FPS). Limited generalizability as a consequence of smaller number of parameters is tackled by efficiently training the network using Mask-Softmax loss and difficult sample mining.

Generalizability: Methods for domain-limited face images have been developed, mostly following the cascade regression approach. [24, 156, 167] have been shown to work well for faces under extreme external object occlusion. On the other hand, [92, 116, 142, 144, 145, 174] achieved satisfactory results on the 300W [123] dataset which contains images in medium pose with almost no occlusion. [73,85,176] have demonstrated their effectiveness for extreme pose datasets with a limited number of fiducial points. However, they do not generalize very well to other datasets. We show that by a small increase in the number of parameters, PCD-CNN can be

extended to most of the publicly available datasets including 300W, COFW, AFLW and AFW yielding variable number of points depending on the protocol.

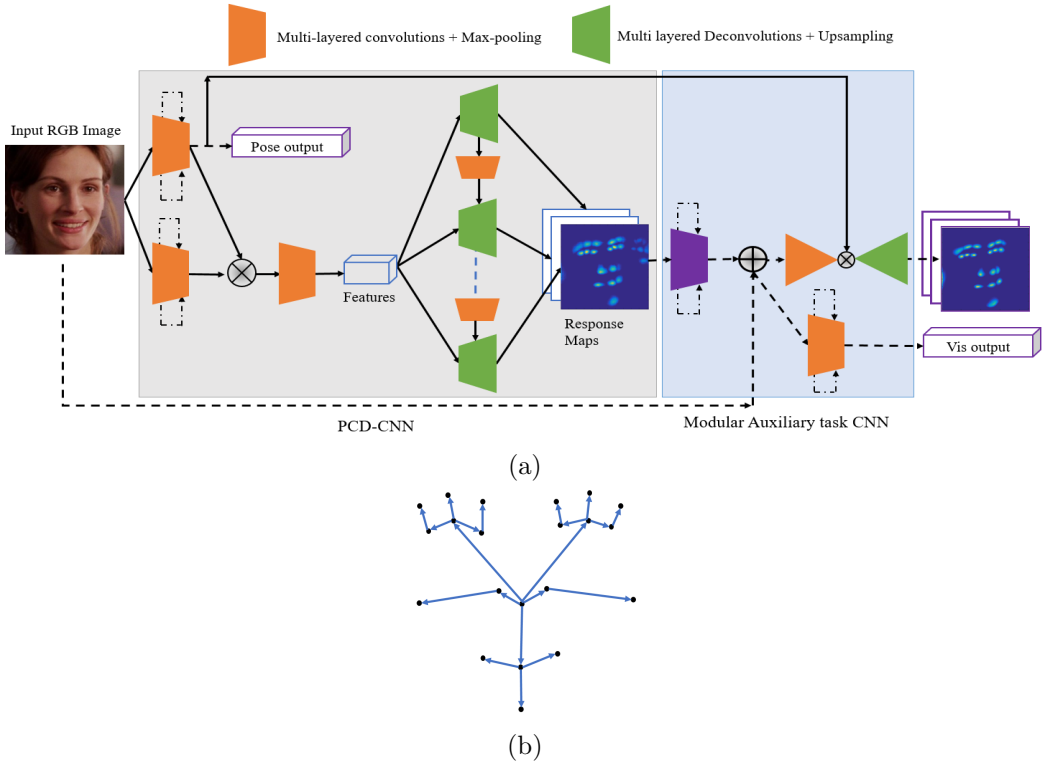


Figure 4.2: (a) Details of the proposed method. The dotted lines on top of convolution layers denote residual connections. The feature maps from the pose model are multiplied element-wise with the feature maps of the keypoint model. The network inside the grey box represents the proposed PCD-CNN, whereas the second network inside the blue box is modular and can be replaced for an auxiliary task. A conv-deconv network for finer localization is used alongside a second regression network for occlusion detection. (b) Proposed dendritic structure of facial landmark points for effective information sharing among landmark points. The nodes of the dendritic structure are the outputs of deconvolutions while the edges between nodes i and j are modeled by convolution functions f_{ij}. For the architecture of deconvolution network refer to Figure 4.3.

To summarize, the main contributions of this work are :

- We propose the Pose Disentangled Dendritic CNN for unconstrained 2D face alignment, where the shape constraint is imposed by the dendritic structure of facial landmarks. The proposed method uses classification followed by classifi-

cation approach as opposed to classification followed by regression. The second auxiliary network is modular and can be designed for fine grained localization or any other auxiliary tasks.

- The proposed method disentangles the head pose using a Bayesian framework and experimentally demonstrates that conditioning on 3D head pose improves the localization performance. The proposed method processes images at real-time speed producing accurate results.
- With a recursive extension, the proposed method can be extended to datasets with arbitrarily different number of points and different auxiliary tasks.
- As a by-product, the network outputs pose estimates of the face image where we achieve close to state-of-the-art result on pose estimation on the AFW dataset. In another experiment, the auxiliary classification network is trained for occlusion detection where we obtain state-of-the-art result for occlusion detection on COFW dataset.

4.2 Prior Work

We briefly review prior work in the area of keypoint localization under the following two categories: Deep Learning-based and Hand crafted features-based methods.

Parametric part-based models such as Active Appearance Models (AAMs) [36] and Constrained Local Models [38] are statistical methods which perform keypoint detection by maximizing the confidence of part locations in a given input image using ***handcrafted features*** such as SIFT and HOG. The tree structure part

model (TSPM) proposed in [177] used deformable part-based model for simultaneous detection, pose estimation and landmark localization of face images modeling the face shape in a mixture of trees model. Later, [9] proposed learning a dictionary of probability response maps followed by linear regression in a Constrained Local Model (CLM) framework. Early cascade regression-based methods such as [8, 25, 134, 142, 144, 158, 174] also used hand crafted features such as SIFT to capture appearance of the face image. The major drawback of regression-based methods is their inability to learn models for unconstrained faces in extreme pose.

Deep learning-based methods have achieved breakthroughs in a variety of vision tasks including landmark localization. One of the earliest works was done in [89, 135] where a cascade of deep models was learnt for fiducial detection. 3DDFA [178] modeled the depth of the face image in a Z-buffer, after which a dense 3D face model was fitted to the image via CNNs. Pose Invariant Face Alignment (PIFA) [73] by Jourabloo et al. predicted the coefficients of 3D to 2D projection matrix via deep cascade regressors. [14] used 3D spatial transformer networks to capture 3D to 2D projection. [69, 76, 99] extended [73] by using CNNs to directly learn the dense 3D coordinates. The proposed method has a dendritic structure which looks at the global appearance of the image while the local interactions are captured by pose conditioned convolutions. PCD-CNN does not assume that all the keypoints are visible and the interactions between keypoints are learned. PCD-CNN is entirely based on 2D images, which captures the 3D information by conditioning on 3D head pose.

Formulating keypoint estimation as the per-pixel labeling task, Hourglass net-

works [109] and Structured feature learning [34] were proposed. Hourglass networks use a stack of 8 very deep hourglass modules and hence, even though based entirely on convolution can process only 8-10 frames per second. [34] implemented message passing between keypoints, however was able to process images at lower resolution due to large number of parameters. PCD-CNN models the dendritic structure in branched deconvolution networks where each network is implemented in Squeezenet [68] fashion and hence has fewer parameters, contributing to real-time operation at full image scale.

In the next few sections, we describe Pose Conditioned Dendritic-CNN in detail and present ablative studies to arrive at the desired architecture.

4.3 Pose Conditioned Dendritic CNN

The task of keypoint detection is to estimate the 2D coordinates of, say N landmark points, given a face image. Observing the effectiveness of deep networks for a variety of vision tasks, we present a single end-to-end trainable deep neural network for landmark localization.

It has been shown in previous works that capturing structural dependencies between different keypoints is important [34]. THis work derives its motivation from the work by Zhu and Ramanan [177] where every keypoint was modeled as a part and mixture of trees was used to select the best fitting model. Modeling such structural interactions between keypoints pose a great challenge in a deep learning framework as the invisible points are not annotated.

Conditioning on 3D pose: Keypoints are susceptible to variations in external factors such as emotion, occlusion and intrinsic face shape. On the other hand, 3D pose is fairly stable to them and can be estimated directly from 2D image [85]. Reasonably accurate 2D keypoint coordinates can be also inferred given 3D pose and a generic 3D model of a human face. However, the converse problem of estimating 3D pose from 2D keypoints is ill posed. Therefore, we make use of the probabilistic formulation over the variables including the image $\mathbf{I} \in \mathbb{R}^{w \times h \times 3}$ of height h and width w, $3D$ head pose denoted by $\mathbf{P} \in \mathbb{R}^3$, $2D$ keypoints $\mathbf{C} \in \mathbb{R}^{N \times 2}$, where N is the number of keypoints. Following the natural hierarchy between the two tasks, the joint and the conditional probabilities can be written as:

$$p(\mathbf{C}, \mathbf{P}, \mathbf{I}) = p(\mathbf{C}|\mathbf{P}, \mathbf{I})p(\mathbf{P}|\mathbf{I})p(\mathbf{I}) \tag{4.1}$$

$$\begin{aligned} p(\mathbf{C}, \mathbf{P}|\mathbf{I}) &= \frac{p(\mathbf{C}, \mathbf{P}, \mathbf{I})}{p(\mathbf{I})} \\ &= \underbrace{p(\mathbf{P}|\mathbf{I})}_{\text{CNN}} \cdot \underbrace{p(\mathbf{C}|\mathbf{P}, \mathbf{I})}_{\text{PCD-CNN}} \end{aligned} \tag{4.2}$$

We implement the first factor with an image-based CNN learned to predict the 3D pose of the face image. The second factor is implemented through a ConvNet and multiple DeconvNets arranged in a dendritic structure. The convolution network maps the image to lower dimension, after which the outputs of several deconvolution networks are stacked to form the keypoint-heatmap. The models are tied together by element-wise product (as (4.1) and (4.2)) to condition the measurement

of 2D coordinates on 3D pose. We choose element-wise product as the operation to condition on the head pose as keypoint heatmaps can be interpreted as probability distribution over the keypoints. The visibility of each keypoint is learnt implicitly as the invisible points are labeled as background.

Multi-tasking-vs-Conditioning: In a multi-tasking method such as [85], several tasks are learnt synergetically and backpropagation impacts all the tasks. On the other hand, in the proposed PCD-CNN, the error gradients backpropagated from keypoint network affect both, keypoint network and pose network; however, the pose network affects the keypoint network only during the forward pass. In other words, multi-tasking approaches try to model the joint distribution $p(\mathbf{C}, \mathbf{P}|\mathbf{I})$, whereas the proposed approach explicitly models the decomposed form $p(\mathbf{P}|\mathbf{I})p(\mathbf{C}|\mathbf{P}, \mathbf{I})$ by learning the individual factors.

Proposed Pose Conditioned Dendritic CNN : We propose the dendritic structure of facial landmarks as shown in figure 4.7b where the nose tip is assumed to be the root node. Such a structure is feasible even in faces with extreme pose. Following this, the keypoint estimation network is modeled with a single CNN in a tree structure composed of convolution and deconvolution layers. The pairwise relationships between different keypoints are modeled via specialized functions, $f_{i,j}$, which are implemented through convolutions and are analogous to the spring weights in the spring-weight model of Deformable Part Models [49]. A low confidence of a particular keypoint is reinforced when the response of $f_{i,j}$ corresponding to the adjacent node is added. With experimental justifications we show that such a deformable tree model outperforms the recently published works [14, 75, 76, 99] which use 3D models

and 3D spatial transformer networks to supplement keypoint detection models. Figure 4.2 shows the overall architecture of the proposed PCD-CNN and the proposed dendritic structure of the facial landmarks.

Instead of going deeper or wider [18, 109] with deep networks, we base our work on the Squeezenet-11 [68] architecture, attributing to its capability to maintain performance with fewer parameters. We use two Squeezenet-11 networks; one for pose and other for keypoints, named as -PoseNet and KeypointNet respectively. Convolutions are performed on the $pool_8$ activation maps of the PoseNet, the response of which is then multiplied element-wise to the response maps of $pool_8$ layers of the KeypointNet. Each convolution layer is followed by ReLU non-linearity and batch normalization. In table 4.10, we show that keypoint localization error reduces when conditioned on 3D head pose.

The design of deconvolution network is non-trivial. To maintain the same property as of SqueezeNet, we first upsample the feature maps using parametrized strided convolutions and then squeeze the output features maps using 1x1 convolutions. We call this network as Squeezenet-DeconvNet. Figure 4.3 shows the detailed architecture of the Squeezenet-DeconvNet. Since, each keypoint in the proposed network is modeled by a separate Squeezenet-DeconvNet, it alleviates the need for large number of deconvolution parameters (256 and 512 3×3 in Hourglass networks). In fact, in the practical version of PCD-CNN, there are only 32 and 16 deconvolution filters which results in the design of networks, which are small enough to fit in a single GPU. The design of networks with fewer filters is motivated by real-time processing consideration. With experiments we show that disentangling the pose

Method	Normalised Error
Without pose conditioning	3.45
With pose conditioning	2.85

Table 4.1: Root mean square error normalized by bounding box size, calculated on the AFLW validation set following the PIFA protocol. The proposed PCD-CNN when conditioned on pose yields better performance for the task of keypoint localization.

Method	Normalised Error
Classification+Regression	3.93
Classification+Classification	3.09

Table 4.2: Mean square error normalized by bounding box size, calculated on the AFLW validation set following the PIFA protocol. This table shows that PCD-CNN when followed by another classification stage results in lower localization error compared to classification followed by regression. Note that conditioning on pose is not used in both the cases above for fair comparison.

by conditioning on it, reinforces the learning of the proposed PCD-CNN with fewer parameters (Table 4.10).

In order to obtain fine grained localization results, we concatenate to the input data, a learned function of the predicted probabilities (represented as purple box in Figure 4.7a) and pass them through the second Squeezenet based conv-deconv network. This function is modeled by a residual unit with 1×1 and 3×3 filters, which are learned end-to-end with the second classification network (while keeping the weights PCD-CNN frozen). For experimental purposes, we replace the second conv-deconv by another regression network designed along the lines of GoogleNet [137]. Table 4.2 shows a comparison between two stage classification approach versus classifcation followed by regression approaches.

One of the goals of this work is to generalize the facial landmark detection to other datasets in order to broaden its applicability. A trivial extension would be

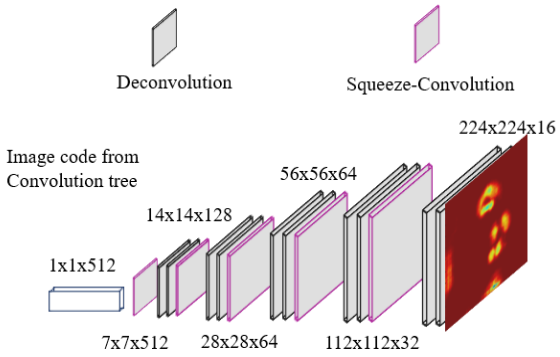


Figure 4.3: Detailed description of a single Squeezenet-DeconvNet network. Note the fewer number of deconvolution filters. Each deconvolution network is identical to the one shown above.

to increase the number of deconvolution branches, which however is infeasible due to limited GPU memory. With a non-trivial extension, PCD-CNN can be extended to yield more landmark points arranged in different configurations. In figure 4.9 we show the proposed tree structures for COFW and 300W datasets with 29 and 68 landmark points respectively. Keeping the basic dendritic structure intact, first the number of output response maps in the last deconvolution layer are increased and then network slicing is performed to produce the desired number of keypoints. For instance, the output of the deconvolution network for eye-center is sliced to produce four outputs as required by the 300W dataset. Depending on the dataset, the second network can be replaced to perform auxiliary tasks resulting in a modular architecture; for instance in the case of COFW dataset we replace the second conv-deconv network with another Squeezenet network to detect occlusion. We direct the readers to the supplementary material for more details on network surgery and a magnified view of figures 4.7b and 4.9.

Each branch of PCD-CNN is designed according to the proposed Squeezenet-Deconv networks shown in Figure 4.3. Due to fewer parameters in the Squeezent-

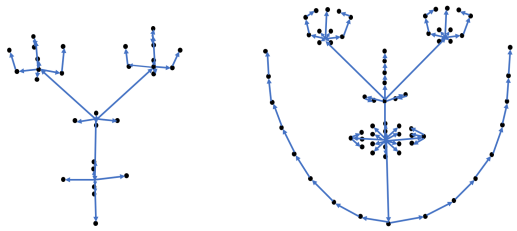


Figure 4.4: The proposed extension of the dendritic structure from Figure 4.2 generalizing to other datasets (COFW and 300W) each with different number of points.

Deconv, we hypothesize limited generalization capacity of the deconvolution network. By means of experiments, we show that effective training methods such as Mask-Softmax and Hard sample mining improves the performance of PCD-CNN by a large margin as a result of better generalization capacity.

Mask-Softmax Loss: To train the network, the localization of fiducial keypoints is formulated as a classification problem. The label for an input image of size $h \times w \times 3$ is a label tensor of same size as the image with $N+1$ channels, where N is the number of keypoints. The first N channels represent the location of each keypoint whereas the last channel represents the background. Each pixel is assigned a class label with invisible points being labeled as background. The objective is to minimize the following loss function:

$$L_0(\mathbf{p}, \mathbf{g}) = \sum_{i=1}^{h} \sum_{j=1}^{w} m(i,j) \sum_{k=1}^{N+1} g_k(i,j) log\left(\frac{e^{p_k(i,j)}}{\sum_l e^{p_l(i,j)}} \right) \tag{4.3}$$

where $k \in \{1, 2 \dots N\}$ is the class index and $g_k(i,j)$ represents the ground truth at location (i,j). $p_l(i,j)$ is the score obtained for location (i,j) after forward pass through the network. Since the number of negative examples is orders of magnitudes larger than the positives, we design a strategic mask $m(i,j)$ which selects all the

Method	Normalised Error
Softmax	4.56
Using Mask-Softmax	2.85

Table 4.3: Root mean square error normalized by bounding box calculated on the AFLW validation set following PIFA protocol. This table indicates the effect of using Mask-softmax over Softmax.

positive pixel samples, and keeps only 50% of the 4-neighborhood pixels and 0.025% of the negative background samples by random selection. During backward pass, the gradients are weighed accordingly. We experimentally show the effect of using Mask-Softmax Loss by training two separate PCD-CNN; with and without the Mask-Softmax Loss; trained under identical training policies(Table 4.3) .

Hard Sample Mining: [77] by Kabkab et al. showed that effective sampling of data improves the classification performance of the network. Following [77], we use an offline hard sample mining procedure to train the proposed PCD-CNN. The histogram of error on the training data is plotted after the network is trained for 10 epochs by random sampling (refer supplementary material). We denote the mode of the distribution as C, and categorize all the training samples producing errors larger than C as hard samples. Next we retrain the proposed PCD-CNN with hard and easy samples, sampled at the respective proportion. This effectively results in retraining the network by reusing the hard samples. Table 4.4 shows that such hard sample mining improves the performance of PCD-CNN (with fewer parameters) by a large margin.

In the next set of experiments, we train PCD-CNN by increasing the number of deconvolution filters to 128 and 64 in each deconvolution network. We follow

Method	Normalised Error
Without Hard Mining	2.85
With Hard Mining	2.49

Table 4.4: Root mean square error normalized by bounding box calculated on the AFLW validation set following PIFA protocol. This table depicts the effect of offline hard sample mining.

Method	Normalised Error
Less Filters+Hard Mining	2.49
More Filters+Hard Mining	2.40

Table 4.5: Root mean square error normalized by bounding box calculated on the AFLW validation set following PIFA protocol. This table shows the effect of offline hard-mining and quadrupling the number of deconvolution filters.

the same strategy of Mask-Softmax and hard sample mining to train this network. Unsurprisingly, we see an improvement in performance for the task of keypoint localization (Table 4.5), although, increasing the number of deconvolution filters leads to slower run time of 11FPS as opposed to 20FPS.

4.4 Magnified version of the Tree

One expects to receive information from all other keypoints in order to optimize the features at a specific keypoint. However, this has two drawbacks: First, to model the interaction between keypoints lying far away such as 'eye corner' and 'chin', convolution kernels with larger size have to be introduced. This leads to increase in the number of parameters. Secondly, relationships between some keypoints are unstable, such as 'left eye corner' and 'right eye corner'. In a profile face image one of the points may not be visible and passing information between those two keypoints may lead to erroneous results. Hence, convolution kernels are learned at the size of

14×14 which ensures keypoints which are closer and have stable relationships to be connected together.

We also describe the process of extending the proposed dendritic structure of facial landmarks to other datasets with variable number of landmark points. Figure 4.9a shows the tree structure of the 21 landmark points compatible with the AFLW dataset. In figure 4.9b and 4.9c the number of points is increased to 29 and 68 respectively compatible with COFW and 300W datasets. We wish to keep the structure of the facial landmarks intact while increasing the number of landmark points. For this, we make use of the network surgery. First, the number of deconvolution filters in the penultimate and ultimate deconvolution layers is increased to 128 and 64 respectively. Next 1×1 convolutions are used to obtain desire number of outputs, which is then sliced and concatenated in order for loss computation. For instance, eye center points is split into 4 landmark points in the case of COFW and 300W datasets, and ear corner points are dropped. An advantage of network surgery is that, it leads to yielding a variable number of landmark points with minimal increase in parameters while keeping the face structure intact.

4.5 Experiments

We select four different datasets with different characteristics to train and evaluate the proposed two stage PCD-CNN.

AFLW [82]and **AFW** [177] are two *difficult* datatsets which comprises of images in extreme pose, expression and occlusion. AFLW consists of $24,386$ in-the-

wild faces (obtained from *Flickr*) with head pose ranging from 0° to 120° for yaw and upto 90° for pitch and roll. AFLW provides at most 21 points for each face. It excludes coordinates for invisible landmarks and in our method such invisible points are labelled as background. For AFLW we follow the PIFA protocol; i.e. the test set is divided into three groups corresponding to three pose groups with equal number of images in each group.

AFW which is a popular benchmark for the evaluation of face alignment algorithms, consisting of 468 in-the-wild faces (also obtained from Flickr) with yaw up to 90°. The images are diverse in terms of pose, expression and illumination and was considered the most difficult publicly available dataset, until AFLW. The number of visible points varies depending on the pose and occlusion with a maximum of 6 points per face image. We use AFW only for evaluation purposes.

A *medium* pose dataset from the popular **300W** face alignment competition [123]. The dataset consists of re-annotated five existing datasets with 68 landmarks: iBug, LFPW, AFW, HELEN and XM2VTS. We follow the work [174] to use $3,148$ images for training and 689 images for testing. The testing dataset is split into three parts: common subset (554 images), challenging subset (135 images) and the full set (689 images).

Another dataset showing extreme cases of external and internal object *occlusion*; **COFW** [155]. COFW is the most challenging dataset that is designed to depict faces in real-world conditions with partial occlusions [24]. The face images show large variations in shape and occlusions due to differences in pose, expression, hairstyle, use of accessories or interactions with other objects. All 1,007 images were

annotated using the same 29 landmarks as in the LFPW dataset, with their individual visibilities. The training set includes 845 LFPW faces + 500 COFW faces, that is 1,345 images in total. The remaining 507 COFW faces are used for testing.

Evaluation Metric: Following most previous works, we obtain the error for each test sample via averaging normalized errors for all annotated landmarks. We illustrate our results with mean error over all samples, or via Cumulative Error Distribution (CED) curve. For AFLW and AFW, the obtained error is normalized by the ground truth bounding box size over all visible points whereas for 300W and COFW, error is normalized by the inter-occular distance. Wherever applicable NME stands for Normalized Mean Error.

Training: The PCD-CNN was first trained using the AFLW training set which was augmented by random cropping, flipping and rotation. The network was trained for 10 epochs where the learning rate starting from 0.01 was dropped every 3 epochs. Keeping the weights of PCD-CNN fixed, the auxiliary network for fine grained classifcation was trained for another 10 epochs using the hard mining strategy explained in section 4.3. PoseNet was kept frozen while training the network for COFW and 300W datasets. All the experiments including training and testing were performed using the Caffe [72] framework and Nvidia TITAN-X GPUs and p6000 GPUs. Being a non-iterative and single shot keypoint prediction method, our method is fast and can process **20** frames per second on 1 GPU only in batch mode.

4.6 Training Details

KeypointNet and PoseNet described in section 3 are designed based on the SqueezeNet architecture, attributing its lower parameter count. The proposed PCD-CNN was first trained using AFLW training set, where Mask-Softmax is used for keypoints and Euclidean Loss for 3D pose estimation. Starting from the learning rate of 0.001, the network was trained for 10 epochs with momentum set to 0.95. The learning rate was dropped by a factor of 10 every 3 epochs. While training PCD-CNN for COFW and 300W datasets, the convolution branch was initialized with the previously trained network, whereas the deconvolution branches were trained from scratch. Since, COFW and 300W datasets does not provide 3D pose ground truth, we leverage the previously trained PoseNet and freeze its weights.

4.6.1 Effect of Pose Disentaglement

Next, we also perform an experiment to observe the effect of 3D pose conditioning on the second auxiliary network designed for fine grained localization. Table 4.10 shows the effect of disentangling pose by conditioning, when the auxiliary conv-deconv network does not receive information from the PoseNet.

4.6.2 Improvement in localization by augmentation during testing

For a fair comparison with the previous state-of-the-art methods we did not perform augmentation during testing. In the next set of experiments along with the test image, we also pass the flipped version of it and the final output is taken as the mean

	AFLW	AFW
Method	**NME**	**NME**
TSPM [177]	-	11.09
CDM [2]	12.44	9.13
RCPR [24]	7.85	-
ESR [25]	8.24	-
PIFA [73]	6.8	9.42
3DDFA [178]	5.32	-
LPFA-3D [74]	4.72	7.43
EMRT [175]	4.01	3.55
Hyperface [115]	4.26	-
Rec Enc-Dec [1]	>6	-
PIFAS [76]	4.45	6.27
FRTFA [14]	4.23	-
CALE [21]	2.63	-
KEPLER [85]	2.98	3.01
Binary-CNN [18]	2.85	-
PCD-CNN(Fast)	**2.85**	**2.80**
PCD-CNN(C+C)	**2.49**	**2.52**
PCD-CNN(Best: C+C+more filters)	**2.40**	**2.47**

Table 4.6: Comparison of the proposed method with other state of the art methods. C+C stands for classification+classification. For AFLW, numbers for other methods are taken from respective papers following the PIFA protocol. For AFW, the numbers are taken from respective published works following the protocol of [177].

of the two outputs. With experimentation we observe that data augmentation while testing also improves the localization performance. This does not incur any increase in run-time as the inputs can be passed through the network in batch mode, keeping the runtime still at 20FPS. Table 4.11 shows the effects of data augmentation during testing.

4.6.3 Training PCD-CNN for COFW

This section covers the details of training for the COFW dataset. The PCD-CNN network was trained using the Mask Softmax and hard negative mining. The second

Method	[0,30]	[30,60]	[60,90]	Mean
HyperFace [115]	3.93	4.14	4.71	4.26
AIO [114]	2.84	2.94	3.09	2.96
Binary-CNN [18]	2.77	2.86	2.90	2.85
PCD-CNN(C+C)	**2.33**	**2.60**	**2.64**	**2.49**

Table 4.7: Comparison of the proposed method with other state of the art methods on AFLW-PIFA test set, categorized by absolute yaw angles. The numbers represent the normalized mean error.

Method	Common	Challenge	Full
RCPR [24]	6.18	17.26	8.35
SDM [158]	5.57	15.40	7.52
ESR [25]	5.28	17.00	7.58
CFAN [168]	5.50	16.78	7.69
LBF [116]	4.95	11.98	6.32
CFSS [174]	4.73	9.98	5.76
TCDCN [172]	4.80	8.60	5.54
DDN [165]	-	-	5.59
MDM [142]	4.83	10.14	5.88
TSR [103]	4.36	**7.56**	4.99
PCD-CNN	**3.67**	7.62	**4.44**

Table 4.8: Comparison of the proposed method with other state-of-the-art methods on 300W dataset. The NME for comparison are taken from the Table 3 of [103].

auxiliary network was trained for the task of occlusion detection. According to the released details about the COFW dataset, around 23% of the landmark points are invisible. Hence, to tackle the class imbalance problem between the visible and invisible points the following loss function was used.

$$L(\mathbf{p}, \mathbf{g}) = \sum_{i=1}^{29} (0.23 * \mathbb{1}_{g_i^{vis}=1} + 0.77 * \mathbb{1}_{g_i^{vis}=0})(p_i^{vis} - g_i^{vis})^2 \tag{4.4}$$

where $\mathbf{p}, \mathbf{g}$ are the vector of predicted and ground-truth visibilities. p_i^{vis} and g_i^{vis} are the values of the individual elements in the vectors of visibilities. The weighted loss function also balances the gradients back-propagated while loss calculation.

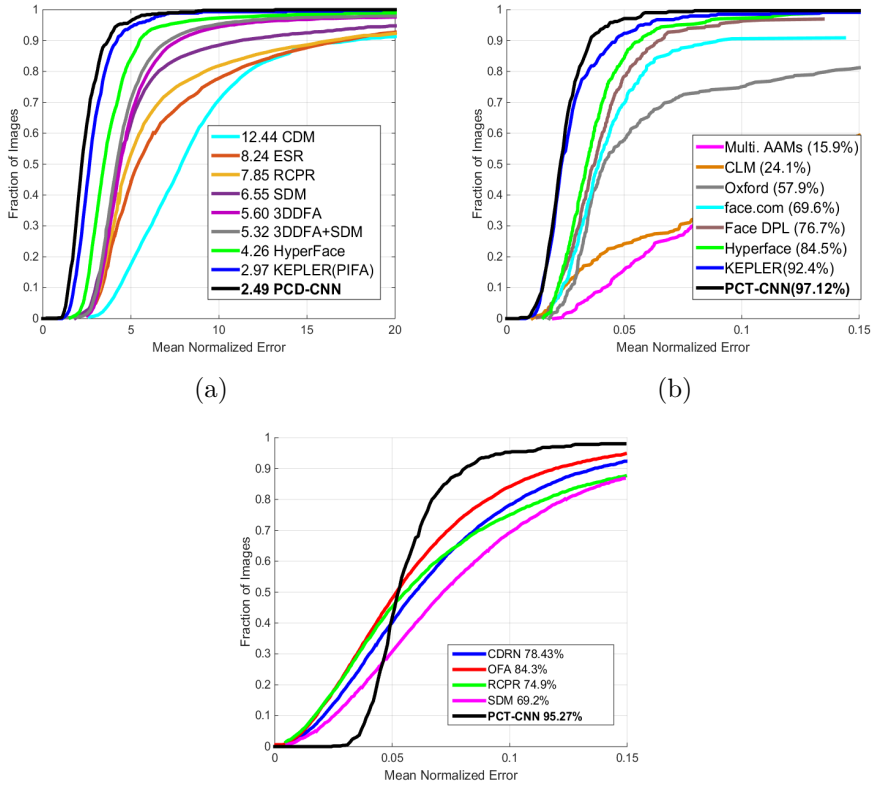


Figure 4.5: Cumulative error distribution curves for landmark localization on AFLW, AFW and COFW dataset respectively. (a) Numbers in the legend represents mean error normalized by the face size. (b) Numbers in the legend are the fraction of testing faces that have average normalized error below 5%. (c) The numbers in the legend are the fraction of testing faces that have average normalized error below 10%.

Figure 4.6 shows the failure rate and error rate on the COFW dataset. The failure rate on the COFW dataset drops to 4.53% bringing down the error rate to 6.02. When testing with the augmented images the error rate further drops to 5.77 bringing it closer to human performance 5.6. Figure 4.8a shows the precision recall curve for the task of occlusion detection on the COFW dataset. PCD-CNN achieves a significantly higher recall of 44.7% at the precision of 80% as opposed to RCPR's [24] 38.2%.

Method	NME	Failure Rate
RCPR [24]	8.5	20%
OFA [167]	6.46	-
HPM [54]	8.48	6.99%
ERCLM [16]	6.49	6.3%
RPP [160]	7.52	16.2%
Human [24]	5.6	0%
PCD-CNN	**6.02**	**4.53%**

Table 4.9: Comparison of the proposed method with other state of the art methods on COFW dataset.

Method	NME
PCD-CNN + Auxiliary Network	2.99
PCD-CNN + Pose Conditioned Auxiliary Network	2.49

Table 4.10: Mean square error normalized by bounding box calculated on AFLW test set following PIFA protocol. When PCD-CNN and fine-grained localization network both are conditioned on pose yields lower error rate.

4.7 Hard mining

Figure 4.7 shows the distribution of average normalized error on the training sets of AFLW and COFW datasets. The error distributions were obtained upon evaluating the PCD-CNN network on the training set, after it is trained with the whole dataset for 10 epochs. The dataset is partitioned into hard and easy samples after choosing the mode of the distribution as the threshold. Next, the network is trained again,

Dataset	Pre-Aug	Post-Aug
AFLW-PIFA (PCD-CNN-Fast)	2.85	**2.81**
AFW (PCD-CNN-Fast)	2.80	**2.66**
AFLW-PIFA (PCD-CNN-C+C)	2.49	**2.40**
AFW (PCD-CNN-C+C)	2.52	**2.36**
COFW (PCD-CNN-Fast)	6.02	**5.77**
300W-Challenge (PCD-CNN-Fast)	7.62	**7.17**

Table 4.11: NME on different datasets Pre-Augmentation and Post-Augmentation during testing.

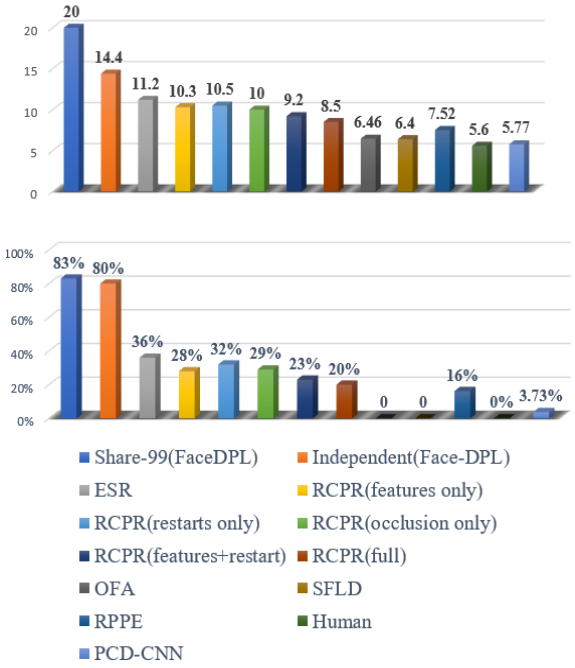
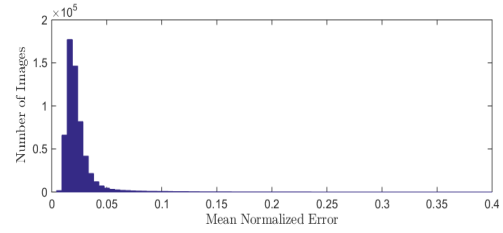
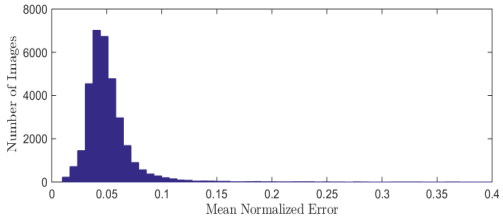


Figure 4.6: Comparison of NME and failure rate over visible landmarks out of 29 landmarks from the COFW dataset.



(a)



(b)

Figure 4.7: Histogram of error, when evaluated on the training set of (a) AFLW (b) COFW.

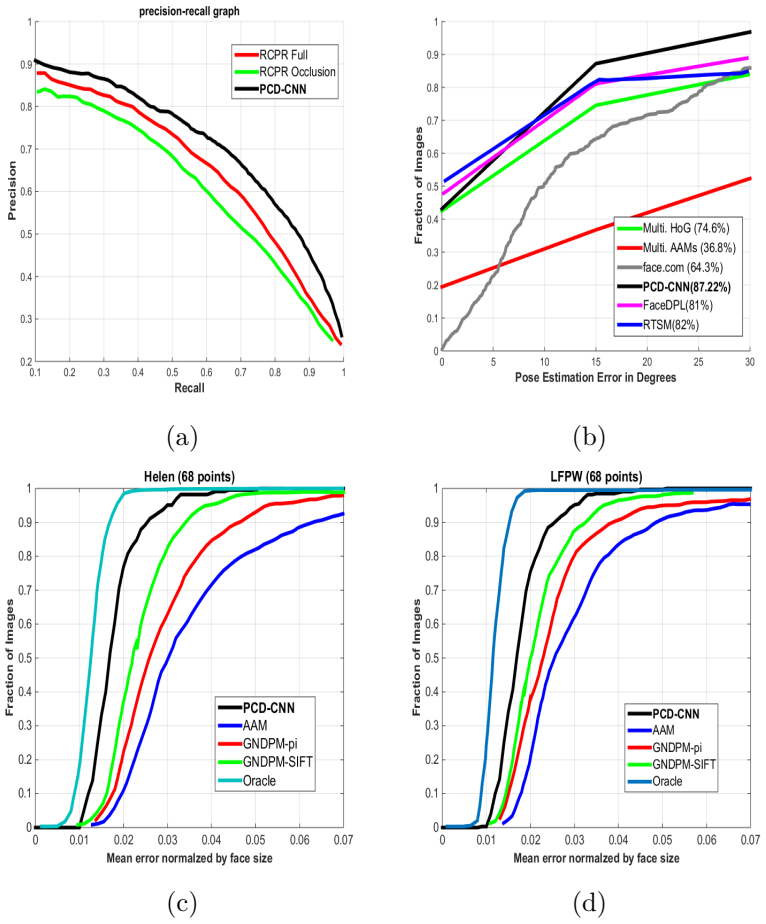


Figure 4.8: (a) Precision Recall for the occlusion detection on the COFW dataset. (b)Cumulative error distribution curves for pose estimation on AFW dataset. The numbers in the legend are the percentage of faces that are labeled within $\pm 15°$ error tolerance. Cumulative Error Distribution curve for (c) Helen (d) LFPW, when the average error is normalized by the bounding box size.

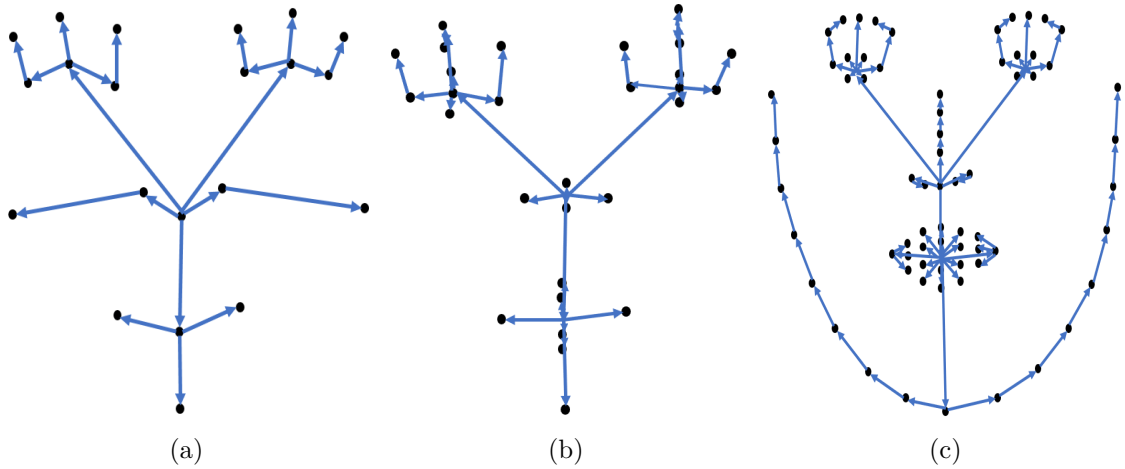


Figure 4.9: The proposed extension of the dendritic structure from Figure 1, generalizing to other datasets with variable number of points.

by sampling equal number of images from both groups, which results in an effective reuse of the hard examples.

4.8 More results on AFLW, AFW, LFPW and HELEN

In this section, we show some more results obtained by the PCD-CNN on AFW, LFPW and Helen datasets. Figure 4.8b shows the cumulative error distribution curves for the prediction of face pose on AFW dataset. We observe that even though the primary objective of PCD-CNN is not pose prediction, it achieves state-of-the-art results when compared to recently published works Face-DPL [177],RTSM [67].

Figures 4.8c and 4.8d show the cumulative error distribution curve on LFPW and Helen datasets, when the average error is normalized by face size. PCD-CNN achieves significant improvement over the recent work of GNDPM [144].

Figure 4.11 shows some of the difficult test samples from AFLW, AFW, COFW and IBUG datasets respectively.

4.8.1 Results

Table 4.6 compares the performance of proposed method over other existing methods on AFLW-PIFA and AFW dataset. Table 4.7 compares the performance on AFLW-PIFA with respect to each pose group. Tables 6.6 and 4.9 compares the mean normalized error on the 300W and COFW datasets respectively. It is clear from the tables that while the proposed PCD-CNN performs comparable to previous state-of-the-art method [18], the two stage PCD-CNN outperforms the state-of-the-art

Figure 4.10: Qualitative results generated from the proposed method. The green dots represent the predicted points. Every two show randomly selected samples from AFLW, AFW, COFW, and 300W respectively with all the visible predicted points.

methods on all three datasets: AFLW, AFW and COFW by large margins. It is not surprising that increasing the number of deconvolution filters improves the performance on all the datasets. Figures 4.5a, 4.5b and 4.5c show the cumulative error distribution for landmark localization in AFLW, AFW and COFW test sets. From the plots, we observe that the proposed PCD-CNN leads to a significant increase in the percentage of images with mean normalized error less than 5%. On AFW, fraction of images having an error of less than 15° for pose estimation is 87.22% compared to 82% in the recent work [67]. On COFW dataset, the NME reduces to 6.02 (close human performance of 5.6) bringing down the failure rate to 4.53%. PCD-CNN achieves a higher recall of 44.7% at the precision of 80% as opposed to RCPR's [24] 38.2%. (refer to the supplementary material for more results.)

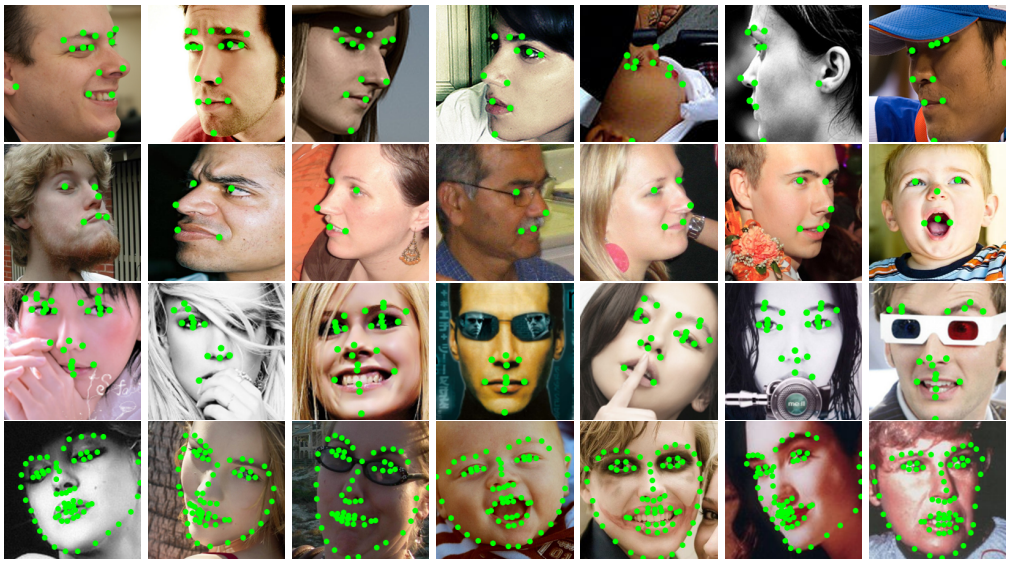


Figure 4.11: Qualitative results generated from the proposed method. The green dots represent the predicted points. Each row shows some of the difficult samples from AFLW, AFW, COFW, and 300W respectively with all the visible predicted points.

Based on our experiments, we observe that two major factors are responsible for achieving state-of-the-art results on the task of face alignment. First, the choices made during the design of PCD-CNN and efficient training; and secondly, disentangling of pose by conditioning on it. With the assistance of above two factors PCD-CNN is able to effectively localize landmark points on unconstrained faces directly from 2D images without using 3D morphable models. Figure 4.11 shows some of the difficult images and the predicted visible keypoints on the four datasets. We also achieve state of the art results on the performance of auxiliary tasks, such as pose estimation on AFW and occlusion prediction on COFW dataset.

4.9 Conclusions

In this work, we present a dendritic CNN which processes images at full scale looking at the images globally and capturing local interactions through convolutions. We also demonstrate that disentangling pose by conditioning on it can influence the localization of landmark points by reducing the mean pixel error by a large margin. We show that due to effective design choices made, the proposed model is not limited to yield a fixed number of points and can be extended to other datasets with different protocols. With the help of ablative studies, impact of effective training of the convolutional network by using sampling strategies such as Mask-Softmax and hard instance sampling is shown. Using smaller and fewer convolution filters, the proposed network is able to process images close to real-time and can be deployed in a real life scenario. The proposed method can be easily extended to 3D face

alignment and human pose estimation tasks, which we plan to pursue in the future.

Chapter 5: A Cascaded Convolutional Neural Network for Age Estimation of Unconstrained Faces

5.1 Introduction

Face analysis is an active research topic in computer vision with applications in surveillance, human-computer interaction, access control, and security. In this work, we focus on apparent age estimation. Traditionally, the problem is tackled through pure classification or regression approaches. In this chapter, we present a cascaded approach which incorporates the advantages of both classification and regression approaches. Given an input image, we first apply the age group classification algorithm to obtain a rough estimate and then perform age group specific regression to obtain an accurate age estimate.

Like other facial analysis techniques, age estimation is affected by many intrinsic and extrinsic challenges, such as illumination variation, race, attributes, etc. One may define the age estimation task as a process of automatically labeling face images with the exact age, or the age group (age range) for each individual. It was suggested in [50] to differentiate the problem of age estimation along four concepts:

- Actual age: real age of an individual.

Figure 5.1: Estimated age on sample images from [45]. Our method is able to predict the age in unconstrained images with variations in pose, illumination, age groups, and expressions.

- Appearance age: age information shown on the visual appearance.
- Apparent age: suggested age by human subjects from the visual appearance.
- Estimated age: recognized age by an algorithm from the visual appearance.

The proposed cascaded classification and regression approach for apparent age estimation is based on a deep convolutional neural network. Our method consists of three main stages: (1) a single coarse age classifier, (2) multiple age regressors, and (3) an error correcting stage to correct the mistakes made by the age group classifer. Since the number of samples for apparent age estimation is limited, we exploit a DCNN model pretrained for large-scale face identification task and finetune the model for age group classification and age regression tasks. This strategy is effective since the face recognition model trained on the CASIA-WebFace dataset [162] (*i.e.* it consists of 10,575 subjects and 494,414 images.) encodes rich information reflecting large variations in facial appearances due to aging and variations in pose, expression

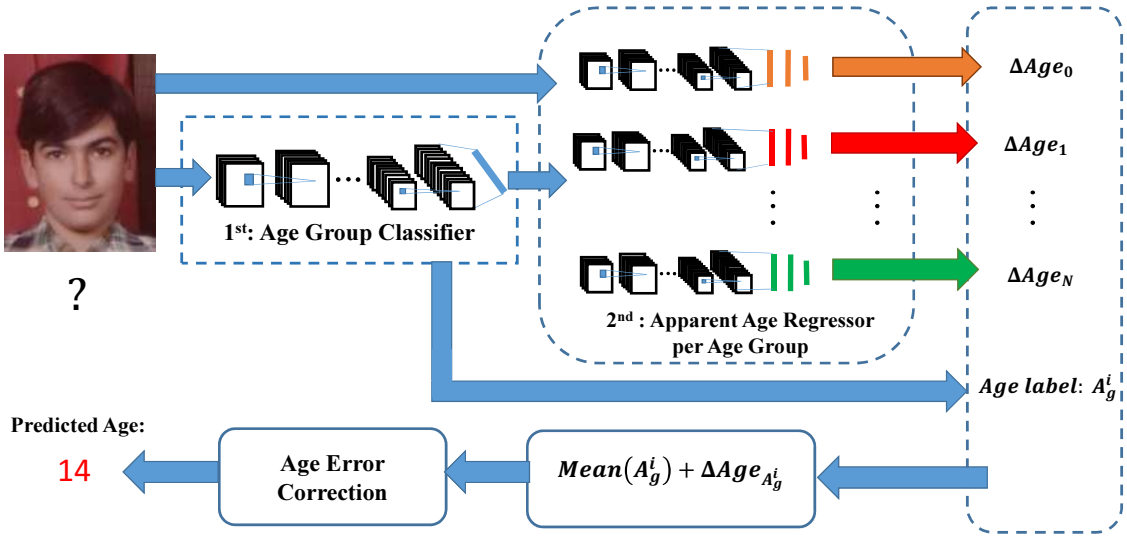


Figure 5.2: An overview of the proposed age cascade apparent age estimator.

and illumination.

The main contribution of this work is to propose the age error correction module which mitigates the common disadvantage of coarse-to-fine approaches. Typically, the errors made at the initial classification stage cannot be recovered by the regressors at the following stage. In this work, we set up the baseline algorithm which is based on the proposed regression algorithm in Section 5.3.6 and study how the coarse-to-fine strategy and the error correction module improve the prediction performance. Figure 5.2 presents an overview of the proposed age estimation method.

The rest of the chapter is organized as follows: Section 5.2 provides a brief overview of the related works. The proposed approach is presented in Section 5.3 with a concrete example. Experimental results are provided in Section 5.4, and Section 5.5 concludes the chapter with a brief summary and discussion.

5.2 Related Work

Most of the earlier age estimation methods have focused on using shape or textural features. These features are then fed to a regression method or a classifier to estimate the apparent age [111, 112, 143, 152].

Holistic approaches usually adopt subspace-based methods, while feature-based approaches typically extract different facial regions and compute anthropometric distances. Geometry-based methods [111, 143] are inspired by studies in neuroscience, which suggest that facial geometry strongly influences age perception [111]. As such, these methods address the age estimation problem by capturing the face geometry, which refers to the location of 2D facial landmarks on images. Recently, Wu *et al.* [152] proposed an age estimation method that presents the facial geometry as points on a Grassmann manifold. To solve the regression problem on the Grassmann manifold, [152] then used the differential geometry of the manifold. However, the Grassmannian manifold-based geometry method suffers from a number of drawbacks. First, it heavily relies on the accuracy of landmark detection step, which might be difficult to obtain in practice. For instance, if an image is taken from a bearded person, then detecting landmarks would become a very challenging task. In addition, different ethnic-groups usually have slightly different face geometry, and to appropriately learn the age model, a large number of samples from different ethnic groups is required.

Unlike the traditional methods discussed, the proposed method is based on DCNN to encode the age information from a given image. Recent advances in deep

learning methods have shown that compact and discriminative image representation can be learned using DCNN from very large datasets [29]. There are various neural-network-based methods, which have been developed for facial age estimation [52, 83, 129] . However, as the number of samples for estimating the apparent age task is limited, (i.e. not enough to properly learn discriminative features, unless a large number of external data is added), the traditional neural network methods often fail to learn an appropriate model.

Thukral *et. al.* [140] proposed a cascaded approach for apparent age estimation based on classifiers using the naive-Bayes approach and a support vector machine (SVM) and regressors using the relevance vector machine (RVM). However, the difference between [140] and the proposed approach is that we leverage the rich information contained in the DCNN model pretrained using a large-scale face dataset for age estimation. Also, the proposed error correction module mitigates the influences of the errors made at initial classification stage.

5.3 Proposed Method

Figure 5.2 shows an overview of our CNN-based cascaded age estimation method. Our approach consists of three main components: (1) age group classifier, (2) age regressor to predict the relative age with respect to each age group mean, and (3) apparent age error correction. Given a face image, we first apply the age group classifier to get a rough estimate of the age range from the image. Then, we choose the corresponding age regressor based on the classification results to predict the

relative age with respect to the predicted group mean and combine them to get the apparent age estimate. Then, we utilize the characteristic of the classification plus regression framework to design an age error correction scheme to correct age classification and regression errors. Finally, the algorithm outputs the final age estimate for the given input image. In what follows next, we will describe each of these component in detail.

5.3.1 Face Preprocessing

In our work, all the face detection and facial landmark detection are handled using the open source library dlib [148] [78]. Three landmark points (the center of the left eye, the center of the right eye, and the nose base) are used to align the detected faces into the canonical coordinate system using the similarity transform.

5.3.2 Deep Face Feature Representation

We use the DCNN model with the architecture similar to the one proposed in [162] which is pretrained for the face-identification task with softmax loss using the CASIA-WebFace dataset [162]. The CASIA-WebFace dataset consists of 10,575 subjects and 494,414 images. The architecture is composed of 10 convolutional layers, 5 pooling layers and 1 fully connected layer. In our work, we use PReLU [62] instead of ReLU as the nonlinear activation function and data augmentation to train the network. The input is a color image of aligned faces of dimension $100 \times 100 \times 3$. The details of this architecture are given in Table 5.1. We do net surgery on this

network (*i.e.*, we cut off the part after pool5 layer.) and use its pretrained weights on the CASIA-WebFace dataset to finetune on the age group dataset and apparent age estimation dataset to perform age group classification and relative age regression with respect to each age group.

5.3.3 Age Group Classifier

Inspired by the Viola and Jones face detection algorithm [148], we quantize the human age into several age groups (*e.g.* 0-7, 8-14, 15-23, etc.) which is an easier problem than directly performing classification or regression for the whole age range which requires a large amount of training data. To train the age group classifier, we remove the original fully connected layer, add the PReLU units and the fully connected layer with 512 outputs and finetune it on the the Images of Groups [51], Adience [43] and FGNet [61] datasets to obtain the DCNN-based age group classifier.

5.3.4 Apparent Age Regressor Per Age Group

To train the age regressor for each age group, we prepare the training data by splitting each training sample into the corresponding age group based on its ground truth age, and then subtract the mean of that group. The regressors are trained in two ways. The first one is to extract the pool5 features and use them to train the regressors with a large batch size. The other is to train the regressor through end-to-end network finetuning but with a smaller batch size. (*i.e.*, Similarly, we keep the part before pool5 layer and add fully connected layers.) Since the pool5 feature

in the face identification task is followed by the fully connected layer with 10,575 output corresponding to the number of subject in the CASIA-WebFace dataset, the pool5 features should contain strong discriminative information from all the face images to classify a large number of subjects in the training data. In addition, we also adopt a novel loss function called, the Gaussian Loss, which takes the a rough age (*i.e.* the age is represented as a mean and a standard derivation instead of the exact age) as input and is robust for apparent age estimation. The role of the new loss function in learning the nonlinear regression method is discussed in Section 5.3.6.

For the pre-training of DCNN face representation model, we use the standard batch size 128 for the training phase. The initial negative slope for PReLU is set to 0.25 as suggested in [62]. The weight decay rates of all the convolutional layers are set to 0, and the weight decay of the final fully connected layer to 5e-4. In addition, the learning rate is set to 1e-2 initially and reduced by half every 100,000 iterations. The momentum is set to 0.9. Finally, we use the snapshot of 1,000,000th iteration as our pretrained model. For the finetuning of the age group classifier, we use the learning rate, 1e-4, for the convolutional layers and 1e-3 for the fully connected layers with 100,000 iterations. For training each age regressor, we first extract all the 320-d feature vectors for each age group and feed them at once into the age regressor network. We train it with 30,000 iterations using the learning rate, 1e-2, and momentum, 0.9. For the end-to-end finetuning of the regressors, we use batch size, 128, with the learning rate, 1e-4, for the convolutional layers and 1e-3 for the fully connected layers. The 120,000th models are used for each age regressor. Data

augmentation is performed by randomly cropping 100 × 100 regions from a 128 × 128 box and horizontally face flipping.

5.3.5 Age Error Correction

In practice, the age group classifier will make errors and these errors significantly affect the final age estimation results for the second stage regressors. To handle these errors, we employ an error correcting approach. When we train the regressor for each age group, we also include the training examples from the neighboring age group. For example, given 3 age groups, (1) 8-14, (2) 15-21, and (3) 22-28, if we want to train the age regressor for the first age group, besides the training samples with ages ranging from 8 to 14 years old, we also add the training samples from its neighboring group (*i.e.*, we added the samples from ±2 groups for the experiments.), that is the second age group. Thus, when the classifier mistakenly assigns the subject to the neighboring age group, the regressor is able to predict a large enough value and correct the error caused by the age group classifier. Furthermore, to take the classifier error into consideration, we also add the misclassified samples to augment the training samples of all the regressors in between the true and wrong groups to increase the chance of correcting the imprecise age estimate so that it is close to the ground truth through our error correction scheme. The detailed step-by-step illustration for the age error correction scheme and other components will be presented in the following subsection. The pseudo code for our age correction approach is given in Algorithm 1.

Algorithm 1 AGE ESTIAMTION ALGORITHM

Require: (a) Input face image, I, (b) maxIter iterations, (c) age group classifier, G_0, and age regressor per age group, A_0, A_1, ..., A_{N-1} where N is the number of age groups and both age group classifier and age regressors are all DCNN-based models.

Ensure: Predicted apparent age, $\hat{a}$.

$g_\ell = G_0(I)$, where g_ℓ is the predicted age group label.
For i = 0 to N-1
    $\Delta a_i = A_i(I)$.
End For
$\hat{a} = \text{mean}(g_\ell) + \Delta a_{g_\ell}$.
// Age estimation error correction
For i = 0 to maxIter - 1
    $\hat{g}_\ell = L(\hat{a})$, where $L(\cdot)$ returns the age group label of $\hat{a}$.
    IF $\hat{g}_\ell = g_\ell$
        Return $\hat{a}$
    ELSE
        $\hat{a} = \text{mean}(\hat{g}_\ell) + \Delta a_{\hat{g}_\ell}$
    End IF
    $g_\ell = \hat{g}_\ell$
End For
Return $\hat{a}$

Name	Type	Filter Size/Stride	#Params
Conv11	convolution	3×3×1 / 1	0.28K
Conv12	convolution	3×3×32 / 1	18K
Pool1	max pooling	2×2 / 2	
Conv21	convolution	3×3×64 / 1	36K
Conv22	convolution	3×3×64 / 1	72K
Pool2	max pooling	2×2 / 2	
Conv31	convolution	3×3×128 / 1	108K
Conv32	convolution	3×3×96 / 1	162K
Pool3	max pooling	2×2 / 2	
Conv41	convolution	3×3×192 / 1	216K
Conv42	convolution	3×3×128 / 1	288K
Pool4	max pooling	2×2 / 2	
Conv51	convolution	3×3×256 / 1	360K
Conv52	convolution	3×3×160 / 1	450K
Pool5	avg pooling	7×7 / 1	
Dropout	dropout (40%)		
Fc6	fully connection	10575	3305K
Cost	softmax		
total			5015K

Table 5.1: The base architecture of DCNN model used in this work [162] to finetune on the age group classification and Δage regression for each age group.

5.3.6 Non-linear Regression

We use a 3-layer neural network to learn the age regressor for each age group. The number of layers is determined experimentally to be 3. The regression is learned by optimizing the Gaussian loss function as follows [45]. The Gaussian loss function is useful since the apparent age labels are usually not exact.

$$L = \frac{1}{N}\sum_{i=1}^{i=N} 1 - e^{-\frac{(\Delta x_i - \mu_i)^2}{2\sigma_i^2}}, \tag{5.1}$$

where L is the average loss for all the training samples, Δx_i is the predicted shift in age from the mean of the corresponding age group. μ_i is the ground truth shift in age and σ_i is the standard deviation in age increment for the i^{th} training sample. The network parameters are trained using the back-propagation algorithm [118] with batch gradient descent. The gradient obtained for the loss function is given by (5.2). This gradient is used for updating the network weights during training using back-propagation.

$$\frac{\partial L}{\partial \Delta x_i} = \frac{1}{N\sigma^2}(\Delta x_i - \mu_i)e^{-\frac{(\Delta x_i - \mu_i)^2}{2\sigma_i^2}}. \tag{5.2}$$

We apply dropout [132] after each fully connected layers to reduce the over-fitting due to the limited number of training data. The amount of dropout applied is 0.4, 0.3 and 0.2 for the input, first and second layers of the network respectively. The dropout ratio is applied in a decreasing manner to cope up with the decrease in the number of parameters for the deeper layers. Each layer is followed by the (PReLU) [62] activation function except the last one which predicts the age. The first layer is

the input layer which takes the 320 dimensional feature vector obtained from the face-identification task. The output of this layer, after the dropout and PReLU operation, is fed to the fist hidden layer containing 320 hidden units. Subsequently, the output propagates to the second hidden layer containing 160 hidden units. The output from this layer is used to generate a scalar value that would describe the apparent age. Figure 5.3 depicts the 3-layer neural network used.

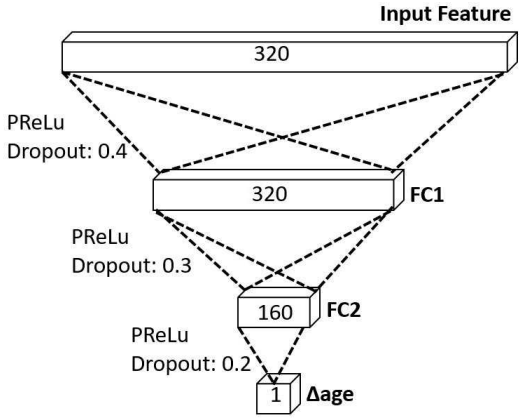


Figure 5.3: The 3-layer neural network used for estimating the increment in age for each age group.

5.3.7 A Toy Example

To illustrate the end-to-end pipeline of the proposed age estimation algorithm, we present a toy example below. In this example, we use the 3 age group setting for the age group classifier where (1) the first age group is from 8 to 14 years, (2) the second 15 to 21, and (3) the third 22 to 28. The age regressor will predict Δage with respect to the mean age of its corresponding group. For example, the regressor for the first age group takes charge of predicting the real value ranging from -3 (*i.e.* 8 - 11 = -3, where 11 is the mean age of the first group) to +3 (*i.e.* 14 - 11 = 3). Now, given a

face image with ground truth age 27 years old, ideally the predicted age group label should be 3 after passing the image into the age group classifier. Then, we will use the third age regressor to predict its Δage which should ideally predict the value as +2 and then we can estimate the apparent age as 25 + 2 = 27 by combining the results of the age group classifier and its corresponding age regressor where 25 is the group mean for the third age group. However, as mentioned in Section 5.3.5, in practice, if the age group classifier makes mistakes, the age estimation results will be wrong. To handle this error, we do the age error correction as described in Section 5.3.5. Now, given another face image with ground truth age 14, incorrectly being classified into third age group, we augment the misclassified samples when we train the regressor. Thus, it can be expected that the Δage should be negative enough, say -5, and as a result, the age estimation will be 25 - 5 = 20 which is still wrong but falls in the range of the second group. Then, we can pass the image again to the second group regressor to get a new estimate, say 18 - 4 = 14. We stop correcting the error when the predicted age and the previous predicted age falls in the same group or reach the maximum number of iterations. That is, we will pass the image to the first regressor again and it will predict 11 + 3 = 14 and then we stop. Otherwise, we continue to perform the correction.

The proposed age estimation algorithm is summarized in Algorithm 1. The execution orders for both the classification and regression parts are written in parallel, and thus it runs in one age group classification plus N Δage regression simultaneously in total. The maximum number of iterations is preset to avoid looping.

5.4 Experimental Results

We evaluate the proposed method on two publicly available datasets: Adience [43] and FG-Net [61]. Both datasets include unconstrained images of individuals which are labeled by their actual biological ages. In addition to these two datasets, we present results on the ICCV 2015 Chalearn 'Looking at people-Age Estimation' challenge dataset [45]. The main difference between this dataset and Adience and FG-Net datasets is that Chalearn includes unconstrained images of individuals labeled by their apparent ages.

5.4.1 Datasets

Adience dataset [43] consists of 26,580 unconstrained images of 2,284 subjects in 8 age groups (0-2, 4-6, 8-13, 15-20, 25-32, 38-43, 48-53, 60+). The standard five-fold, subject-exclusive cross-validation protocol is used for testing (*i.e.*, we merge 0-2 and 4-6 into one for the experiments of Challenge and FG-Net datasets.)

FG-Net aging dataset [61] contains a collection of 1,002 images of 82 subjects, where each image is annotated with true age.

Images of groups [51] consists of 28,231 faces in 5,080 images. Each face is annotated with a label corresponding to one of the seven age groups; 0-2, 3-7, 8-12, 13-19, 20-36, 37-65, 66+ .

Chalearn Workshop Challenge dataset is the first dataset on apparent age estimation containing annotations. The dataset consists of 2,476 training images, 1,136 validation images, and 1,087 test images, which were taken from individuals

aged between 0 to 100. The images are captured in the wild, with variations in pose, illumination and quality. Figure 5.4 shows the distribution of the 'Chalearn Looking at People' Challenge dataset across the different age groups. It is evident from this figure that most of the data are distributed around the age group of 20-50, while there are very few samples in the range of 0-15 and above 55. The remaining data consists of the test set which has not been released publicly.

5.4.2 Experimental Details

For the first stage of age classification, we augmented the training set with the training splits of Adience [43], FG-Net [61] and Images of groups [51] datasets. To evaluate on the FG-Net, we train the seven regressor networks and then pass them through our proposed error correcting mechanism to predict the final age. Although the recently released IMDB-WIKI dataset [121] contains a large collection of images with ages, the number of the images for the young and old age groups is much smaller than other groups and some of the annotations for the dataset are noisy. Due to these factors, we confine the age group ranges to the ones defined by Adience [43] and focus on those previosly well-labelled datasets for this work. The study of the influences by different ranges of age group intervals is left for future work. All the models were trained using Caffe [71]. We also compare the performance of our proposed method with a recently proposed geometry-based method [152], which is referred to as Grassmann-Regression (G-LR).

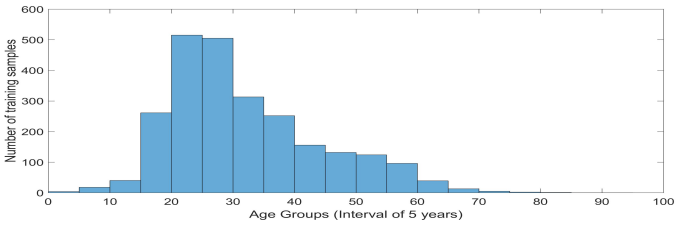


Figure 5.4: Training data distribution of ICCV-2015 Chalearn Looking at People Apparent Age Estimation Challenge, with regard to age groups.

5.4.3 Results

To evaluate the performance of age classification algorithm, we conduct experiments on the Adience dataset [43], by following the 5 fold cross validation protocol described in [94]. From Table 5.2, it can be seen that our approach achieve better performance than the previous state-of-the-art methods. One thing worth noticing is that the accuracy for exact age group classification is around 53%, but the 1-off accuracy is 88.45% (*i.e.*, 1-off means the predicted label is within the neighboring groups of the true one, and 2-off means ± 2 groups). The results demonstrate the need of our error correction module to make the coarse-to-fine strategy to work better.

Method	**Exact**	**1-off**
Best from [43]	45.1 ± 2.6	79.5 ± 1.4
Best from [94]	50.7 ± 5.1	84.7 ± 2.2
Ours	$\mathbf{52.88 \pm 6}$	$\mathbf{88.45 \pm 2.2}$

Table 5.2: Age estimation results on the Adience benchmark. Listed are the mean accuracy ± standard error over all age categories. Best results are marked in bold.

After age group classification, we evaluated the performance of the proposed

method following the protocol provided by the Chalearn 'Looking at People' challenge dataset to further investigate how the coarse-to-fine strategy and error correction mechanism help the age estimation. The error is computed as follows:

$$\varepsilon = 1 - e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \tag{5.3}$$

where x is the estimated age, μ is the provided apparent age label for a given face image, average of at least 10 different user opinions, and σ is the standard deviation of all (at least 10) gauged ages for the given image. We evaluate our method on the validation set of the challenge [45], as the test set annotations are not available for performing analysis. Our baseline approach is to perform age estimation by a single deep regressor (as described in Section 5.3.6) on top of all the DCNN features. From Table 5.3, it shows that the coarse-to-fine strategy improves the prediction results of the baseline approach, and the error correction module further significantly boosts the performance which also demonstrates that the error correction module effectively fixes the errors made by the age classification step. In addition, we also show that the results of end-to-end finetuning on the training data of the challenge data for both baseline and our approach outperform the ones which are trained separately. (*i.e.*, For the results of baseline with end-to-end finetuning, we use the 500,000th model which are trained with the same batch size and learning rate for the proposed approach.) Some prediction sample results from this dataset are shown in Figure 5.5.

By looking at the images, we can infer that our method is robust to pose and

Method	Gaussian Error
G-LR [152]	0.62
Baseline	0.39
Our method without error correction	0.382
Our method with error correction	0.355
Baseline with end-to-end finetuning	0.312
Our method with end-to-end finetuning and error correction	0.297

Table 5.3: Performance comparison on the Chalearn Challenge dataset.

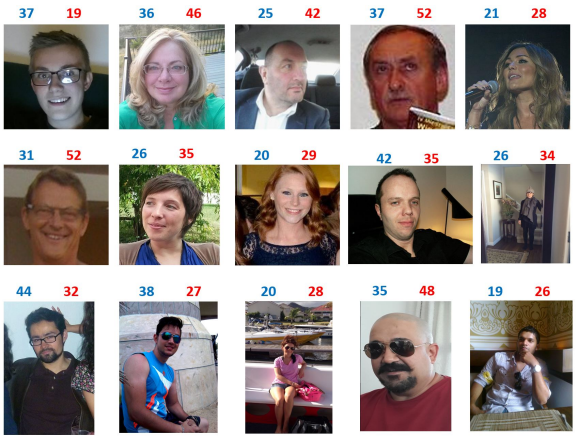


Figure 5.5: Age estimates on the Chalearn Validation set. The incorrect age obtained without using the self correcting module is shown in blue, while the corrected age is given in red.

resolution changes to a certain extent. It fails mostly for extreme illumination and extreme pose scenarios. On further inspection of the Chalearn challenge dataset, we observe the the first stage classification fails to classify correctly when the images have attributes such as hats, glasses, microphone, etc. However, the proposed error correcting mechanism makes it robust to such artifacts. The performance of our method can be improved considerably if we train using age labeled data.

Finally, we further evaluate the proposed method with end-to-end finetuning on the FG-Net dataset (*i.e.*, For FGNet, we set $\sigma = 2$ for Gaussian loss.). Since the training of DCNN is computationally intensive, a fair amount of time is needed to complete the full leave-one-person out (LOPO) evaluations. Thus, we chose to compromise and show a result that demonstrates the performance level as compared to other methods. We randomly chose 73 subjects and used their images as the training data and the rest for testing. Table 5.4 shows the empirical evaluation of our method compared with several other methods proposed in recent years (*i.e.,* Since the test protocol is different from LOPO used for other methods, the results of the proposed method are not directly comparable to others but only as an empirical performance evaluation.). From this table, it can be seen that our method performs comparable to other state-of-the-art age estimation methods. The approach with error correction module performs much better than the one without considering neighboring samples for error correction during training.

Reference	Method	Training/Testing	Result (MAE)
Luu2009 [102]	2 stage SVR in AAM subspace	800/200	4.37
Ylioinas2013 [164]	LBP Kernel Density Estimate	LOPO	5.09
Geng2013 [53]	Label Distribution (CPNN)	LOPO	4.76
Chen2013 [31]	Cumulative Attribute SVR	LOPO	4.67
El Dib2010 [44]	Biologically-Inspired features	LOPO	3.17
Han2013 [61]	Component and holistic BIF	LOPO	4.6
Hong2013 [65]	Biologically InspiredAAM	LOPO	4.18
Chao2013 [28]	Label-sensitive learning	LOPO	4.38
Proposed method	**Classification+Regression**	**890 train , 112 test**	**4.8**
Proposed method	**Classification+Regression+EC**	**890 train , 112 test**	**3.49**

Table 5.4: Performance comparison of different age estimation algorithms on the FG-Net aging database using mean absolute error(MAE). Since the training of DCNNs is computationally intensive, the evaluation of the proposed approach does not follow the full LOPO protocol. The results are for an empirical evaluation to show the performance level of the proposed approach.

5.4.4 Runtime

All the experiments were performed using NVIDIA GTX TITAN-X GPU and the CUDNN library on a 2.3Ghz computer. The first stage training for the classification task took approximately 8 hours whereas training for the second stage took approximately 8 hours per regressor. The system is fully automated with minimal human intervention. The end-to-end system takes about 2.5 seconds per image for age estimation, with only 0.8 seconds being spent in age estimation given the aligned face while the remaining time being spent on face detection and alignment.

5.5 Conclusions

In this work, we proposed a cascaded classification-regression framework to perform unconstrained facial apparent age estimation. The proposed approach estimates the apparent age in a coarse-to-fine manner. The age group classifier gives the

rough age estimate, the regressor per age group gives the fine-grained age estimate, and the age error correcting module fixes incorrect prediction. Our experimental results demonstrate the effectiveness of the proposed approach, especially when only a limited number of training data available in the target domain.

Although our age classifiers and regressors are all based on DCNN, our framework is generic and can be extended to other non-DCNN models. In addition, the same classification-regression framework can be also applied to other vision problems, such as head pose estimation.

Chapter 6: S^2LD : Semi Supervised Landmark Detection for Low Resolution Images

6.1 Introduction

Convolution Neural Networks have revolutionized the computer vision research, to the point that current systems can recognize faces with more than 99.7% [41] accuracy or achieve detection, segmentation and pose estimation results upto subpixel accuracy. These are only few of the many tasks which have seen a significant performance improvements in the last five years. However, CNN-based methods assume access to good quality images. ImageNet [122], COCO [97], CASIA [163], 300W [123] or MPII [4] datasets all consist of high resolution images. As a result of *domain shift*, much lower performance is observed when networks trained on these datasets are applied to images which have suffered degradation due to intrinsic or extrinsic factors. In this work, we address landmark localization in low resolution images. Although, we use face images in our case, the proposed method is also applicable to other tasks, such as human pose estimation. Throughout this chapter we use ***HR and LR to denote high and low resolutions*** respectively.

Facial landmark localization, also known as keypoint or fiducial detection,

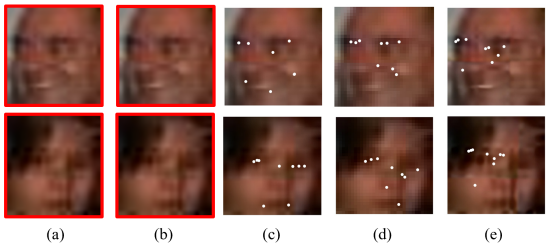


Figure 6.1: Inaccurate landmark detections on low resolution images. We show landmark predicted by different systems. (a) MTCNN [169] and (b) [19] are not able to detect any face in the LR image. (c) Current practice of directly upsampling the low-resolution image to a fixed size of 128×128 by bilinear interpolation. (d) Output from a network trained on downsampled version of HR images. (e) Landmark detection using super-resolved images. **Note:** For visualization purposes images have been reshaped after respective processing. Actual size of the images is in the range of 20×20 pixels

refers to the task of detecting specific points such as eye corners and nose tip on a face image. The detected keypoints are used to align images to canonical coordinates, which are then used as inputs to different convolution networks. It has been experimentally shown in [10], that accurate face alignment leads to improved performance in face verification. Though great strides have been made in this direction, mainly addressing large-pose face alignment, landmark localization for low resolution images, still remains an understudied problem, mostly because of the absence of large scale labeled dataset(s). To the best of our knowledge, for the first time, landmark localization directly on low resolution images is addressed in this work.

Main motivation: In Figure 6.1, we examine possible scenarios which are currently practiced when low resolution images are encountered. Figure 6.1 shows the predicted landmarks when the input image is a LR image of size less than 32×32 pixels. Typically, landmark detection networks are trained with 224×224 crops of HR images taken from AFLW [82] and 300W [123] datasets. During inference, irrespective of resolution, an incoming image is rescaled to 224×224. We deploy two

methods: MTCNN [169] and Bulat *et al.* [19], which have detection and localization built in a single system. In Figure 6.1(a) and (b) we see that these networks failed to detect face in the given image. Figure 6.1(c), shows the outputs when a network trained on high resolution images is applied to a rescaled low resolution one. It is important to note that the trained network, say HR-LD high resolution landmark detector (detailed in Section 6.5.1) achieves state of the art performance on AFLW and 300W test sets. A possible solution is to train a network on sub-sampled images as a substitute for low resolution images. Figure 6.1(d) shows the output of one such network. It is evident from these experiments that networks trained with HR images or subsampled images are not effective for real life LR images. It can also be concluded that subsampled images are unable to capture the distribution of real LR images.

Super-resolution is widely used to resolve LR images to reveal more details. Significant developments have been made in this field and methods based on encoder-decoder architectures and GANs [56] have been proposed. We employ two recent deep learning based methods, SRGAN [91] and ESRGAN [149] to resolve given LR images. It is worth noting that the training data for these networks also include face images. Figure 6.1(e) shows the result when the super-resolved image is passed through HR-LD. It can be hypothesized that possibly, the super-resolved images do not lie in the same space of images using which HR-LD was trained. Super resolution networks are trained using synthetic low resolution images obtained by downsampling the image after applying Gaussian smoothing. In some cases, training data for super-resolution networks consists of paired low and high resolution images.

Neither of the mentioned scenarios is applicable in real life situations.

Main Idea: Different from these approaches, the proposed method is based on the concept of 'generate to adapt'. This work aims to show that landmark localization in LR images can not only be achieved, but it also improves the performance over the current practice. To this end, we first train a deep network which generates LR images from HR images and tries to model the distribution of real LR images in pixel space. Since, there is no publicly available dataset, containing low resolution images along with landmark annotations, we take a semi-supervised approach for landmark detection. We train an adversarial landmark localization network on the generated LR images and hence, switching the roles of generated and real LR images. Heatmaps predicted for unlabelled LR images are also included in the inputs of the discriminators. The adversarial training procedure is designed in a way that in order to fool the discriminators, the heatmap generator has to learn the structure of the face even in low resolution. We perform extensive set of experiments explaining all the design choices. In addition, we also propose new state of the art landmark detector for HR images.

6.2 Related Work

Being one of the most important pre-processing steps in face analysis tasks, facial landmark detection has been a topic of immense interest among computer vision researchers. We briefly discuss some of the methods which use Convolution Neural Networks (CNN). Different algorithms have been proposed in the recent past such

as direct regression approaches of MTCNN by Zhang *et al.* [172] and KEPLER by Kumar *et al.* [85]. The convolution neural networks in MTCNN and KEPLER act as non-linear regressors and learn to directly predict the landmarks. Both works are designed to predict other attributes along with keypoints such as 2D pose, visibility of keypoints, gender and many others. Hyperface by Ranjan *et al.* [115] has shown that learning tasks in one single network does in fact, improves the performance of individual tasks. Recently, architectures based on Encoder-Decoder architecture have become popular and have been used intensively in tasks which require per-pixel labeling such as semantic segmentation [110, 119] and keypoint detection [1, 87, 88, 168]. Despite making significant progress in this field, predicting landmarks on low resolution faces still remains a relatively unexplored topic. All of the works mentioned above are trained on high quality images and their performance degrades on LR images.

One of the closely related works, is Super-FAN [20] by Bulat *et al.*, which makes an attempt to predict landmarks on LR images by super-resolution. However, as shown in experiments in Section 6.4.3, face recognition performance degrades even on super-resolved images. This necessitates that super-resolution, face-alignment and face recognition be learned in a single model, trained end to end, making it not only slow in inference but also limited by the GPU memory constraints. The proposed work is different from [20] in many respects as it needs labeled data only in HR and learns to predict landmarks in LR images in an unsupervised way. Due to adversarial training, the network not only acts as a facial parts detector but also learns the inherent structure of the facial parts. The proposed method makes

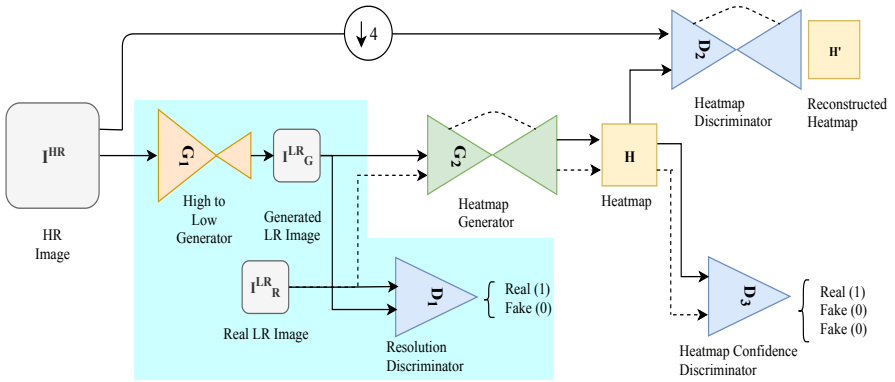


Figure 6.2: Overview of the proposed approach. High resolution input is passed through High-to-Low generator G_1 (shown in cyan colored block). The discriminator D_1 learns to distinguish generated LR images vs. real LR images in an unpaired fashion. This generated image is fed to heatmap generator G_2. Heatmap discriminator D_2 distinguishes generated heatmap vs. groundtruth heatmaps. The pair G_2, D_2 is inspired from BEGAN [13]. In addition to generated and groundtruth heatmaps, the discriminator D_3 also receives predicted heatmaps for real LR images. This enables the generator G_2 to generate realistic heatmaps for un-annotated LR images.

the pre-processing task faster and independent of face verification training. During inference, only the heatmap generator network is used which is based on the fully convolutional architecture of U-Net [119] and works at the spatial resolution of 32×32 making the alignment process real time.

6.3 Proposed Method

S^2LD predicts landmarks directly on a LR image of spatial size less than 32×32 pixels. We show that predicting landmarks directly in LR is more effective than the current practices of rescaling or super-resolution. The entire pipeline can be divided into two stages: (a) Generation of LR images in an unpaired manner (b) Generating heatmaps for target LR images in a semi-supervised fashion. An overview of the proposed approach is shown in Figure 6.2. Being a semi-supervised method, it is

important to first describe the datasets chosen for the experiments.

High Resolution Dataset: We construct the HR dataset by combining the $20,000$ training images from AFLW and the entire 300W dataset. We divide the Widerface dataset [161] into two groups based on their spatial size. The first group consists of images with spatial size between 20×20 and 40×40, whereas the second group consists of images with more than 100×100 pixels. We combine the second group in HR training set, resulting in a total of $35,543$ HR faces. The remaining $4,386$ images from AFLW are used as validation images for the ablative study and test set for the landmark localization task.

Low Resolution Datasets:

- The first group from Widerface dataset consists of $47,046$ faces is used as real LR images for ablative study.
- For face verification experiments, we use recently published TinyFace dataset [33] as the target LR dataset.
- Due to the absence of LR annotated dataset, we create a real LR landmark detection dataset which we call Annotated LR Faces (ALRF) by manually annotating 700 LR images of TinyFace dataset. The details of ALRF creation is discussed in the supplementary materials.

6.3.1 High to Low Generator and Discriminator

High to low generator G_1, shown in Figure 6.8, is designed following the Encoder-Decoder architecture, where both encoder and decoder consists of multiple residual

blocks. The input to the first convolution layer is the HR image concatenated with the noise vector which has been projected using a fully connected layer and reshaped to match the input size. Similar architectures have also been used in [**?**,91]. The encoder in the generator consists of eight residual blocks each followed by a convolution layer to increase the dimensionality. Max-pooling is used after every 2 residual block to decrease the spatial resolution to 4×4, for HR image of 128×128 pixels. The decoder is composed of six residual units followed by up-sampling and convolution layers. Finally, one convolution layer is added in order to output a three channel image. BatchNorm is used after every convolution layer.

The discriminator D_1, shown in Figure 6.8 is also constructed in a similar way, except that due to low spatial resolution of the input image, max-pooling is only used in the last three layers. In Figure 6.2, we use I^{HR} for HR input images of size 128×128, I_G^{LR} for generated LR images of size 32×32 and I_R^{LR} for target LR images of the same size. Spectral Normalization [107] is also used in the convolutional layers of D_1 to satisfy the Lipschitz constraint $\sigma(W) = 1$, presented in Equation 6.1:

$$W_{SN}(W) = \frac{W}{\sigma(W)} \tag{6.1}$$

We train G_1 using a weighted combination of GAN loss; L_2 pixel loss to encourage convergence in initial training iterations and perceptual loss back-propagated from a pre-trained VGG network. The final loss is summarized in Equation 6.2.

$$l_{G_1} = \alpha l_{GAN}^{G} + \beta l_{pixel} + \gamma l_{perceptual} \tag{6.2}$$

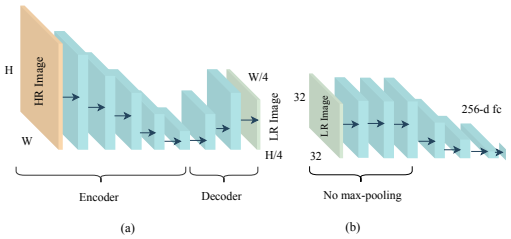


Figure 6.3: (a) High to low generator G_1. Each $\rightarrow$ represents two residual blocks followed by a convolution layer. (b) Discriminator used in D_1 and D_2. Each $\rightarrow$ represents one residual block followed by a convolution layer.

where α, β and γ are hyperpameters which are empirically set. Following recent developments in GANs we experimented with different loss functions. However, we settled on the hinge loss. In Equation 6.2, l^G_{GAN} is computed as:

$$l^G_{GAN} = E_{\hat{x} \in P_g}[\min(0, -1 + D_1(\hat{x}))] \tag{6.3}$$

where P_g is the distribution of generated images I^{LR}_G. Also L_2 pixel loss, l_{pixel}, is derived from the following expression:

$$l_{pixel} = \frac{1}{H \times W} \sum_{i=1}^{W} \sum_{i=1}^{H} (F(I^{HR}) - I^{LR}_G)^2 \tag{6.4}$$

where W and H represent the generated image width and height respectively; also the operation F is implemented as a sub-sampling operation obtained by passing I^{HR} through four average pooling layers. This loss is used to minimize the distance between the generated and sub-sampled images which ensures that the content is not lost during the generation process. To train discriminator D_1 we use hinge loss with gradient penalty and Spectral Normalization for faster training. The discriminator

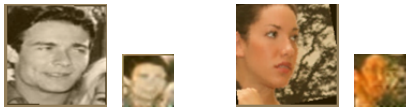


Figure 6.4: Sample outputs of High to Low generation of AFLW dataset. For more results please refer to the supplementary material.

D_1 loss can be defined as:

$$l_{D_1} = l^D_{GAN} + GP \tag{6.5}$$

where

$$l^D_{GAN} = E_{x \in P_r}[\min(0, -1 + D_1(x))] + E_{\hat{x} \in P_g}[\min(0, -1 - D_1(\hat{x}))] \tag{6.6}$$

and P_r is the distribution of real LR images I^{LR}_R from Widerface dataset. GP in Equation 6.5 represents the gradient penalty term. Figure 6.4 shows some sample LR images generated from the network G_1.

6.3.2 Semi-Supervised Landmark Localization

6.3.2.1 Heatmap Generator G_2

The key-point heatmap generator, G_2 in Figure 6.5 produces heatmaps corresponding to N (in our case 19 or 68) key-points in a given image. As mentioned earlier, the objective of this work is to show that landmark prediction directly on LR images is feasible even in the absence of labeled LR data. To this end, we choose a simple network based on the U-Net architecture as the heatmap generator. The

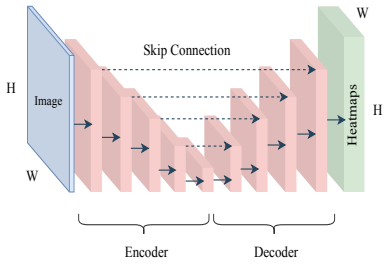


Figure 6.5: Architecture of the heatmap generator G_2. Architecture of this network is based on U-Net. Each $\rightarrow$ represents two residual blocks. $\dashrightarrow$ represents skip connections between the encoder and decoder.

network consists of 16 residual blocks where both encoder and decoder have eight residual blocks. In the last layer, G_2 outputs (N+1) feature maps corresponding to N key-points and 1 background channel. After experimentation, this design for landmark detection has proven to be very effective and results in state of the art resutls for HR landmark predictions. Further architectural details are presented in the supplementary materials.

6.3.2.2 Heatmap Discriminator D_2

The heatmap discriminator D_2 follows the same architecture as the heatmap generator G_2 with different number of input channels, *i.e.*, input to the discriminator is a set of heatmaps concatenated with their respective color images. D_2 receives two sets of inputs: *generated LR image with down-sampled groundtruth heatmaps* and *generated LR images with predicted heatmaps*. This discriminator predicts another set of heatmaps and learns whether the key-points described by the input heatmaps are correct and correspond to the input face image. The quality of the output heatmaps is determined by their similarity to the input heatmaps, following the notion of an autoencoder. The loss is computed as the error between the input

and the reconstructed heatmaps.

6.3.2.3 Heatmap Confidence Discriminator D_3

The architecture of D_3 is identical to D_1 except for the number of input channels. This discriminator receives three inputs: *generated LR image with corresponding groundtruth heatmaps*, *generated LR image with predicted heatmaps* and *target LR image with predicted heatmaps*. D_3 learns to distinguish between the groundtruth and predicted heatmaps. To fool this discriminator, G_2 should learn to: (a) generate heatmaps for generated LR images similar to their respective groundtruth, (b) generate heatmaps for unlabeled target LR images with similar statistical properties to the groundtruth heatmap, *i.e.*, G_2 should understand the inherent structure of the face in LR images and generate accurate and realistic heatmaps.

6.3.3 Semi-supervised Learning

The learning process of this setup is inspired by the seminal works BEGAN [13] and [173] called Energy-based GANs. It is worth recalling that HR images have annotations associated with them and we assume key-point locations in a generated LR image stay relatively the same as its down-sampled version. Therefore, while training G_2, the down-sampled annotations are considered to be groundtruth for the generated LR images.

The discriminator D_2, when the input consists of groundtruth heatmaps, is trained to recognize it and reconstruct a similar one to minimize the error between

the groundtruth and reconstructed heatmaps. On the other hand, if the input consists of generated heatmaps, the discriminator is trained to reconstruct different heatmaps to drive the error as large as possible. The losses are expressed as

$$l_D^{real} = \sum_{i=1}^{N+1} (H_i - D_2(H_i, I_G^{LR}))^2 \tag{6.7}$$

$$l_D^{fake} = \sum_{i=1}^{N+1} (\hat{H}_i - D_2(\hat{H}_i, I_G^{LR}))^2 \tag{6.8}$$

$$l_D^{kp} = l_D^{real} - k_t l_D^{fake} \tag{6.9}$$

where H_i and $\hat{H}_i$ represent the i^{th} key-point groundtruth and generated heatmap of the generated LR image I_G^{LR}. Inspired by BEGAN, we use a variable k_t to control the balance between heatmap generator and discriminator. The variable is updated every t iterations. The adaptive term k_t is defined by:

$$k_{t+1} = k_t + \lambda_k(\gamma l_D^{real} - l_D^{fake}) \tag{6.10}$$

where k_t is bounded between 0 and 1, and λ_k is a hyperparameter. As in Equation 6.9, k_t controls the emphasis on l_D^{fake}. When the generator is able to fool the discriminator, l_D^{fake} becomes smaller than γl_D^{real}. As a result of this k_t increases, making the term l_D^{fake} dominant. The amount of acceleration to train on l_D^{fake} is adjusted proportional to $\gamma l_D^{real} - l_D^{fake}$, *i.e* the distance the discriminator falls behind the generator. Similarly, when the discriminator gets better than the generator, k_t decreases, to slow down the training on l_D^{fake} making the generator and the discriminator train

together.

The discriminator D_3 is trained using the loss function from Least squares GAN [104] as shown in Equation 6.11. This loss function was chosen to be consistent with the losses computed by D_2.

$$l_D^{conf} = \mathbb{E}_{x \in \mathbb{P}_r}[(D_3(x) - 1)^2] + \mathbb{E}_{\hat{x} \in \mathbb{P}_g}[D_3(\hat{x})^2] + \mathbb{E}_{\hat{y} \in \mathbb{P}_g}[D_3(\hat{y})^2] \tag{6.11}$$

It is noteworthy to mention in this case $\mathbb{P}_r$ represents the groundtruth heatmaps distribution on generated LR images, while $\mathbb{P}_g$ represents the distribution on generated heatmaps of generated LR images and real LR images.

The generator G_2 is trained using a weighted combination of losses from the discriminators D_2 and D_3 and l_{MSE} heatmap loss. The loss functions for the generator G_2 are described in the following equations:

$$l_G^{MSE} = \sum_{i=1}^{N+1} (H_i - G_2(I_G^{LR}))^2 \tag{6.12}$$

$$l_G^{kp} = \sum_{i=1}^{N+1} (\hat{H}_i - D_2(\hat{H}_i, I_g^{LR}))^2 \tag{6.13}$$

$$l_G^{conf} = \mathbb{E}_{x \in \mathbb{P}_g}[(D_3(x) - 1)^2] \tag{6.14}$$

$$l_G = a l_G^{MSE} + b l_G^{kp} + c l_G^{conf} \tag{6.15}$$

where a, b and c are hyper parameters set empirically obeying $a l_G^{MSE} > b l_G^{kp} > c l_G^{conf}$. We put more emphasis on l_G^{MSE} to encourage convergence of the model in initial

Figure 6.6: Sample key-point detections on TinyFace images.

iterations. Some target LR images with key-points predicted from the G_2 are shown in Figure 6.6.

6.4 Experiments and Results

6.4.1 Ablation Experiments

We experimentally demonstrated in Section 6.1 (Figure 6.1) that networks trained on HR images perform poorly on LR images. Therefore, we propose the semi-supervised learning as mentioned in Section 6.3. With the above mentioned networks and loss functions it is important to understand the implication of each component. This section examines each of the design choices quantitatively. To this end, we first train the high to low resolution networks, and generate LR images of $4,386$ AFLW test images. In the absence of real LR images with annotated landmarks, this is done to create a substitute for low resolution dataset with annotations on which localization performance can be evaluated. We also generate subsampled version of the $20,000$ AFLW trainset and $4,386$ AFLW testset using average pooling after applying Gaussian smoothing. Data augmentation techniques such as random scaling $(0.9, 1.1)$, random rotation $(-30°, 30°)$ and random translation upto 20 pixels are used.

Evaluation Metric: Following most previous works, we obtain error for each

test sample by averaging normalized errors for all annotated landmarks. For AFLW, the obtained error is normalized by the ground truth bounding box size over all visible points whereas for 300W, the error is normalized by the inter-pupil distance. Wherever applicable NRMSE stands for Normalized Root Mean Square Error.

Training Details: All the networks are trained in Pytorch using the Adam optimizer with an initial learning rate of $2\text{E}{-4}$ and β_1, β_2 values of $0.5, 0.9$. We train the networks with a batch size of 32 for 200 epochs, while dropping the learning rates by 0.5 after 80 and 160 epochs.

Setting S1: *Train networks on subsampled images?* We only train network G_2 with the subsampled AFLW training images using the loss function in Equation 6.12, and evaluate the performance on generated LR AFLW test images.

Setting S2: *Train networks on generated LR images?* In this experiment, we train the network G_2 using generated LR images, in a supervised way using the loss function from Equation 6.12. We again evaluate the performance on generated LR AFLW test images.

Observation: From the results summarized in Table 6.1b it is evident that there is a significant reduction in localization error when G_2 is trained on generated LR images validating our hypothesis that subsampled images on which many super-resolution networks are trained may not be a correct representative of real LR images. Hence, we need to train the networks on real LR images.

Setting S3: *Does adversarial training help?* This question is asked in order to understand the importance of training the heatmap generator G_2 in an adversarial way. In this experiment, we train G_2 and D_2 using the losses in Eqs 6.7, 6.8, 6.12,

Method	NRMSE (all)	NRMSE (479 images)	Time
MTCNN [169]	-	0.9736	0.388 s
HRNet [133]	0.4055	0.3107	0.076 s
SAN [42]	0.3901	0.3141	0.0178 s
Proposed	**0.257**	**0.1803**	**0.0105 s**

(a)

Setting	NRMSE±std	auc@0.07	auc@0.08
S1	11.33 ± 9.81	11.897	21.894
S2	4.23 ± 4.52	50.843	55.751
S3	4.120 ± 4.43	51.889	56.791
S4	4.123 ± 4.394	51.775	56.697

(b)

Table 6.1: (a) Landmark Detection Error on Real Low Resolution dataset. (b) Table for ablation experiments under different settings on synthesized LR images.

6.13. Metrics are calculated on the generated LR AFLW test images and compared against the experimental setting mentioned in S2 above.

Setting S4: *Does G_2 trained in adversarial manner scale to real LR images?* In this experiment, we wish to examine if training networks G_2, D_2 and D_3 jointly, improves the performance on real LR images from Widerface dataset.(see Section 6.3 for datasets)

Observation: From Table 6.1b we observe that the network trained with setting S3 performs marginally better compared to setting S4. However, since there are no keypoint annotations available for the Widerface dataset, conclusions cannot be drawn from the drop in performance. Hence, in the following subsection 6.4.3, we leap towards understanding this phenomenon indirectly, by aligning the faces using the models from setting S3 and setting S4 and evaluating face recognition performances.

6.4.2 Experiments on Low Resolution images

We choose to perform direct comparison on a real LR dataset. Two recent state of the art methods Style Aggregated Networks [42] and HRNet [133]. To create a real LR landmark detection dataset which we call Annotated LR Faces (ALRF), we randomly selected 700 identities from the TinyFace dataset, out of which one LR image (less than 32×32 pixels and more than 15×15 pixels) per identity was randomly selected, resulting in a total of 700 LR images. Next, three individuals were asked to manually annotated all the images with 5 landmarks(two eye centers, nose tip and mouth corners) in MTCNN [169] style, where invisible points were annotated with -1. The mean of the points obtained from the three users were taken to be the groundtruth. As per convention, we used Normalised Mean Square Error (NRMSE), averaged over all visible points and normalized by the face size as the comparison metric. Table 6.1a shows the results of this experiment. We also calculate time for forward pass of one image in a single gtx1080. Without loss of generality, the results can be extrapolated to other existing works as [42] and [133] are currently state of the art. MTCNN which has detection and alignment in a single system was able to detect only 479 faces out of 700 test images.

6.4.3 Face Recognition experiments

In the previous section, we performed ablative studies on the generated LR AFLW images. Although convenient to quantify the performance, it does not uncover the importance of training three networks jointly in a semi-supervised way. Therefore,

Figure 6.7: Snippet of the annotation tool used.

in this section, we choose to evaluate the models from setting S3 and setting S4 (Section 6.4.1), by comparing the statistics obtained by applying the two models to align face images for face recognition task.

We use recently published and publicly available, Tinyface [33] dataset for our experimental evaluation. It is one of the very few datasets aimed towards understanding LR face recognition and consists of $5,139$ labeled facial identities with an average of three face images per identity, giving a total of $15,975$ LR face images (average 20×16 pixels). All the LR faces in TinyFace are collected from the web (PIPA [170] and MegaFace2 [108]) across diverse imaging scenarios, captured under uncontrolled viewing conditions in pose, illumination, occlusion and background. $5,139$ known identities is divided into two splits: $2,570$ for training and the remaining $2,569$ for test.

Evaluation Protocol: In order to compare model performances, we adopt the closed-set face identification (1:N matching) protocol. Specifically, the task is to match a given probe face against a gallery set of enrolled face images with true match from the gallery at top-1 of the ranking list. For each test class, half of

Setting	L1	L2	L3	L4	L5
top-1	31.17	35.11	39.03	39.87	43.82

(a)

Setting	top-1	top-5	top-10	top-20	mAP
Baseline (ArcFace [41])	34.71	44.82	49.01	53.70	0.32
I1	34.01	41.98	45.36	49.22	0.29
I2	45.04	56.30	60.11	63.71	0.43
I3	**51.10**	**61.05**	**64.38**	**67.89**	**0.47**

(b)

Table 6.2: Verification performance on Tinyface dataset under different settings (a) LightCNN trained from scratch (b) Using Inception-ResNet pretrained on MsCeleb-1M

the face images are randomly assigned to the probe set, and the remaining to the gallery set. For the purpose of this chapter, we drop the distractor set as this does not divulge new information while significantly slowing down the evaluation process. For face recognition evaluation, we report statistics on Top-k (k=1,5,10,20) statistics and mean average precision (mAP).

Experiments with network trained from scratch: Since the number of images in TinyFace dataset is much smaller compared to larger datasets such as CASIA [163] or MsCeleb-1M [60], we observed that training a very deep model like Inception-ResNet [136], quickly leads to over-fitting. Therefore, we adopt a CNN with fewer parameters, specifically, LightCNN [154]. Since inputs to the network are images of size 32×32, we disable first two max-pooling layers. After detecting the landmarks, training and testing images are aligned to the canonical coordinates using affine transformation. We train 29 layer LightCNN models using the training split of TinyFace dataset under the following settings:

Setting L1: *Train networks on generated LR images?* In this setting, we use the model trained under the setting S2 from the previous section 6.4.1. In this

setting, network G_2 is trained using generated LR images in a supervised way using the loss function from Equation 6.12.

Setting L2: *Does adversarial training help?* We use the model trained from setting S3 (section 6.4.1) to align the faces in training and testing sets. In this setting networks G_2 and D_2 are trained using a weighted combination of L_2 pixel loss and GAN losses from Equations 6.7, 6.8, 6.12, 6.13.

Setting L3: *Does G_2 trained in adversarial manner scale to real LR images?* In this setting, networks G_2, D_2 and D_3 are trained jointly in a semi-supervised way. We use Tinyface training images as real low resolution images. Later, Tinyface training and testing images are aligned using the trained model for training LightCNN model.

Setting L4: *End-to-end training?* Under this setting, we also train the High to Low networks G_1 and D_1, using the training images from Tinyface dataset as real LR images. We reduce the amount of data-augmentation in this case to resemble tiny face dataset images. With the obtained trained model, landmarks are extracted and images are aligned for LightCNN training.

Setting L5: *End-to-end training with pre-trained weights?* This setting is similar to the setting L4 above, except instead of training a LightCNN model from scratch we initialize the weights from a pre-trained model, trained with CASIA-Webface dataset.

Observation: The results in Table 6.2a summarizes the results of the experiments done under the settings discussed above. We see that although, we observed a drop in performance in landmark localization when training the three networks

jointly (Table 6.1b), there is a significant gap in rank-1 performance between setting L2 and L3. This indicates that with semi-supervised learning G_2 generalizes well to real LR data, and hence also validates our hypothesis of training G_2, D_2 and D_3 together. Unsurprisingly, insignificant difference is seen between settings L3 and L4.

Experiments with pre-trained network: Next, to further understand the implications of joint semi-supervised learning, we design another set of experiments. In these experiments, we use a pre-trained Inception-ResNet model, trained on MsCeleb-1M using ArcFace [41] and Focal Loss [96]. This model expects an input of size 112×112 pixels, hence the images are resized after alignment in low resolution. Using this pre-trained network, we perform the following experiments:

Setting	top-1	top-5	top-10	top-20	mAP
A1	11.75	14.58	24.57	30.47	0.10
A2	26.21	34.76	39.03	43.99	0.24

Table 6.3: Face recognition performance using super-resolution before face-alignment

Baseline: For the baseline experiment, we choose to follow the usual practice of re-scaling the images to a fixed size irrespective of resolution. We trained our own HR landmark detector (HR-LD) on $20,000$ AFLW images for this purpose. Tinyface gallery and probe images are resized to 128×128 and used by the landmark detector as inputs. Using the predicted landmarks, images are aligned to a canonical coordinates similar to ArcFace [41]. Baseline performance was obtained by computing cosine similarity between gallery and probe features extracted from the network after feed-forwarding the aligned images.

Setting I1: *Does adversarial training help?* The model trained for S3 (Section

6.4.1) is used to align the images directly in low resolution. Features for gallery and probe images are extracted after the rescaling the images and cosine distance is used to measure the similarity and retrieve the images from the gallery.

Setting I2: *Does G_2 trained in adversarial manner scale to real LR images?* For this experiment, the model trained for L3 in Section 6.4.3 is used for landmark detection in LR. To recall, in this setting, the three models G_2, D_2 and D_3 (with G_1 and D_1 frozen) are trained jointly in a semi-supervised way and Tinyface training images are used as real LR data for D_3.

Setting I3: *End-to-end training?* In this case, we align the images using the model from setting L4 from Section 6.4.3. In this case, we also trained High to low networks (G_1 and D_1) using training images from Tinyface dataset as real LR images. After training the model for 200 epochs, the weights are frozen to train G_2, D_2 and D_3 in a semi-supervised way.

Observation: With no surprise, we observe that (from Table 6.2b) training the heatmap prediction networks in a semi-supervised manner, and aligning the images directly in low resolution, improves the performance of any face recognition system trained with HR images.

6.5 Evaluation on the IJB-S dataset

Along with the method to predict landmarks in low resolution images, this work presents a rather counter-intuitive result that performing landmark detection directly in low resolution leads to higher face recognition performance. To understand

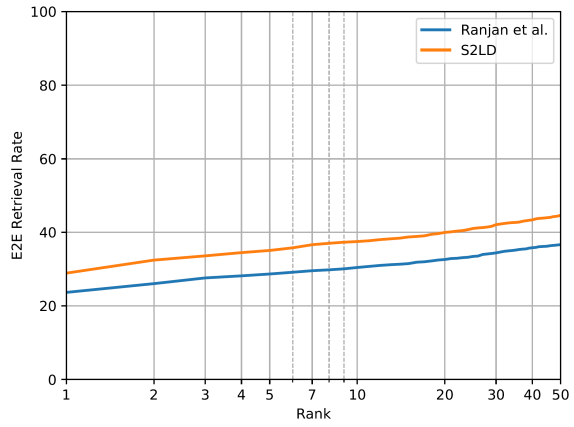
	UltraFace	**Semi-Supervised**
Rank 1	23.65	28.88
Rank 2	26.03	32.42
Rank 3	27.58	33.57
Rank 4	28.14	34.46
Rank 5	28.64	35.05
Rank 7	29.54	36.61
Rank 10	30.42	37.46
Rank 20	32.58	39.95
Rank 30	34.38	42.05
Rank 40	35.79	43.34
Rank 50	36.69	44.61

Table 6.4: Retrieval rates at different ranks(Higher is better)

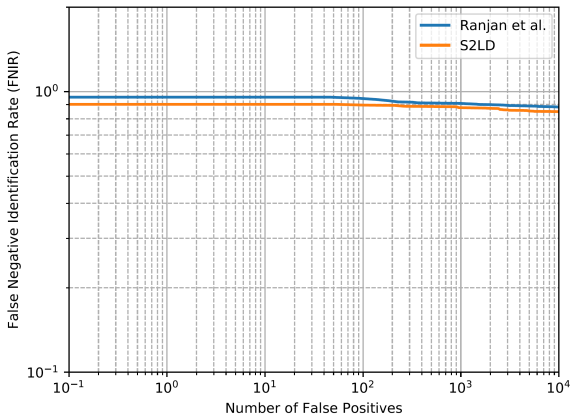
FPIR/Method	**UltraFace**	**Semi-Supervised**
1e2	0.9450	0.8959
1e3	0.9081	0.8767
1e4	0.8808	0.8485
1e5	0.8114	0.7720

Table 6.5: False negative rates at different false positive rates. (Lower is better)

this further we performed experiments on recently released IJB-S dataset [**?**]. IJB-S dataset is one of the most challenging dataset available, and consists of several videos collected with surveillance cameras. The subjects in this dataset are extremely challenging to verify because of the distance from the camera and low resolution. We randomly selected 10 videos from the dataset which contained at least 5 subjects from the two galleries the dataset provides. We used surveillance to booking protocol for the purpose of this experiment. Only 10 videos were chosen attributing to the fact that IJB-S is an extremely large dataset and experimenting on the entire dataset takes more than a month on a single GPU machine. Tables 6.4 and 6.5 shows retrieval rates at different ranks and false negative rates vs false positives. We compare with [113].



(a)



(b)

Figure 6.8: (a) Retrieval rates at different ranks. (b) False negatives at different false positive rates.

6.5.1 Additional Experiments:

Setting A1: *Does Super-resolution help?* The aim of this experiment is to understand if super-resolution can be used to enhance the image quality before landmark detection. We use SRGAN [91] to super-resolve the images before using face alignment method from Bulat *et al.* [19] to align the images.

Setting A2: *Does Super-resolution help?* In this case, we use ESRGAN [149] to super-resolve the images before using HR-LD (below) to align.

Observation: It can be observed from Table 6.3, that face recognition per-

formance obtained after aligning super-resolved images is not at par even with the baseline. It can be hypothesized that possibly super-resolved images do not represent HR images using which [19] or HR-LD are trained.

High Resolution Landmark Detector (HR-LD) For this experiment, we train G_2 on high resolution images of size 128×128 (for AFLW and 300W) using $l_M SE$ loss from Equation 6.12. We evaluate the performance of this network on common benchmarks of AFLW-Full test and 300W test sets, shown in Table 6.6. A few sample outputs are shown in Figure 6.9

Method		**300W**		**AFLW**
	Common	**Challenge**	**Full**	**Full**
RCPR [24]	6.18	17.26	8.35	-
SDM [158]	5.57	15.40	7.52	5.43
CFAN [168]	5.50	16.78	7.69	-
LBF [116]	4.95	11.98	6.32	4.25
CFSS [174]	4.73	9.98	5.76	3.92
TCDCN [172]	4.80	8.60	5.54	-
MDM [142]	4.83	10.14	5.88	-
PCD-CNN [88]	3.67	7.62	4.44	2.36
SAN [42]	3.41	7.55	4.24	1.91
LAB [153]	3.42	6.98	4.12	1.85
HR-LD	**3.60**	**7.301**	**4.325**	**1.753**

Table 6.6: Comparison of the proposed method with other state of the art methods on AFLW (Full) and 300-W testsets. The NMEs for comparison on 300W dataset are taken from the Table 3 of [103]. In this case G_2 is trained in supervised manner using high resolution images of size 128×128.

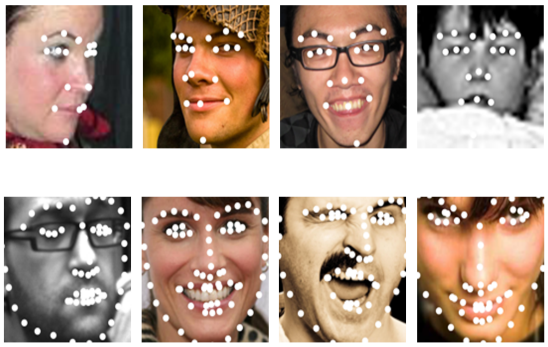


Figure 6.9: Sample outputs obtained by training G_2 with HR images. First row shows samples from AFLW test set. Second row shows sample images from 300W test set. Last two columns of second row shows outputs from challenging subset of 300W

6.6 Conclusion

In this chapter, we first present an analysis of landmark detection methods when applied to LR images, and the implications on face recognition. We also discuss the proposed method for predicting landmarks directly on LR images. We show that the proposed method improves face recognition performance over commonly used practices of rescaling and super-resolution. As a by-product, we also developed a simple but state of the art landmark detection network. Although, low resolution is chosen as the source of degradation, however, the method can trivially be extended to capture other degradations in the imaging process, such as motion blur or climatic turbulence. In addition, the proposed method can be applied to detect human keypoints in LR in order to improve skeletal action recognition. In the era of deep learning, LR landmark detection and face recognition is a fairly untouched topic, however, we believe this work will open new avenues in this direction.

Chapter 7: Conclusion

This dissertation has addressed one of the major face-centric computer vision problems: non-rigid alignment of deformable faces. We discussed four different methods for facial keypoint localization. With extensive experiments we demonstrated the state-of-the art performance of each of the method.

In Chapter 1 we discussed the motivation behind the problem of face alignment and the associated challenges. Next we presented a cascade linear regressor based method which takes localized deep features from a face verification network in order to localize landmark points. It was shown by experiments that face verification networks capture localized information to verify faces and can also be used for landmark localization. The proposed method is one of the first methods to use deep features for keypoint localization.

We detailed another cascade regression based method KEPLER, based on multi-task learning framework in Chapter 2. The approach of cascade regression makes the method somewhat slower but yields precise locations of keypoints. Along with the keypoints KEPLER is also able to predict 3D head pose from a single image. We also developed a new Channeled Inception Network which was trained in a multi-task fashion to achieve precision over keypoint locations. To tackle the

effect of vanishing gradients in a very deep network we also used a novel loss function.

In Chapter 3, we discussed Pose Conditioned Dendritic CNN, where the prediction of keypoints was conditioned on the 3D head pose. We showed that the knowledge of 3D headpose assist in obtaining accurate keypoints. We also modelled the geometric relationships among different facial parts in a dendritic network. An auxiliary network was used to predict other attributes, such as occlusion and visibility. The proposed method is able to predict different attributes of a face image including keypoints in a single pass. This tackles the slower run time of the two methods by learning the locations of keypoints in a single convolution method making it faster. To tackle the imbalance between positive and negative samples we also discussed a novel Mask Softmax Loss Function.

In Chapter 5, we discussed an application of face alignment for the task of apparent age estimation. Face images are aligned with LDDR before being passed through the CNN for age estimation. We analyzed the properties of the convolution networks and develop efficient error correction strategy for better age estimates.

The above methods assumed access to high quality images while training and testing. However, a huge amount of data collected are from closed circuit cameras which capture images in much lower resolution. In the semi-supervised method presented in Chapter 6 we showed how we can transfer the knowledge learnt from high resolution images to predict keypoints in naturally degraded images. We also showed the impact keypoint localization has on the task of face verification. With experiments we demonstrated that aligning keypoints in lower resolution achieves better face verification performance than the current practice of upsampling and

aligning.

7.0.1 Future Work

- **Alignment in videos**: The proposed methods are suitable for obtaining precise keypoint locations from still images. However, we observe a temporal relationships between keypoints in a video. One future direction is in exploiting the temporal information and utilizing it for simultaneous tracking and keypoint localization.

- **Alignment of climatically degraded images**: In the age of technical advancement, people are always taking images, in adverse climatic and illumination conditions, such as in rain or under the sun. Images are also taken while in motion, such as running or in a bus. These degrade the quality of images and the current systems of keypoint localization perform poorly on these images. In future, we plan to extend this research, which will enable accurate keypoint localization even under extreme degradation.

Bibliography

[1] A recurrent autoencoder-decoder for sequential face alignment. `http://arxiv.org/abs/1608.05477`. Accessed: 2016-08-16.

[2] *Pose-free Facial Landmark Fitting via Optimized Part Mixtures and Cascaded Deformable Shape Model*, 2013.

[3] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2037–2041, Dec 2006.

[4] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[5] E. Antonakos, J. Alabort i medina, and S. Zafeiriou. Active pictorial structures. In *CVPR*, pages 5435–5444, Boston, MA, USA, June 2015.

[6] E. Antonakos, P. Snape, G. Trigeorgis, and S. Zafeiriou. Adaptive cascaded regression. In *ICIP'16*, Phoenix, AZ, USA, September 2016.

[7] Epameinondas Antonakos, Joan Alabort-i Medina, and Stefanos Zafeiriou. Active pictorial structures. June 2015.

[8] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic. Incremental face alignment in the wild. In *CVPR 2014*, 2014.

[9] Akshay Asthana, Stefanos Zafeiriou, Shiyang Cheng, and Maja Pantic. Robust discriminative response map fitting with constrained local models. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '13, pages 3444–3451, Washington, DC, USA, 2013. IEEE Computer Society.

[10] Ankan Bansal, Carlos Castillo, Rajeev Ranjan, and Rama Chellappa. The do's and don'ts for cnn-based face verification. *arXiv preprint arXiv:1705.07426*, 2017.

[11] V. Belagiannis and A. Zisserman. Recurrent human pose estimation. In *2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017)*, pages 468–475, May 2017.

[12] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 545–552, Washington, DC, USA, 2011. IEEE Computer Society.

[13] David Berthelot, Tom Schumm, and Luke Metz. BEGAN: boundary equilibrium generative adversarial networks. *CoRR*, abs/1703.10717, 2017.

[14] Chandrasekhar Bhagavatula, Chenchen Zhu, Khoa Luu, and Marios Savvides. Faster than real-time facial alignment: A 3d spatial transformer network approach in unconstrained poses. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[15] Chandraskehar Bhagavatula, Chenchen Zhu, Khoa Luu, and Marios Savvides. Faster than real-time facial alignment: A 3d spatial transformer network approach in unconstrained poses. *CoRR*, abs/1707.05653, 2017.

[16] Vishnu Naresh Boddeti, Myung-Cheol Roh, Jongju Shin, Takaharu Oguri, and Takeo Kanade. Face alignment robust to pose, expressions and occlusions. *CoRR*, abs/1707.05938, 2017.

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

[18] Adrian Bulat and Georgios Tzimiropoulos. Binarized convolutional landmark localizers for human pose estimation and face alignment with limited resources. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[19] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In *International Conference on Computer Vision*, 2017.

[20] Adrian Bulat and Georgios Tzimiropoulos. Super-fan: Integrated facial landmark localization and super-resolution of real-world low resolution faces in arbitrary poses with gans. *CoRR*, abs/1712.02765, 2017.

[21] Adrian Bulat and Yorgos Tzimiropoulos. Convolutional aggregation of local evidence for large pose face alignment. In Edwin R. Hancock Richard C. Wilson and William A. P. Smith, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 86.1–86.12. BMVA Press, September 2016.

[22] X. P. Burgos-Artizzu, P. Perona, and P. Dollar. Robust face landmark estimation under occlusion. In *2013 IEEE International Conference on Computer Vision*, pages 1513–1520, Dec 2013.

[23] Xavier P. Burgos-Artizzu, Pietro Perona, and Piotr Dollar. Robust face landmark estimation under occlusion. *Computer Vision, IEEE International Conference on*, 0:1513–1520, 2013.

[24] 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.

[25] Joao Carreira, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. Human pose estimation with iterative error feedback. 2015.

[26] Jan Cech, Vojtěch Franc, Michal Uricar, and Jiri Matas. Multi-view facial landmark detection by using a 3d shape model. *Image and Vision Computing*,

47:60 – 70, 2016. 300-W, the First Automatic Facial Landmark Detection in-the-Wild Challenge.

[27] Wei-Lun Chao, Jun-Zuo Liu, and Jian-Jiun Ding. Facial age estimation based on label-sensitive learning and age-oriented regression. *Pattern Recognition*, 46(3):628 – 641, 2013.

[28] Jun-Cheng Chen, Vishal M. Patel, and Rama Chellappa. Unconstrained face verification using deep CNN features. *CoRR*, abs/1508.01722, 2015.

[29] Jun-Cheng Chen, Rajeev Ranjan, Amit Kumar, Ching-Hui Chen, Vishal M. Patel, and Rama Chellappa. An end-to-end system for unconstrained face verification with deep convolutional neural networks. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, December 2015.

[30] Ke Chen, Shaogang Gong, Tao Xiang, and C.C. Loy. Cumulative attribute space for age and crowd density estimation. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2467–2474, June 2013.

[31] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *CoRR*, abs/1412.7062, 2014.

[32] Zhiyi Cheng, Xiatian Zhu, and Shaogang Gong. Low-resolution face recognition. *CoRR*, abs/1811.08965, 2018.

[33] Xiao Chu, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Structured feature learning for pose estimation. In *CVPR*, 2016.

[34] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models—their training and application. *Comput. Vis. Image Underst.*, 61(1):38–59, January 1995.

[35] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(6):681–685, Jun 2001.

[36] David Cristinacce and Tim Cootes. Feature detection and tracking with constrained local models. pages 929–938, 2006.

[37] David Cristinacce and Tim Cootes. Automatic feature localisation with constrained local models. *Pattern Recogn.*, 41(10):3054–3067, October 2008.

[38] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, June 2005.

[39] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[40] Jiankang Deng, Jia Guo, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. *CoRR*, abs/1801.07698, 2018.

[41] Xuanyi Dong, Yan Yan, Wanli Ouyang, and Yi Yang. Style aggregated network for facial landmark detection. In *CVPR*, pages 379–388, 2018.

[42] E. Eidinger, R. Enbar, and T. Hassner. Age and gender estimation of unfiltered faces. *Information Forensics and Security, IEEE Transactions on*, 9(12):2170–2179, Dec 2014.

[43] M.Y. El Dib and M. El-Saban. Human age estimation using enhanced bio-inspired features (ebif). In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 1589–1592, Sept 2010.

[44] S. Escalera, J. Fabian, P. Pardo, X. Baro, J. Gonzalez, H.J. Escalante, and I. Guyon. Chalearn 2015 apparent age and cultural event recognition: datasets and results.

[45] S. Escalera, M. T. Torres, B. Martínez, X. Baró, H. J. Escalante, I. Guyon, G. Tzimiropoulos, C. Corneanu, M. Oliu, M. A. Bagheri, and M. Valstar. Chalearn looking at people and faces of the world: Face analysisworkshop and challenge 2016. In *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 706–713, June 2016.

[46] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[47] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1915–1929, 2013.

[48] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9):1627–1645, September 2010.

[49] Y. Fu, G. Guo, and T. Huang. Age synthesis and estimation via faces: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(11):1955–1976, 2010.

[50] A. Gallagher and T. Chen. Understanding images of groups of people. In *Proc. CVPR*, 2009.

[51] X. Geng, C. Yin, and Z. Zhou. Facial age estimation by learning from label distributions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(10):2401–2412, 2013.

[52] Xin Geng, Zhi-Hua Zhou, and K. Smith-Miles. Automatic age estimation based on facial aging patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(12):2234–2240, Dec 2007.

[53] G. Ghiasi and C. C. Fowlkes. Occlusion coherence: Localizing occluded faces with a hierarchical deformable part model. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1899–1906, June 2014.

[54] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition*, 2014.

[55] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

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

[57] Ralph Gross, Iain Matthews, and Simon Baker. Active appearance models with occlusion. *Image Vision Comput.*, 24(6):593–604, June 2006.

[58] Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and Simon Baker. Multi-pie. *Image Vision Comput.*, 28(5):807–813, May 2010.

[59] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. *CoRR*, abs/1607.08221, 2016.

[60] Hu Han, C. Otto, and A.K. Jain. Age estimation from face images: Human vs. machine performance. In *Biometrics (ICB), 2013 International Conference on*, pages 1–8, June 2013.

[61] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *arXiv preprint arXiv:1502.01852*, 2015.

[62] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852, 2015.

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

[64] Lijun Hong, Di Wen, Chi Fang, and Xiaoqing Ding. A new biologically inspired active appearance model for face age estimation by using local ordinal ranking. In *Proceedings of the Fifth International Conference on Internet Multimedia Computing and Service*, ICIMCS '13, pages 327–330, New York, NY, USA, 2013. ACM.

[65] G. S. Hsu, K. H. Chang, and S. C. Huang. Regressive tree structured model for facial landmark localization. In *ICCV*, Dec 2015.

[66] Gee-Sern Hsu, Kai-Hsiang Chang, and Shih-Chieh Huang. Regressive tree structured model for facial landmark localization. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[67] Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5mb model size. *arXiv:1602.07360*, 2016.

[68] Aaron S Jackson, Adrian Bulat, Vasileios Argyriou, and Georgios Tzimiropoulos. Large pose 3d face reconstruction from a single image via direct volumetric cnn regression. *International Conference on Computer Vision*, 2017.

[69] Vidit Jain and Erik Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. Technical Report UM-CS-2010-009, University of Massachusetts, Amherst, 2010.

[70] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM International Conference on Multimedia*, pages 675–678, 2014.

[71] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[72] Amin Jourabloo and Xiaoming Liu. Pose-invariant 3d face alignment. In *ICCV*, Santiago, Chile, December 2015.

[73] Amin Jourabloo and Xiaoming Liu. Large-pose face alignment via cnn-based dense 3d model fitting. In *CVPR*, Las Vegas, NV, June 2016.

[74] Amin Jourabloo and Xiaoming Liu. Large-pose face alignment via cnn-based dense 3d model fitting. In *Proc. IEEE Computer Vision and Pattern Recognition*, Las Vegas, NV, June 2016.

[75] Amin Jourabloo, Xiaoming Liu, Mao Ye, and Liu Ren. Pose-invariant face alignment with a single cnn. In *In Proceeding of International Conference on Computer Vision*, Venice, Italy, October 2017.

[76] Maya Kabkab, Azadeh Alavi, and Rama Chellappa. Dcnns on a diet: Sampling strategies for reducing the training set size. *CoRR*, abs/1606.04232, 2016.

[77] V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1867–1874, 2014.

[78] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *CVPR*, 2014.

[79] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4873–4882, June 2016.

[80] Brendan F. Klare, Ben Klein, Emma Taborsky, Austin Blanton, Jordan Cheney, Kristen Allen, Patrick Grother, Alan Mah, Mark Burge, and Anil K. Jain. Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. June 2015.

[81] 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 *First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies*, 2011.

[82] S. N. Kohail. Using artificial neural network for human age estimation based on facial images. In *International Conference on Innovations in Information Technology*, pages 215–219. IEEE, 2012.

[83] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[84] A. Kumar, A. Alavi, and R. Chellappa. Kepler: Keypoint and pose estimation of unconstrained faces by learning efficient h-cnn regressors. In *2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017)*, pages 258–265, May 2017.

[85] Amit Kumar, Azadeh Alavi, and Rama Chellappa. KEPLER: keypoint and pose estimation of unconstrained faces by learning efficient H-CNN regressors. *CoRR*, abs/1702.05085, 2017.

[86] Amit Kumar and Rama Chellappa. A convolution tree with deconvolution branches: Exploiting geometric relationships for single shot keypoint detection. *CoRR*, abs/1704.01880, 2017.

[87] Amit Kumar and Rama Chellappa. Disentangling 3d pose in A dendritic CNN for unconstrained 2d face alignment. *CoRR*, abs/1802.06713, 2018.

[88] Amit Kumar, Rajeev Ranjan, Vishal M. Patel, and Rama Chellappa. Face alignment by local deep descriptor regression. *CoRR*, abs/1601.07950, 2016.

[89] Vuong Le, Jonathan Brandt, Zhe Lin, Lubomir Bourdev, and Thomas S. Huang. Interactive facial feature localization. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part III*, ECCV'12, pages 679–692, Berlin, Heidelberg, 2012. Springer-Verlag.

[90] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. *CoRR*, abs/1609.04802, 2016.

[91] D. Lee, H. Park, and C. D. Yoo. Face alignment using cascade gaussian process regression trees. In *CVPR*, pages 4204–4212, June 2015.

[92] Donghoon Lee, Hyunsin Park, and Chang D. Yoo. Face alignment using cascade gaussian process regression trees. June 2015.

[93] Gil Levi and Tal Hassner. Age and gender classification using convolutional neural networks. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) workshops*, June 2015.

[94] Lin Liang, Rong Xiao, Fang Wen, and Jian Sun 0001. Face alignment via component-based discriminative search. In David A. Forsyth, Philip H. S. Torr, and Andrew Zisserman, editors, *ECCV (2)*, volume 5303 of *Lecture Notes in Computer Science*, pages 72–85. Springer, 2008.

[95] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *CoRR*, abs/1708.02002, 2017.

[96] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing.

[97] Feng Liu, Dan Zeng, Qijun Zhao, and Xiaoming Liu. *Joint Face Alignment and 3D Face Reconstruction*, pages 545–560. Springer International Publishing, Cham, 2016.

[98] Yaojie Liu, Amin Jourabloo, William Ren, and Xiaoming Liu. Dense face alignment. In *In Proceeding of International Conference on Computer Vision Workshops*, Venice, Italy, October 2017.

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

[100] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.

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

[102] Jiangjing Lv, Xiaohu Shao, Junliang Xing, Cheng Cheng, and Xi Zhou. A deep regression architecture with two-stage re-initialization for high performance facial landmark detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[103] Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, and Zhen Wang. Multi-class generative adversarial networks with the L2 loss function. *CoRR*, abs/1611.04076, 2016.

[104] Iain Matthews and Simon Baker. Active appearance models revisited. *Int. J. Comput. Vision*, 60(2):135–164, November 2004.

[105] K. Messer, J. Matas, J. Kittler, and K. Jonsson. Xm2vtsdb: The extended m2vts database. In *In Second International Conference on Audio and Video-based Biometric Person Authentication*, pages 72–77, 1999.

[106] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *CoRR*, abs/1802.05957, 2018.

[107] Aaron Nech and Ira Kemelmacher-Shlizerman. Level playing field for million scale face recognition. *CoRR*, abs/1705.00393, 2017.

[108] Alejandro Newell, Kaiyu Yang, and Jia Deng. *Stacked Hourglass Networks for Human Pose Estimation*, pages 483–499. Springer International Publishing, Cham, 2016.

[109] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. *arXiv preprint arXiv:1505.04366*, 2015.

[110] A. J. O'Toole, T. Price, T. Vetter, J. C. Bartlett, and V. Blanz. 3d shape and 2d surface textures of human faces: The role of "averages" in attractiveness and age. *Image and Vision Computing*, 18(1):9–19, 1999.

[111] S. Ramanathan, B. Narayanan, and R. Chellappa. Computational methods for modeling facial aging: A survey. *Journal of Visual Languages & Computing*, 20(3):131–144, 2009.

[112] R. Ranjan, A. Bansal, J. Zheng, H. Xu, J. Gleason, B. Lu, A. Nanduri, J. Chen, C. D. Castillo, and R. Chellappa. A fast and accurate system for face detection, identification, and verification. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 1(2):82–96, April 2019.

[113] R. Ranjan, S. Sankaranarayanan, C. D. Castillo, and R. Chellappa. An all-in-one convolutional neural network for face analysis. In *2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017)*, pages 17–24, May 2017.

[114] Rajeev Ranjan, Vishal M. Patel, and Rama Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *CoRR*, abs/1603.01249, 2016.

[115] Shaoqing Ren, Xudong Cao, Yichen Wei, and Jian Sun. Face alignment at 3000 FPS via regressing local binary features. In *2014 IEEE Conference on*

Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014, pages 1685–1692, 2014.

[116] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[117] Martin Riedmiller and Heinrich Braun. A direct adaptive method for faster backpropagation learning: The rprop algorithm. In *IEEE INTERNATIONAL CONFERENCE ON NEURAL NETWORKS*, pages 586–591, 1993.

[118] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[119] J. Roth, Y. Tong, and X. Liu. Adaptive 3d face reconstruction from unconstrained photo collections. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4197–4206, June 2016.

[120] R. Rothe, R. Timofte, and L. V. Gool. Dex: Deep expectation of apparent age from a single image. In *ICCV, ChaLearn Looking at People workshop*, December 2015.

[121] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 2015.

[122] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *2013 IEEE International Conference on Computer Vision Workshops*, pages 397–403, Dec 2013.

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

[124] Jason Saragih. Principal regression analysis. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*, pages 2881–2888, 2011.

[125] Jason M. Saragih, Simon Lucey, and Jeffrey F. Cohn. Face alignment through subspace constrained mean-shifts. In *ICCV*, pages 1034–1041. IEEE, 2009.

[126] Jason M. Saragih, Simon Lucey, and Jeffrey F. Cohn. Deformable model fitting by regularized landmark mean-shift. *Int. J. Comput. Vision*, 91(2):200–215, January 2011.

[127] Patrick Sauer, Tim Cootes, and Chris Taylor. Accurate regression procedures for active appearance models. In *In BMVC*, 2011.

[128] A. Saxena, S. Sharma, and V. K. Chaurasiya. Neural network based human age-group estimation in curvelet domain. *Procedia Computer Science*, 54:781–789, 2015.

[129] Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *International Conference on Learning Representations (ICLR 2014)*. CBLS, April 2014.

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

[131] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[132] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019.

[133] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In *CVPR*, pages 3476–3483, June 2013.

[134] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep convolutional network cascade for facial point detection. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '13, pages 3476–3483, Washington, DC, USA, 2013. IEEE Computer Society.

[135] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261, 2016.

[136] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[137] Graham W. Taylor, Rob Fergus, Yann LeCun, and Christoph Bregler. Convolutional learning of spatio-temporal features. In *Proceedings of the 11th European Conference on Computer Vision: Part VI*, ECCV'10, pages 140–153, Berlin, Heidelberg, 2010. Springer-Verlag.

[138] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. Face2Face: Real-time Face Capture and Reenactment of RGB Videos. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2016.

[139] P. Thukral, K. Mitra, and R. Chellappa. A hierarchical approach for human age estimation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1529–1532. IEEE, 2012.

[140] Philip Tresadern, Patrick Sauer, and Tim Cootes. Additive update predictors in active appearance models. In *Proceedings of the British Machine Vision Conference*, pages 91.1–91.12. BMVA Press, 2010. doi:10.5244/C.24.91.

[141] G. Trigeorgis, P. Snape, M. A. Nicolaou, E. Antonakos, and S. Zafeiriou. Mnemonic descent method: A recurrent process applied for end-to-end face alignment. In *CVPR*, Las Vegas, NV, USA, June 2016.

[142] P. Turaga, S. Biswas, and R. Chellappa. The role of geometry in age estimation. In *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 946–949. IEEE, 2010.

[143] G. Tzimiropoulos and M. Pantic. Gauss-newton deformable part models for face alignment in-the-wild. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1851–1858, June 2014.

[144] Georgios Tzimiropoulos. Project-out cascaded regression with an application to face alignment. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[145] Roberto Valle, José Miguel Buenaposada, Antonio Valdés, and Luis Baumela. *Head-Pose Estimation In-the-Wild Using a Random Forest*, pages 24–33. Springer International Publishing, Cham, 2016.

[146] M. F. Valstar, T. Almaev, J. M. Girard, G. McKeown, M. Mehu, L. Yin, M. Pantic, and J. F. Cohn. Fera 2015 - second facial expression recognition and analysis challenge. In *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, volume 06, pages 1–8, May 2015.

[147] P. Viola and M. J. Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[148] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Chen Change Loy, Yu Qiao, and Xiaoou Tang. ESRGAN: enhanced super-resolution generative adversarial networks. *CoRR*, abs/1809.00219, 2018.

[149] Peter Welinder and Pietro Perona. P.: Cascaded pose regression. In *In: IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[150] T. Wu, P. Turaga, and R. Chellappa. Age estimation and face verification across aging using landmarks. *IEEE Transactions on Information Forensics and Security*, 7(6):1780–1788, 2012.

[151] Wayne Wu, Chen Qian, Shuo Yang, Quan Wang, Yici Cai, and Qiang Zhou. Look at boundary: A boundary-aware face alignment algorithm. In *CVPR*, 2018.

[152] Xiang Wu, Ran He, and Zhenan Sun. A lightened CNN for deep face representation. *CoRR*, abs/1511.02683, 2015.

[153] Y. Wu and Q. Ji. Robust facial landmark detection under significant head poses and occlusion. In *ICCV*, pages 3658–3666, Dec 2015.

[154] Yue Wu, Chao Gou, and Qiang Ji. Simultaneous facial landmark detection, pose and deformation estimation under facial occlusion. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[155] X. Xiong and F. De la Torre. Global supervised descent method. In *CVPR*, 2015.

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

[157] Junjie Yan, Zhen Lei, Dong Yi, and Stan Z. Li. Learn to combine multiple hypotheses for accurate face alignment. In *Proceedings of the 2013 IEEE International Conference on Computer Vision Workshops*, ICCVW '13, pages 392–396, Washington, DC, USA, 2013. IEEE Computer Society.

[158] H. Yang, X. He, X. Jia, and I. Patras. Robust face alignment under occlusion via regional predictive power estimation. *IEEE Transactions on Image Processing*, 24(8):2393–2403, Aug 2015.

[159] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

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

[161] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z. Li. Learning face representation from scratch. *CoRR*, abs/1411.7923, 2014.

[162] Juha Ylioinas, Abdenour Hadid, Xiaopeng Hong, and Matti Pietikäinen. Age estimation using local binary pattern kernel density estimate. In Alfredo Petrosino, editor, *Image Analysis and Processing – ICIAP 2013*, volume 8156 of

Lecture Notes in Computer Science, pages 141–150. Springer Berlin Heidelberg, 2013.

[163] Xiang Yu, Feng Zhou, and Manmohan Chandraker. Deep deformation network for object landmark localization. *CoRR*, abs/1605.01014, 2016.

[164] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013.

[165] Jie Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Occlusion-free face alignment: Deep regression networks coupled with de-corrupt autoencoders. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[166] Jie Zhang, Shiguang Shan, Meina Kan, and Xilin Chen. Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision ECCV 2014*, volume 8690 of *Lecture Notes in Computer Science*, pages 1–16. Springer International Publishing, 2014.

[167] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, Oct 2016.

[168] Ning Zhang, Manohar Paluri, Yaniv Taigman, Rob Fergus, and Lubomir D. Bourdev. Beyond frontal faces: Improving person recognition using multiple cues. *CoRR*, abs/1501.05703, 2015.

[169] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. *Facial Landmark Detection by Deep Multi-task Learning*, pages 94–108. Springer International Publishing, Cham, 2014.

[170] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Facial landmark detection by deep multi-task learning. In *ECCV*, pages 94–108, 2014.

[171] Junbo Jake Zhao, Michaël Mathieu, and Yann LeCun. Energy-based generative adversarial network. *CoRR*, abs/1609.03126, 2016.

[172] Shizhan Zhu, Cheng Li, Chen Change Loy, and Xiaoou Tang. Face alignment by coarse-to-fine shape searching. June 2015.

[173] Shizhan Zhu, Cheng Li, Chen Change Loy, and Xiaoou Tang. Towards arbitrary-view face alignment by recommendation trees. *CoRR*, abs/1511.06627, 2015.

[174] Shizhan Zhu, Cheng Li, Chen-Change Loy, and Xiaoou Tang. Unconstrained face alignment via cascaded compositional learning. In *CVPR*, June 2016.

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

[176] Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Z. Li. Face alignment across large poses: A 3d solution. *CoRR*, abs/1511.07212, 2015.