Deep learning for real world face alignment

Bulat Adrian

School of Computer Science
University of Nottingham

This dissertation is submitted for the degree of
Doctor of Philosophy

February 2019

Acknowledgements

Dear reader, before letting your eyes move forward, delving into the work that occupied my time and mind during the last three years I would like to first and foremost express my deepest gratitude to my supervisor, **Dr. Georgios Tzimiropoulos**, who kept and brought hope when I had lost mine. Thanking him for his "pixel-wise" attention to details and inspiring guidance. This would have not been possible without your help and support! THANK YOU!

For their helpful comments, time and effort I would like to thank my examiners, Prof. Thomas Gärtner and Prof. Ioannis Patras. Your suggestions and feedback where monumental in shaping the final version of this thesis.

Special thanks go to Dr. Robert Gabriel Lupu who encouraged and helped me in my endeavor toward pursuing my PhD. Thanks to all my professors that shaped me and carefully passed their knowledge and their advices during my reckless undergrad years, specially to Dr. Andrei Stan, Dr. Calin Mircea Monor and Prof. Florina Ungureanu.

While remaining anonymous, I thank to all my reviewers for their constructive comments.

Thanks to the amazing team that I was lucky to found here. It was both a pleasure and an honor to be part of the Computer Vision Lab. To all of my current and past lab colleagues from B86: Aaron, Andreas, Dotty, Farhad, Jing, Themos, Kike, Keerthy and Joy, that learned to support me and my bad jokes, brought a smile or an useful advice, thank you. All of you will have a special place in my hearth and mind.

Thanks to all my friend, from childhood, high school, or from the crazy years spent in T19, it will always be a joy to have a chat with you. Thanks to all my friends from MSP, we were an amazing team.

Special thanks to my parents and my relatives:
Dragii mei, vă mulțumesc din suflet pentru toate sacrificiile pe care le-ați făcut, pentru momentele frumoase pe care mi le-ați dăruit, pentru suport, răbdare și înțelegere! Va mulțumesc pentru tot.

Abstract

Face alignment is one of the fundamental steps in a vast number of tasks of high economical and social value, ranging from security to health and entertainment. Despite the attention received from the community for more than 2 decades and the success of cascaded regression based approaches, many challenges were yet to be solved, such as the case of near-profile poses and low resolution faces.

In this thesis, we successfully address a series of such challenges in the area of face alignment and super-resolution, significantly pushing the state-of-the-art by proposing novel deep learning-based architectures specially tailored for fine grained recognition tasks. In summary, we address the following problems: (I) fitting faces found in large poses (Chapter 3), (II) in both 2D and 3D space (Chapter 4), creating in the process (III) the largest "in-the-wild" large pose 3D face alignment dataset (Chapter 4). While the case of high resolution faces was actively explored in the past, in this thesis we systematically study and address a new challenge: that of (IV) fitting landmarks in very low resolution faces (Chapter 6). While deep learning based approaches achieved remarkable results on a wide variety of tasks, they are usually slow having high computational requirements. As such, in Chapter 5, we propose (V) a novel residual block carefully crafted for binarized neural networks that significantly improves the speed, due to the use of binary operations for both the weights and the activations, while maintaining a similar or competitive accuracy.

The results presented through out this thesis set the new state-of-the-art on both 2D & 3D face alignment and face super-resolution.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

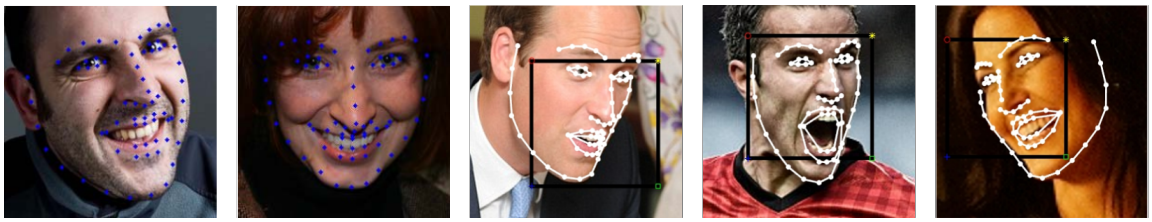
1.1 Introduction

Analysing, localising and understanding humans is one of the long-standing problems in Computer Vision, yet its roots significantly precede the domain itself. This set of human-centric problems are usually grouped in a sub-domain of its own, called *human sensing*. Out of this amalgam of problems one of them, in particular, stands out that of *facial image analysis*, having a large number of use cases of high economical and cultural value. While not limited to them alone, in the following, we will categorize its application in four broad domains: entertainment, health, marketing and security.

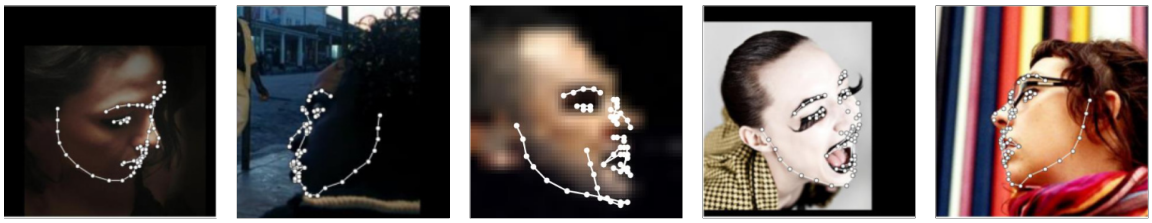
In **security**, while fingerprint authentication is the long term standard, recently, more and more products, ranging from laptops to smartphones, are adding face recognition based authentication due to its easy of use. Its application goes however beyond this and such technologies could assist law enforcement agencies in identifying wanted individuals, helping prevent crimes or improve traffic flow at border check points. In **marketing**, there are a plethora of software solutions that help companies to automatically analyse the impact and efficiency of their advertising campaigns, their product branding or the placement of items in the store. In the **health** area, its application varies from emotion recognition, helping to identify mental disorders (e.g. depression), to automatic pain assessment or assisted psychological problems diagnostic. Finally, in the **entertainment** industry, face analysis-based technologies are ubiquitous and are present at the hearth of the currently most popular entertainment applications (facebook, skype, snapchat etc.) where facial analysis is used to automatically tag/identify persons, enhance or augment the photos, generate personalised avatars and many more. While this is by no means an exhaustive list of face analysis applications, it sheds light into its importance and its numerous application areas.

A fundamental step in facial image analysis and by extension, in all the above mentioned applications, is the task of facial landmark localisation also called face alignment, the aim of which is the localisation of a pre-defined set of points of interest, also called fiducial points. Please see Chapter 1.2 for more details regarding the problem setup and formulation.

Given the importance of facial landmark localisation, a numerous of outstanding ideas were proposed during the past years with the Active Appearance Model [21] and Cascaded Regression [29] as two of the most important and influential checkpoints. These methods usually perform well when the face is well illuminated and in a near-frontal pose. More recently, deep learning based methods showed promising results on fine-grained recognition tasks such as the related task of human pose estimation[8, 69, 75]. These results inspired us on our endeavour on the problem of face alignment. Through this thesis we make reference to a series of components (ie. Convolutional layers, Pooling etc) related to the area of artificial neural networks. To this end we encourage the reader to check the reference book on deep learning [33] for a better understanding.



(a) State-of-the-art results at the beginning of this PhD (2015). The examples illustrate some of the most challenging cases solved at the time. Images taken from [103, 102].



(b) Results for some very challenging images obtained using the methods proposed in Chapter 4 and 6.

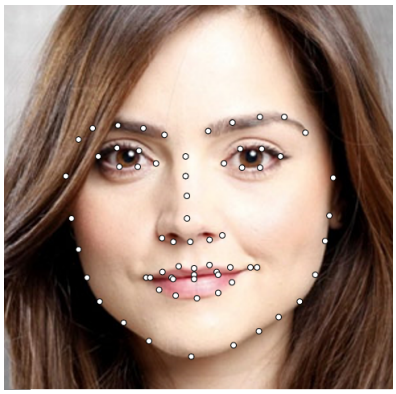
Fig. 1.1 Advancement of the state-of-the-art as a result of the methods proposed in this thesis. Notice that the results of our work are more accurate, can handle extreme poses of faces captured in challenging illumination conditions and can work even for very low resolution images.

Despite the success achieved by Cascaded Regression-based approaches, there were still a large number of open research questions. While this is detailed in Section 1.3, it is important to mention here that in this thesis we seek to address and solve the following problems/cases: (a) detect the facial landmarks (fits) for faces found in large and extreme poses, exhibiting extreme illumination conditions in both 2D and (b) 3D settings. (c) Provide a large scale "in-the-wild" 3D facial landmark dataset. (d) Localise the landmarks on low quality images containing faces at very low resolution. Equally importantly, achieving all of this while (e) providing a real-time or near real-time performance. Overall, we will try to answer the following question: "How far are we from solving the face alignment problem?". Note, that herein, by solving we refer to achieving human or near-human accuracy on the available datasets (i.e. the detections fall near the noise of human annotators).

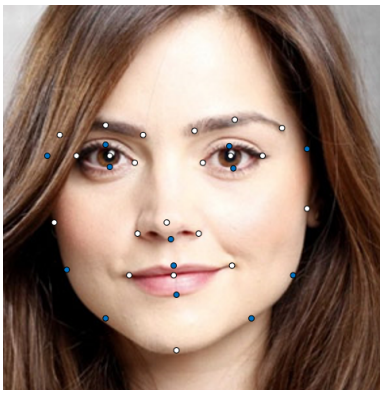
1.2 Problem definition

Face alignment (i.e. facial point localisation) is the process of estimating the configuration of a given set of fiducial points from a single, usually monocular, facial image. The facial image itself is a result of a detection process that establishes a region of interest (ROI) as follows: The input image is passed to a face detector that returns a bounding box for each corresponding face present in the image, which in turn is used to compute an approximative scale and location for each face. Finally, the facial image (i.e. ROI) is defined as a function of the face scale and location.

While the number of targeted points can vary from 5 to more than 100, most of the recent works have agreed on a 68 points configuration (see Fig. 1.2), which is used for the experiments reported through out this work with the exception of Chapter 3 where a 21 and 31-based point configuration will be used.



(a) 68-points based mark-up.



(b) 21 and 31-based points configuration. The additional 10 points for the 31-configuration are marked with blue.

Fig. 1.2 The mark-up points used in the 68-points configuration (a) and 21/31 respectively (b).

1.3 Main Challenges & Contributions

Being a long-standing problem in Computer Vision research, a multitude of approaches with various degree of success have been proposed so far to solve the face alignment task. With the advent of cascaded regression [29] and its application to face alignment [17, 107, 102], the state-of-the-art (prior to the advent of Deep Learning) was considered to have reached a satisfactory level of performance for frontal faces, including faces with relatively difficult illumination, expression and occlusion. However, despite past efforts, a series of problems are still present making the current methods unsuitable for an "in-the-wild" setting. In the following, we will briefly explain each of them. Later on, in this thesis, we will present novel approaches that advance the state-of-the-art and to some extent solve the current challenges.

1.3.1 Large poses

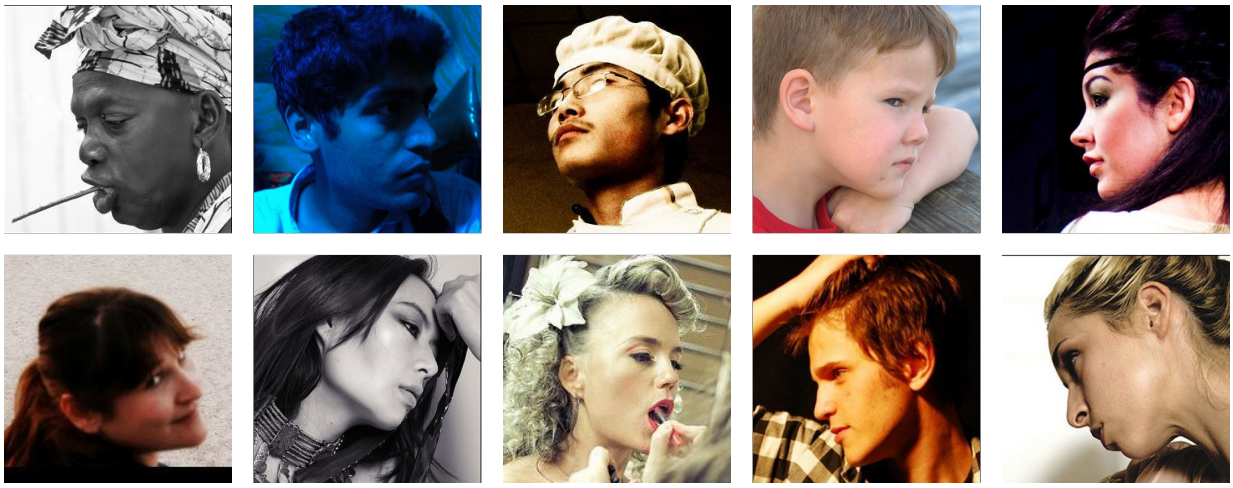


Fig. 1.3 Examples of faces found in large poses.

We consider a face to be in a large pose if the head orientation (in terms of yaw, pitch and roll) deviates significantly (usually more than 30°) from its frontal position (see Fig. 1.3).

Despite recent efforts, the problem of face alignment under very large pose variation (up to 90^o) has received little attention, with most of the regression-based methods being unable to cope well with faces found in large poses. In addition, it is important to notice that a well-performing method should be also able to deal well with the large variety of facial expressions and occlusions (both object and self-occlusion). A few examples of such challenging cases can be found in Fig. 1.3.

Main Contribution. To address this, in Chapter 3 we propose a novel deep learning based method capable on dealing with faces found in arbitrary poses. The proposed method halved the error on the most challenging datasets available.

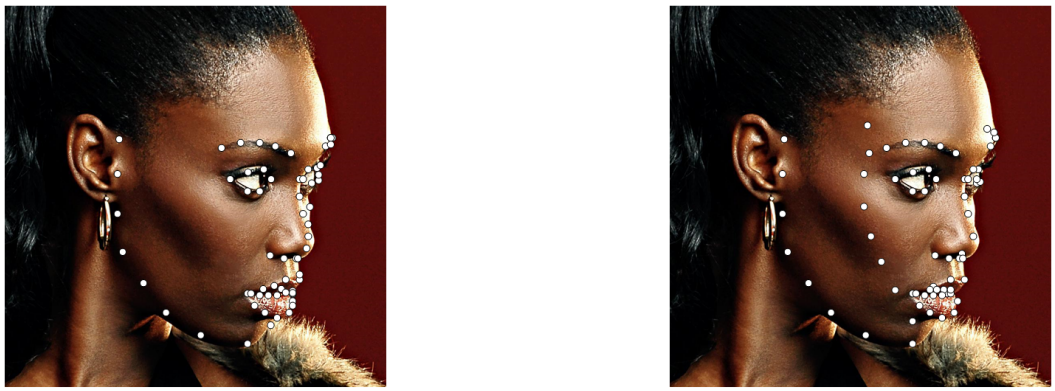
1.3.2 2D vs 3D

In addition to the large facial poses problem, most of the prior work treated the face as a 2D object. This assumption is valid only if the face is found in a frontal pose and is planar. As the face orientation changes, this assumption does not hold any longer and the annotated landmarks will lose correspondence. Fig. 1.4 illustrates the difference between the 2D and 3D annotations.

Main Contribution. In Section 4, we introduce a new method that detects the points in 3D space, predicting also the z coordinate(i.e. depth). The proposed approach significantly outperforms previous methods setting a new state-of-the-art result.

1.3.3 Sensitivity to initialisation

While the regression-based approaches achieved good performance, they also tended to be sensitive to initialisation, which is a direct consequence of the way they are initialised. Usually



(a) 2D-perspective mark-up. (b) 3D-perspective mark-up

Fig. 1.4 The mark-up points used in the 68-points configuration for 2D (a) and 3D respectively (b). Notice how the 2D-based annotation losses the correct spatial correspondence for larger poses.

this is done by placing a mean shape enclosed by the detected face bounding box. Due to this, the performance is directly linked to the face detector used at train time, with the performance quickly degrading if a different one is used at test time.

Main Contribution. To alleviate this, in Section 3 we describe a two-stage CNN-based approach that significantly reduces the dependency on the face detector used, with the first stage having also the role of making the network location-invariant.

1.3.4 Training data scarcity

Prior to the LS3D-W dataset introduced in Chapter 4, most of face alignment datasets were either small (e.g. 300-W [82]), or contained errors in the annotations (e.g. 300-VW [87], AFLW-2000 [132]), or mainly contained faces found in frontal poses (e.g. 300-W [82], 300-VW [87]), or they were synthetically warped to large poses (300-W-LP [132]).

For a complete review of the available face alignment datasets, including the one introduced in this work see Section 2.3. For details regarding the newly introduced dataset that addresses most of the aforementioned issues see Chapter 4.

1.3.5 Performance considerations

As we will see in Chapter 3 and 4, methods based on Convolutional Neural Networks demonstrate results of remarkable accuracy even in the most challenging conditions. However, such methods are computationally expensive, requiring one ore more high-end GPUs, thus making the method unsuitable for real-time or mobile applications. Typically, a simple decrease in the model size will result in a noticeable performance drop, especially for the cases found at the edge of the distribution and as such is suboptimal.

Main Contribution. To address this, in Chapter 5 we propose a novel residual-block, specially designed for binary networks and optimised for fine-grained tasks. Not only is the

proposed method up to 53× faster on a CPU, but it also outperforms previous top performing methods, such as the one presented in Chapter 3.

1.3.6 Low resolution faces

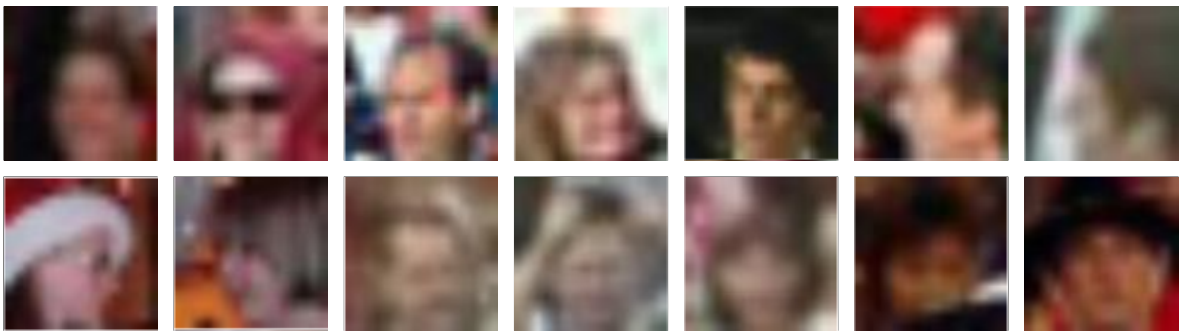


Fig. 1.5 Examples of natural low resolution faces.

When evaluated on low and very low resolution images, with a face resolution as low as 10×10px, the current top performing face alignment methods suffer an extreme degradation in performance as shown in Chapter 4. Examples of such naturally occurring images can be found in Fig. 1.5.

Main Contribution. To address this, in Chapter 6, we propose a novel method that jointly learns to perform face super-resolution and alignment simultaneously, with both tasks working symbiotically, improving each other. The results produced significantly reduce the performance gap between high and low resolution images.

1.3.7 Contributions

In summary, our main contributions are:

- We are the first to apply heatmap regression in the context of face alignment (Chapter 3).
- A novel, yet simple, CNN architecture for large pose face alignment which we call Convolutional Aggregation of Local Evidence (CALE). CALE by-passes the requirement for accurate face detection by firstly using a CNN detector to perform facial landmark detection. Next, CALE aggregates the local evidence for each facial landmark through joint CNN regression of the confidence scores, in order to refine the landmarks' location. The proposed system achieves large performance improvement over the state-of-the-art (Chapter 3).
- We construct, for the first time, a very strong baseline by combining a state-of-the-art architecture for landmark localization with a state-of-the-art residual block and train it on a very large yet synthetically expanded 2D facial landmark dataset. Then, we evaluate it on all other 2D datasets (230,000 images), investigating how far are we from solving 2D and 3D face alignment (Chapter 4).

- In order to overcome the scarcity of 3D face alignment datasets, we propose a guided-by-2D landmarks CNN which converts 2D annotations to 3D and use it to create LS3D-W, the largest and most challenging 3D facial landmark dataset to date (230,000 images), obtained from unifying almost all existing datasets to date (Chapter 4).
- We are the first to study the effect of binarization on state-of-the-art CNN architectures for the problem of face alignment. To this end, we exhaustively evaluate various design choices, and identify performance bottlenecks, describing multiple orthogonal ways to boost performance (Chapter 5).
- We propose Super-FAN: the very first end-to-end system that addresses face super-resolution and alignment simultaneously, via integrating a sub-network for facial landmark localization through heatmap regression into a GAN-based super-resolution network, and incorporating a novel heatmap loss (Chapter 6).
- Quantitatively, we report, for the first time, results across the whole spectrum of facial poses on the LS3D-W dataset, and show large improvement over the state-of-the-art on both super-resolution and face alignment. Qualitatively, we show, for the first time, good visual results on real-world low resolution facial images taken from the WiderFace dataset (Chapter 6).

1.4 Publications

The research presented in this thesis has been published in the following conferences:

- A. Bulat and G. Tzimiropoulos. Convolutional aggregation of local evidence for large pose face alignment. In *British Machine Vision Conference (BMVC)*, 2016
- A. Bulat and G. 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 (ICCV)*, 2017
- A. Bulat and G. Tzimiropoulos. Binarized convolutional landmark localizers for human pose estimation and face alignment with limited resources. In *International Conference on Computer Vision (ICCV)*, 2017 (ORAL)
- A. Bulat and G. Tzimiropoulos. Super-fan: Integrated facial landmark localization and super-resolution of real-world low resolution faces in arbitrary poses with gans. In *IEEE Conference on ComputerVision and Pattern Recognition (CVPR)*, 2018 (SPOTLIGHT)
- A. Bulat and Y. Tzimiropoulos. Hierarchical binary cnns for landmark localization with limited resources. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018 (Best of ICCV17 SI)

In addition to them, while not being directly included in this thesis, the following publications were monumental toward achieving the results described in this work:

- A. Bulat and G. Tzimiropoulos. Human pose estimation via convolutional part heatmap regression. In *European Conference on Computer Vision*, pages 717–732. Springer, 2016
- A. Bulat and G. Tzimiropoulos. Two-stage convolutional part heatmap regression for the 1st 3d face alignment in the wild (3dfaw) challenge. In *European Conference on Computer Vision (ECCV)*, pages 616–624. Springer, 2016 (Challenge Winner)

Chapter 2

Literature review

This chapter reviews related work on human sensing focusing on the more recent deep learning based approaches. In particular, firstly, we start by offering an overview of the more generic problem of CNN heatmap regression (Section 2.1). Secondly, we delve into the face alignment problem (both 2D & 3D) reviewing the existing methods and datasets alongside their current limitations (Section 2.2). Then, we analyse the problem at hand from a performance perspective, reviewing the recent work on network quantization (Section 2.4). Finally, in Section 2.5 we review related work in image and face super-resolution.

2.1 CNN heatmap regression

This section offers a review of the related work on heatmap-based regression using CNNs.

Recently, methods based on CNNs have been shown to produce state-of-the-art results for many Computer Vision tasks like image recognition [88], object detection [32] and semantic image segmentation [66]. In the context of landmark localisation, it is natural to formulate the problem as a regression one in which CNN features are regressed in order to provide a joint prediction of the landmarks, see for example recent works on human pose estimation [100, 74, 5, 8]. A notable development has been the replacement of the standard L2 loss between the predicted and ground truth part locations with the so-called confidence map regression which defines an L2 loss between predicted and ground truth confidence maps encoded as 2D Gaussians centred at the part locations [99, 74]. As a mapping from CNN features to part locations might be difficult to learn in one shot, regression-based methods can be also applied sequentially (i.e. in a cascaded manner) [100, 19, 105]. In the context of human pose estimation, [74] performs confidence map regression (based on a L2 loss). Then, the pre-confidence maps are used as input to a subsequent regression network, however such maps are too localised providing small spatial support. In order to improve performance, regression methods applied in a sequential, cascaded fashion have been recently proposed in [19, 105]. In particular, [105] has recently reported outstanding results on both LSP [52] and MPII [1] data sets using a six-stage CNN cascade.

Notably, all the aforementioned methods were developed prior to the advent of residual learning [38]. Very recently, residual learning was applied for the problem of human pose estimation in [45] and [69]. Residual learning was used for part detection in the system of [45]. The "stacked hourglass network" of [69] elegantly extends FCN [66] and deconvolution nets [121] within residual learning, also allowing for a more sophisticated and heavy processing during top-down processing.

In the context of face alignment, while CNNs have been applied recently to it [94, 124], we are the first to explore heatmap-based regression for 2D & 3D face alignment. The work in [124] proposes to combine facial landmark localisation with attribute classification through multi-task learning. One limitation of above methods is that they can detect 5 landmarks only. Very recent work includes [132, 54] and [101] which extends [108] within recurrent neural networks. We will further detail this in section 2.2.

2.2 Face Alignment

This section reviews related work on 2D & 3D face alignment.

2.2.1 2D face alignment

Prior to the advent of Deep Learning, methods based on cascaded regression have emerged as the state-of-the-art in 2D face alignment, see for example [17, 108, 129, 102]. Most commonly, such methods rely on hand-crafted features, are sensitive to face detection initialisation [111], might require a cascade with many steps, and most notably have been shown to work well mainly for frontal datasets like LFPW [6], Helen [60] and 300-W [83] in which most of the landmarks are visible. On the contrary, our method, presented in Chapter 3, does not rely on accurate face detection, uses a single regression step and can cope well with arbitrary poses and severe self-occlusion. This method will be further improved and expanded in Chapter 4 to 3D face alignment. Notably, the idea of aggregating local evidence for facial landmark localisation has been explored within methods based on the so-called Constrained Local Model (CLM) [26, 86, 3, 4]. Note that all CLM-based methods use hand-crafted features and have been shown to be largely outperformed by cascaded regression methods. On the contrary, we show that the proposed methods largely outperforms all prior work on 2D and 3D large pose face alignment.

2.2.2 Large pose face alignment

State-of-the-art methods for large pose face alignment include techniques that attempt to perform face alignment by fitting a 3D Morphable Model (3DMM) to a 2D facial image [53, 54, 132]. The work in [53] aligns faces using a sparse 3D point distribution model the parameters of which along with the projection matrix are estimated by cascaded regression. Notably, [53] introduces AFLW-PIFA, a challenging dataset for large pose unconstrained face alignment. The work

in [54] extends [53] by fitting a dense 3DMM using a cascade of CNNs. A similar approach to [54] has been also proposed in [132]. Besides 3DMM-based approaches, the work in [132] performs large pose 2D face alignment based on compositional cascaded learning, a novel way to perform model averaging within cascaded regression. Despite its elegant formulation, [132] completely avoids regressing non-visible landmarks and suffers from many of the problems common in all cascaded regression techniques (please see above). Compared to [53, 54, 132], our system by-passes the burden of fitting a 3D model and compared to [132], our method avoids the limitations of cascaded regression. On AFLW-PIFA, the system proposed in Chapter 3 reduces the error reported in [53, 54, 132] by more than 50% ([132] does not report performance on this dataset). This performance is further improved in Chapter 4 where we set the new state-of-the-art on 300-W, AFLW2000-3D, Menpo and the newly introduced dataset from the present work, LS3D-W.

2.2.3 3D face alignment

In the case of 2D face alignment, the face is treated as a 2D planar object, which causes the annotated points to lose correspondence as the face departs from a frontal pose. To alleviate this, recently, 3D face alignment was proposed as a solution. Traditionally, for both 3D and 2D large pose face alignment a 3DMM was fitted to a given input image [132, 54]. Given the scarcity of available 3D data, an important milestone was the the release of the Workshop on 3D Face Alignment in the Wild(3DFAW) Dataset & Challenge [49]. We participated in this challenge and we were the challenge winners [9]. To this end, we built on top of the network architecture we introduced in [8] expanding it for the 3D case by appending an auxiliary regression network, based on a ResNet-202 [39], that given the predicted 2D landmarks and the input image estimates the depth for each landmark. In this thesis, we further improve on top of our work in [9] in Chapter 4, significantly outperforming the state-of-the-art method from [132]. At the same time we introduce the largest 3D face alignment dataset.

2.2.4 Dataset expansion by transferring landmark annotations

There are a few works that have attempted to unify facial alignment datasets by transferring landmark annotations, typically through exploiting common landmarks across datasets [128, 90, 122]. Such methods have been primarily shown to be successful when landmarks are transferred from more challenging to less challenging images, for example in [128] the target dataset is LFW [43] or [90] provides annotations only for the relatively easy images of AFLW [57]. Hence, the community primarily relies on the unification performed manually by the 300-W challenge [82] which contains less than 5,000 near frontal images annotated from a 2D perspective.

Using 300-W-LP [132] as a basis, Chapter 4 presents the first attempt to provide 3D annotations for *all other* datasets, namely AFLW-2000 [132] (2,000 images), 300-W test set [81] (600 images), 300-VW [87] (218,595 frames), and Menpo training set (9,000 images). To this end,

we propose in Chapter 4 a guided-by-2D landmarks CNN which converts 2D annotations to 3D and unifies all aforementioned datasets.

2.3 Face Alignment Datasets

In this Section, an in-depth description of existing 2D and 3D datasets is provided. We note that the 3D annotations preserve correspondence across pose as opposed to the 2D ones and, in general, they should be preferred. We emphasize that the 3D annotations are actually the 2D projections of the 3D facial landmark coordinates but for simplicity we will just call them 3D and refer to the actual 3D ones explictly in the text.

Dataset	Size	pose	annot.	synt.
300-W	4,000	$[-45°, 45°]$	2D	No
300-W-LP-2D	61,225	$[-90°, 90°]$	2D	Yes
300-W-LP-3D	61,225	$[-90°, 90°]$	3D	Yes
AFLW	25,993	$[-90°, 90°]$	2D	No
AFLW-PIFA	5,200	$[-90°, 90°]$	3D	No
AFLW2000-3D	2,000	$[-90°, 90°]$	3D	No
Menpo	9,000	$[-90°, 90°]$	2D&3D	No
300-VW	218,595	$[-45°, 45°]$	2D	No
LS3D-W (ours)	230,000	$[-90°, 90°]$	3D	No
LS3D-W-balanced (ours)	7,800	$[-90°, 90°]$	3D	No

Table 2.1 Summary of the most popular face alignment datasets and their main characteristics.

2.3.1 2D datasets

300-W. The 300-W dataset was introduced in [82] by combining the images from LFPW[6], AFW[133], HELEN[60] and XM2VTS[68] and re-annotating them in a consistent manner using the 68 points based configuration depicted in Fig. 1.2a. In addition to this another 135 images exhibiting difficult poses and expression were added, totalling 3000 images.
300-W test set. The 300-W test set consists of the 600 images used for the evaluation purposes of the 300-W Challenge [81]. The images are split in two categories: *Indoor* and *Outdoor*. All images were annotated with the same 68 2D landmarks as the ones used in the 300-W data set.
300-VW. 300-VW[87] is a large-scale face tracking dataset, containing 114 videos and in total 218,595 frames. From the total of 114 videos, 64 are used for testing and 50 for training. The test videos are further separated into three categories (A, B, and C) with the last one being the most challenging. It is worth noting that some videos (especially from category C) contain very low resolution/poor quality faces. Due to the semi-automatic annotation approach (see [87] for

more details), in some cases, the annotations for these videos are not so accurate (see Fig. 4.4). Another source of annotation error is caused by facial pose, i.e. large poses are also not accurately annotated (see Fig. 4.4).

Menpo. Menpo is a recently introduced dataset [119] containing landmark annotations for about 9,000 faces from FDDB [48] and ALFW. Frontal faces were annotated in terms of 68 landmarks using the same annotation policy as the one of 300-W but profile faces in terms of 39 different landmarks which are not in correspondence with the landmarks from the 68-point mark-up.

AFLW is a large-scale face alignment dataset that contains faces in various poses and expressions collected from Flickr. All 25,993 faces present in the dataset were annotated with up to 21 points.

2.3.2 3D datasets

Prior to the work presented in Chapter 4, the existing 3D face alignment datasets were either small or artificially rendered from frontal poses and as such were often unrealistic and inaccurate. Bellow, the current datasets are listed alongside the newly introduced LS3D-W dataset.

300-W-LP is a synthetically expanded dataset obtained by artificially rendering the faces from 300-W [82] into large poses (-90^0 to 90^0). While the dataset contains 61,225 images, there are only about 3,000 unique faces. Also, the images are affected by artefacts caused by the warping procedure. We included the entire dataset in our training set.

AFLW-PIFA is a subset of 5,200 grayscale images selected from AFLW and re-annotated with up to 34 points and occlusion labels [53]. The dataset has a balanced distribution of yaw angles (from -90° to 90°).

AFLW2000-3D. AFLW2000-3D [132] is a dataset constructed by re-annotating the first 2000 images from AFLW [57] using 68 3D landmarks in a consistent manner with the ones from 300-W-LP-3D. The faces of this dataset contain large-pose variations (yaw from -90^o to 90^o), with various expressions and illumination conditions. However, some annotations, especially for larger poses or occluded faces are not so accurate (see Fig. 4.7).

LS3D-W. LS3D-W is the dataset introduced in this work (Chapter 4), being the largest up-to-date 3D face alignment dataset containing more than 230,000 images found in arbitrary/natural conditions. Full details regarding its content and the method used to create it are described in Chapter 4.

LS3D-W balanced is a subset of the LS3D-W dataset, containing 7,200 images captured in-the-wild, in which each pose range ($[0^0 - 30^0], [30^0 - 60^0], [60^0 - 90^0]$) is equally represented (2,400 images each).

2.3.3 Other related datasets

In addition to the above mentioned face alignment datasets, for the work on joint super-resolution and face alignment presented in Chapter 6, we used a few additional datasets required for the task at hand that are described bellow.

Celeb-A is a large-scale facial attribute dataset containing 10,177 unique identities and 202,599

facial images in total. Most of the images are occlusion-free and in frontal or near-frontal poses. To avoid biasing the training set towards frontal poses, we only used a randomly selected subset of approx. 20,000 faces.

WiderFace is a face detection dataset containing 32,203 images with faces that exhibit a high degree of variability in pose, occlusion and quality. In order to assess the performance of our super-resolution method on in-the-wild, real-world images, we randomly selected 200 very low resolution, heavily blurred faces for qualitative evaluation.

2.4 Efficient Convolutional Neural Networks

Despite their unprecedented accuracy, most of the deep learning-based methods remain slow and are unsuitable for devices with limited computational resources (i.e. smartphones, FPGAs etc). To alleviate this, recently, a series of techniques were proposed. In this section an overview of such approaches is presented in the following two subsections.

2.4.1 Network quantization

Prior work [40] suggests that high precision parameters are not essential for obtaining top results for image classification. In light of this, [22, 62] propose 16- and 8-bit quantization, showing negligible performance drop on a few small datasets [58]. [126] proposes a technique which allocates different numbers of bits (1-2-6) for the network parameters, activations and gradients.

Binarization (i.e. the extreme case of quantization) was long considered to be impractical due to the destructive property of such a representation [22]. Recently [93] showed this not to be the case and that by quantizing to $\{-1, 1\}$ good results can be actually obtained. [23] introduces a new technique for training CNNs that uses binary weights for both forward and backward passes, however, the real parameters are still required during training. The work of [24] goes one step further and binarizes both parameters and activations. In this case multiplications can be replaced with elementary binary operations [24]. By estimating the binary weights with the help of a scaling factor, [79] is the first work to report good results on a large dataset (ImageNet). Notably, our method, presented in Chapter 5, makes use of the recent findings from [79] and [24] using the same way of quantizing the weights and replacing multiplications with bit-wise *xor* operations.

Our method (Chapter 5) differs from all aforementioned works in two key respects: (a) instead of focusing on image classification, we are the first to study neural network binarization in the context of a fine-grained computer vision task namely landmark localization (human pose estimation and facial alignment) by predicting a dense output (heatmaps) in a fully convolutional manner, and (b) instead of enhancing the results by improving the quantization method, we follow a completely different path, by enhancing the performance via proposing a novel architectural design for a hierarchical, parallel and multi-scale residual block.

2.4.2 Block design

Most of the work presented in this thesis uses a residual-based architecture and hence the starting point of our work is the *bottleneck* block described in [38, 39]. More recently, [106] explores the idea of increasing the cardinality of the residual block by splitting it into a series of c parallel (and much smaller so that the number of parameters remains roughly the same) sub-blocks with the same topology which behave as an ensemble. Beyond bottleneck layers, Szegedy et. al. [96] propose the inception block which introduces parallel paths with different receptive field sizes and various ways of lowering the number of parameters by factorizing convolutional layers with large filters into smaller ones. In a follow-up paper [95], the authors introduce a number of inception-residual architectures. The latter work is the most related one to the proposed method.

Our method, described in Chapter 5, is different from the aforementioned architectures in the following ways: we create a hierarchical, parallel and multi-scale structure that (a) increases the receptive field size inside the block and (b) improves gradient flow, (c) is specifically designed to have (almost) the same number of parameters as the original bottleneck, (d) our block does not contain 1×1 convolutions, and (e) our block is derived from the perspective of improving the performance and efficiency of binary networks.

2.5 Image and face resolution enhancement

2.5.1 Image super-resolution.

Early attempts on super-resolution using CNNs [30, 56] used standard L_p losses for training which result in blurry super-resolved images. To alleviate this, rather than using an MSE over pixels (between the super-resolved and the ground truth HR image), the authors of [51] proposed an MSE over feature maps, coined perceptual loss. Notably, in our method presented in Chapter 6, we also use a perceptual loss. More recently, in [61], the authors presented a GAN-based [34] approach which uses a discriminator to differentiate between the super-resolved and the original HR images and the perceptual loss. In [84], a patch-based texture loss is proposed to improve reconstruction quality.

Notice that all the aforementioned image super-resolution methods can be applied to all types of images and hence do not incorporate face-specific information, as it will be proposed in the present work in Chapter 6. Also, in most cases, the aim is to produce high-fidelity images given an image which is already of good resolution (usually 128×128) while face super-resolution methods typically report results on very low resolution faces (16×16 or 32×32).

From all the above mentioned methods, our work, presented in Chapter 6, is more closely related to [51] and [61]. In particular, one of our contributions is to describe an improved GAN-based architecture for super-resolution, which we used as a strong baseline on top of which we built our integrated face super-resolution and alignment network.

2.5.2 Face super-resolution

The recent work of [117] uses a GAN-based approach (like the one of [61] without the perceptual loss) to super-resolve very low-resolution faces. The method was shown to work well for frontal and pre-aligned faces taken from the CelebA dataset [65]. In [118], the same authors proposed a two-step decoder-encoder-decoder architecture which incorporates a spatial transformer network to undo translation, scale and rotation misalignments. Their method was tested on pre-aligned, synthetically generated LR images from the *frontal* dataset of CelebA [65]. Notably, our network described in Chapter 6 does not try to undo misalignment but simply learns how to super-resolve, respecting at the same time the structure of the human face by integrating a landmark localization sub-network.

The closest work to our method presented in Chapter 6 is [131] which performs face super-resolution and dense facial correspondence in an alternating manner. Their algorithm was tested on the frontal faces of PubFig [59] and Helen [60] while few results on real images (4 in total) were also shown with less success. The main difference with our work is that, in [131], the dense correspondence algorithm is not based on neural networks, but on cascaded regression, is pre-learned disjointly from the super-resolution network and remains fixed. As such, [131] suffers from the same problem of having to detect landmarks on blurry faces which is particularly evident for the first iterations of the algorithm. On the contrary, in Chapter 6, we propose learning both super-resolution and facial landmark localization *jointly* in an *end-to-end* fashion, and use *just one shot* to jointly super-resolve the image and localize the facial landmarks. See Fig. 6.2. As it is shown, this results in large performance improvement and generates images of high fidelity across the whole spectrum of facial poses.

It is worth noting that, in Chapter 6, we go beyond the state-of-the-art and rigorously evaluate super-resolution and facial landmark localization across facial pose both quantitatively and qualitatively. As opposed to prior work which primarily uses frontal datasets [118, 16, 44, 131, 117, 110] (e.g. CelebA, Helen, LFW, BioID) to report results, the low resolution images in our experiments were generated using the newly created LS3D-W balanced dataset [11] introduced in Chapter 4 which contains an even number of facial images per facial pose. We also report qualitatively results on more than 200 real-world low resolution facial images taken from the WiderFace dataset [112]. To our knowledge, this is the most comprehensive evaluation of face super-resolution algorithms on real images.

Chapter 3

Face alignment via Convolutional Part Heatmap Regression

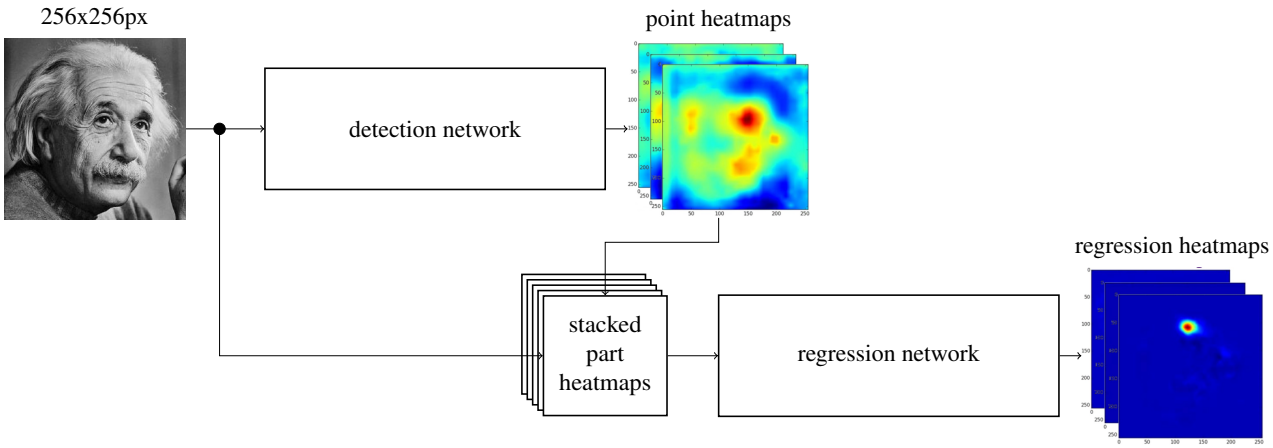


Fig. 3.1 **Proposed architecture**: Our CNN cascade consists of two connected deep sub-networks. The first one (upper part in the figure) is a detection network trained to detect the individual fiducial points using a per-pixel sigmoid loss. Its output is a set of N point heatmaps. The second one is a regression subnetwork that jointly regresses the detection heatmaps stacked along with the input image to confidence maps representing the location of the keypoints.

This Chapter presents a novel approach for large pose face alignment based on heatmap regression that addresses for the first time, to a satisfactory extent, two of the most important face alignment requirements simultaneously: (a) the method must not rely on accurate initialisation/face detection and (b) it should perform equally well for the whole spectrum of facial poses. We note that this method will be further developed and expanded in Chapter 4 for 3D face alignment. In Chapter 5, the concept will be further adapted using a novel architecture to devices with low computational resources.

In particular, to remove the requirement for accurate face detection, our system firstly performs facial part detection, providing confidence scores for the location of each of the facial landmarks (local evidence). Next, these score maps along with early CNN features are aggregated by our system through joint regression in order to refine the landmarks' location. Besides playing

the role of a graphical model, CNN regression is a key feature of our system, guiding the network to rely on context for predicting the location of occluded landmarks, typically encountered in very large poses. Through the chapter, we will be referring to the proposed method as CALE (Convolutional Aggregation of Local Evidence).

When applied to one of the most challenging human face alignment test sets, our method provides more than 50% gain in localisation accuracy when compared to other recently published methods for large pose face alignment. Lastly, in this chapter we go beyond human faces, demonstrating that the proposed method is effective in dealing with very large changes in shape and appearance, typically encountered in animal faces.

The contributions of this Chapter have been published at BMVC 2016 in [7].

3.1 Method

The proposed heatmap regression is a CNN cascade illustrated in Fig. 3.1. Our cascade consists of two connected subnetworks. The first subnetwork is a landmark detection network trained to detect individual fiducial points using a per-pixel softmax loss, thus by-passing the requirement for accurate face detection. The output of this network is a set of N detection heatmaps. The second subnetwork is a regression subnetwork that jointly regresses the detection heatmaps stacked with the image/CNN features to confidence maps representing the location of the facial landmarks.

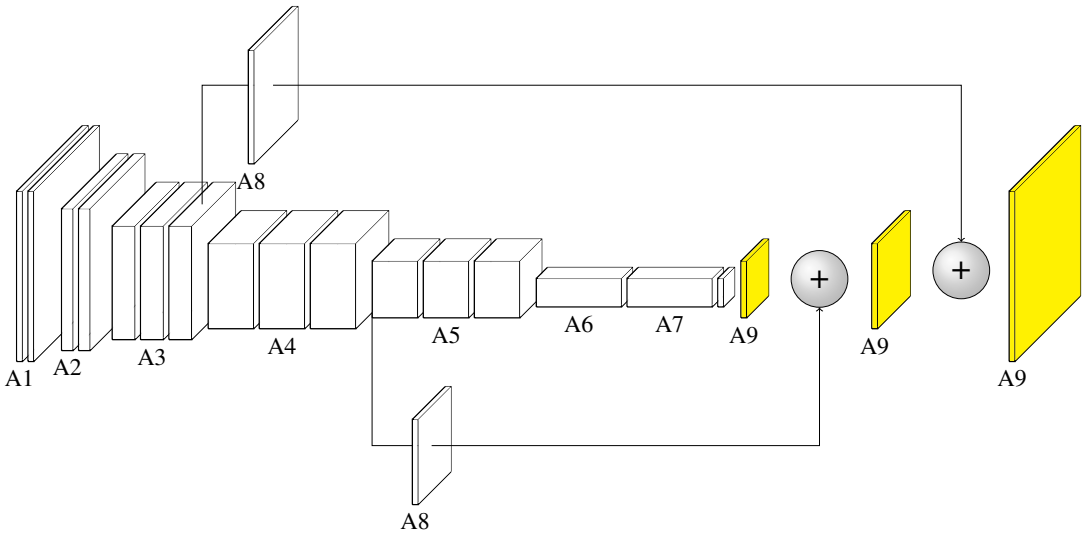


Fig. 3.2 The VGG-FCN subnetwork used for facial landmark detection. The subnetwork takes as input the facial image and outputs a set of N heatmaps, each detecting an individual part. The blocks A1-A9 are defined in Table 3.1.

3.1.1 Detection subnetwork

One of the main issues with almost all prior techniques on face alignment is face detection initialisation. It is well-known that face alignment methods are sensitive to how accurate the face

A1	A2	A3	A4	A5	A6	A7	A8	A9
2× conv layer (64, 3×3, 1×1), pooling	2× conv layer (128, 3×3, 1×1), pooling	3× conv layer (256, 3×3, 1×1), pooling	3× conv layer (512, 3×3, 1×1), pooling	3× conv layer(512, 1×1, 1×1), pooling	conv layer (4096, 7×7, 1×1)	conv layer (4096, 1×1, 1×1)	conv layer(16, 1×1, 1×1)	bilinear upsample

Table 3.1 Block specification for the VGG-FCN facial landmark detection subnetwork. Torch notations (channels, kernel, stride) and (kernel, stride) are used to define the conv and pooling layers.

detection algorithm is, with faces in difficult poses being usually detected with less accuracy. A second important, but not so well-discussed, issue is that typically face alignment methods are tight with a specific face detector, with alignment accuracy rapidly deteriorating if a different face detector (than the one that the face alignment algorithm was trained on) is used. Notably, face alignment methods are usually tight to both the statistics of the face detector and the definition of the face region that the detector was trained on.

To overcome the strong dependency on the face detector, we propose to firstly perform detection of the individual facial landmarks. While [66] uses a per-pixel softmax loss encoding different classes with different numeric levels, in practice, for faces this is suboptimal because the points are usually within close proximity to each other, having high chance of overlapping. Therefore, we follow an approach similar to [123] and encode landmark label information as a set of N binary maps, one for each point, in which the values within a certain radius around the provided ground truth location are set to 1 and the values for the remaining background are set to 0. This way, we thus tackle the problem of having multiple points in the very same region. Note that the detection network is trained using visible points only, which is fundamentally different from the previous regression-based approaches[74, 99, 98] applied to human pose estimation.

The radius defining "correct location" was selected so that the targeted point is fully included inside. Empirically, we determined that a value of 10px to be optimal for a face size of 200px computed as the square root of the tight bounding box.

We train our keypoints detectors jointly using pixelwise sigmoid cross entropy loss function:

$$l_1 = \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{W}\sum_{j=1}^{H}[p_{ij}^n \log \hat{p_{ij}^n} + (1-p_{ij}^n)\log(1-\hat{p_{ij}^n})], \tag{3.1}$$

where p_{ij}^n denotes the ground truth map of the nth landmark at pixel location (i, j) (constructed as described above) and $\hat{p_{ij}^n}$ is the corresponding sigmoid output at the same location.

In terms of architecture, we based our landmark detection network architecture on the VGG-16 network [88] converted to fully convolutional by replacing the fully connected layers with convolutional layers of kernel size of 1 [66]. Because the localization accuracy offered by the 32px stride is insufficient, we make use of the entire algorithm as in [66] by combining the earlier level CNN features, thus reducing the stride to 8px. For convenience, the network is shown in Fig. 3.2 and Table 3.1.

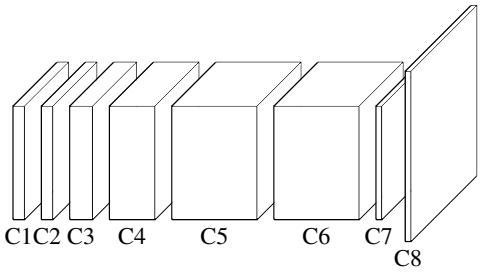


Fig. 3.3 The VGG-based subnetwork used for regression. The subnetwork takes as input the image stacked alongside the heatmaps produced by the detection subnetwork and outputs the final regressed heatmaps. The blocks C1-C8 are defined in Table 3.2.

C1	C2	C3	C4	C5	C6	C7	C8
conv layer(64, 9×9, 1×1)	conv layer(64, 13×13, 1×1)	conv layer(128, 13×13, 1×1)	conv layer(256, 15×15, 1×1)	conv layer(512, 1×1, 1×1)	conv layer(512, 1×1, 1×1)	conv layer(16, 1×1, 1×1)	deconv layer (16, 8×8, 4×4)

Table 3.2 Block specification for the VGG-based regression subnetwork. Torch notations (channels, kernel, stride) and (kernel, stride) are used to define the conv and pooling layers.

3.1.2 Regression subnetwork

While the detectors alone provide good performance, they lack a strong relationship model that is required to improve (a) accuracy and (b) robustness particularly required in situations where specific landmarks are occluded. To this end, we propose an additional subnetwork that jointly regresses the location of all landmarks (both visible and occluded). The input of this subnetwork is a multi-channel representation produced by stacking the N heatmaps produced by the detection subnetwork, along with the input image (see Fig. 3.1). This multichannel representation guides the network where to focus and encodes structural landmark relationships. Additionally, it ensures that our network does not suffer from the problem of regressing occluded landmark appearances: because the detection heatmaps for the occluded landmarks provide low confidence scores, they subsequently guide the regression part of our network to rely on contextual information (provided by the remaining landmarks) in order to predict the location of these landmarks.

The goal of our regression subnetwork is to predict the points' location via regression. However, direct regression of the points is a difficult and highly non-linear problem caused mainly by the fact that only one single correct value needs to be predicted. We address this by following a simpler alternative route [99, 74], regressing a set of confidence maps located in the immediate vicinity of the correct location (instead of regressing a single value). The ground truth consists of a set of N layers, one for each (see Fig. 5.1b), in which the correct location of each landmark, be it visible or not is represented by Gaussian with a standard deviation of 5px.

We train our subnetwork to regress the location of all landmarks jointly using the following L2 loss:

$$l_2 = \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{W}\sum_{j=1}^{H}\left\|\hat{p}_{i,j}^{n} - p_{i,j}^{n}\right\|^2, \tag{3.2}$$

where $\hat{p}_{i,j}^{n}$ and $p_{i,j}^{n}$ represent the predicted and the ground truth confidence maps at pixel location (i, j) for the nth landmark, respectively.

For the regression subnetwork, we have chosen a very simple architecture, consisting of 7 convolutional layers. The network is shown in Fig. 3.3 and Table 3.2. The first 4 of these layers use a large kernel size that varies from 7 to 15, in order to capture a sufficient local context and to increase the receptive field size which is crucial for learning long-range relationships. The last 3 layers have a kernel size equal to 1.

3.1.3 Detection vs regression

While the detection stage targets only the visible parts, ignoring the rest (i.e. it detects the presence or the absence of a part), it does not develop strong relations between them. Additionally, to emphasize its detection nature, the ground truth is represented as a disc with values of 1 in the vicinity of the exact location being centered around it. As opposed to this, the regression stage targets all the keypoints, both visible and occluded, trying to learn an implicit model in the process. The ground truth is more localised penalising the distance from the current pixel to the correct one. In practice, as mentioned in the previous subsection, this is represented using a 2D gaussian centered around the ground truth location.

3.1.4 Training

We trained our CNN landmark detectors by fine-tuning from a VGG-16 network that was previously trained on ImageNet [28]. We followed a training procedure similar to the one described in [66] by firstly, performing a "network surgery" which converts VGG-16 to a fully convolutional network. We firstly trained the 32-stride model with a learning rate of $1e-7$ for 10 epochs. Because the 32-stride version of the network does not provide enough resolution, we went all the way down to 8-stride. The detectors were trained under this setting for 20 epochs (25 for the Cats&Dogs dataset) with a learning rate of $1e-8$. Then, we gradually reduced the learning rate twice, down to $1e-10$. All the new learned layers were initialised with zeros. In order to avoid early divergence, we froze the learning for all CNN detector layers and set temporary the learning rate to 0, training only the CNN regressor. We trained the sub-network for 30 epochs with a learning rate of $1e-6$. After 20 epochs, we lowered it to $1e-7$ and continued the training until convergence was reached. The entire network (CNN detector and CNN regressor) was then trained jointly, in an end-to-end fashion for 5 more epochs. All the new layers added were initialised with a random Gaussian distribution with standard deviation of 0.01. For face alignment, the network was trained on the entire training set of AFLW-PIFA (10% of images that were kept for validation). We followed a similar approach on the Cats&Dogs Dataset, holding 10% of the total of 3000 images for validation and 250 images for testing.

Regarding data augmentation, we applied image flipping and scale jittering (0.8-1.2). Because the images provided in the AFLW-PIFA dataset were grayscale, the human face alignment model was trained with grayscale images, while the one for animals using colour images.

All models were trained and tested using Caffe[50] on a single Titan X GPU.

3.2 Results

We firstly report results on the most challenging and large scale dataset (at the time the method was developed) for large pose human face alignment, namely AFLW-PIFA [53], illustrating that CALE reduces the error achieved by state-of-the-art methods [132, 130] by more than 50%. Then, we report results on our Cats&Dogs dataset, illustrating, for the first time, that a face alignment method is capable of achieving similar performance on both animal and human faces.

3.2.1 Human faces

We have opted not to report results on LFPW [6], Helen [60] and 300-W [83] which are all frontal datasets containing a small portion of test images and are currently being considered as saturated [132, 130]. Instead we report performance on AFLW-PIFA which was at the time the method was developed the most challenging dataset for large pose face alignment [53]. In particular, the authors of [53] created a subset of AFLW [57] that has a balanced distribution of yaw angles (from -90 degrees to 90 degrees) including 3901 images for training and a large number of 1299 for testing. Notably, besides the existing 21 key points, this subset contains 13 new landmarks, making the total number of annotated keypoints equal to 34. All the images are annotated from a 3D perspective which makes the landmark location prediction even more difficult, making AFLW-PIFA the most challenging dataset for face alignment. We report results on the original 21 point annotations [53] as well as on the new ones, based on 34 points [54].

The evaluation metric used for AFLW-PIFA subset is the Normalized Mean Error (NME), which is the average of the normalized (by the face size as defined in [54]) estimation error of the visible landmarks:

$$\text{NME} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{f_i|v_i|_1}\sum_{j}^{N_k} v_i(j)\left\|\widetilde{L}_i(:,j) - L_i(:,j)\right\|, \tag{3.3}$$

where we denoted by $\widetilde{L}_i$ the estimated landmarks location, L_i the corresponding ground truth 2D landmarks, v_i the visibility label and $|v_i|_1$ the number of visible landmarks of image I_i. $L(:,j)$ and $\widetilde{L}(:,j)$ is the jth column of L_i and $\widetilde{L}_i$ respectively. N is the total number of faces and N_k the number of keypoints. For each image, the error is normalized by f_i, which for ALFW-PIFA is the square root of the face size calculated from the bounding box as in [54].

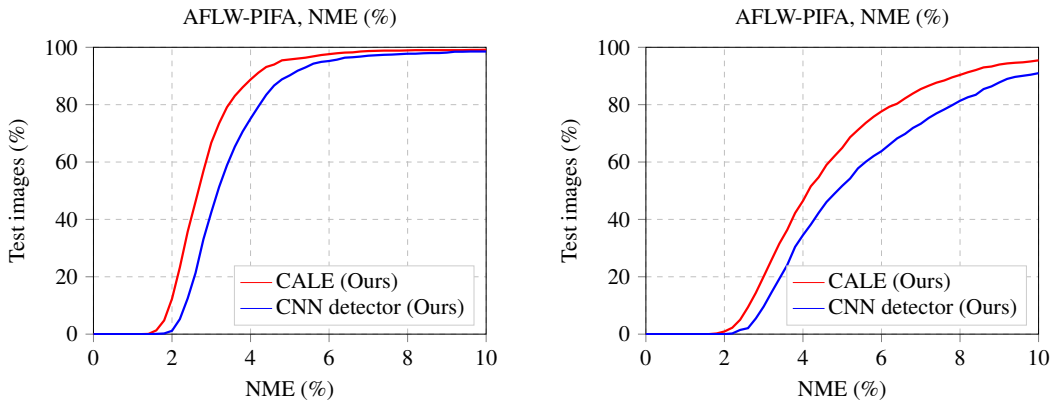
Firstly, we compare the performance of our CNN detector alone with that of the overall CNN architecture (CALE). We opted to report performance on both occluded and visible points. The results on AFLW-PIFA are given in Table 3.3 and Fig. 3.4. We observe that although the CNN

Method	21 points (vis.)	21 points	34 points (vis)	34 points
CNN detector	3.32	5.53	3.63	5.96
CALE	**2.63**	**4.38**	**2.96**	**4.97**

Table 3.3 Performance analysis of CALE on AFLW-PIFA using NME (%). Results are reported on both 21 and 34 points. Results marked with `(vis)` are calculated on visible points only, while the rest are calculated on both occluded and visible landmarks.

detector alone performs very well, CALE largely outperforms it achieving very high alignment accuracy. The performance improvement offered by CALE is even greater on the occluded points, verifying the usefulness of the CNN regressor for the difficult poses and occlusions of AFLW-PIFA.

Next, we compare the performance of our method with that of currently considered state-of-the-art methods for large pose face alignment, also including the very recent works of [54] and [130]. Tables 3.4 and 3.5 summarise our results on AFLW-PIFA on both 21 and 34 points for the visible points only. From Table 3.4, we observe that CALE largely outperforms all other methods by a remarkable more than 50%, reducing the error of the second best performing method [130] to more than half. Similarly, from Table 3.5, we observe that the improvement over the second best performing method approaches 37%. Note that prior work reports on visible points, only. To the best of our knowledge we are the first to report results on non-visible (i.e. either occluded or self-occluded due to the pose) landmarks too (see Table 3.3). Remarkably, the performance of CALE when evaluated on all points - both visible and occluded (see Table 3.3) surpasses the performance of all existing methods when these are evaluated on visible points only (see Tables 3.4 and 3.5). Fitting results from AFLW-PIFA can be seen in Fig. 3.5.



(a) Evaluation on visible points (b) Evaluation on both invisible and visible

Fig. 3.4 NME-based (%) comparison between CNN detector and CALE on AFLW-PIFA on 34 points.

CDM [116]	CFSS [127]	ERT [55]	SDM [108]	PIFA [53]	CCL [130]	Ours
8.59	6.75	7.03	6.96	6.52	5.81	**2.63**

Table 3.4 NME-based (%) comparison on AFLW-PIFA on 21 points (visible landmarks only). The results for CFSS, ERT and SDM are taken from [130].

Evaluation	PIFA [53]	RCPR [14]	PAWF [54]	Ours
AFLW-PIFA	8.04	6.26	4.72	**2.96**

Table 3.5 NME-based (%) comparison on AFLW-PIFA evaluated on 34 points (visible landmarks only). The results for PIFA, RCPR and PAWF are taken from [54].

Fig. 3.5 Qualitative fitting results produced by CALE on AFLW-PIFA test set. Observe that our method copes well for both occlusions and difficult poses. Blue/Yellow points indicate visible/invisible landmarks. All the keypoints are detected from a **3D perspective**, so the non-visible (yellow) points are actually accurately localised for the majority of cases.

Evaluation	Ours
Cats&Dogs (Cats subset)	**2.72**
Cats&Dogs (Dogs subset)	**2.71**

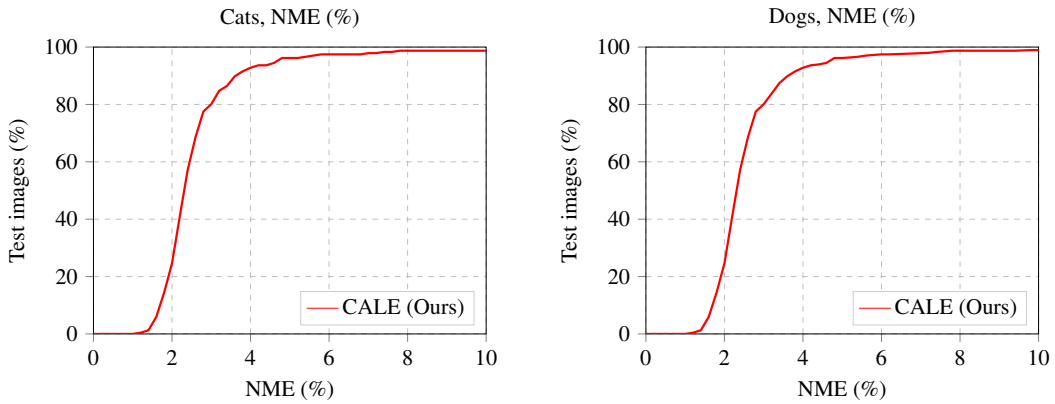
Table 3.6 NME-based (%) performance on Cats&Dogs on 22 points.

3.2.2 Animal faces

While human face alignment is a well-studied problem, the problem of animal face alignment, to the best of our knowledge, has never been systematically explored in the past by the Computer Vision community. As animal faces exhibit a much larger degree of variability in shape and appearance as well as in pose and expression, animal face alignment is considered a much more difficult problem. Cats and dogs, the two species chosen here, are the most popular companion animals, worldwide and of enormous societal and economic importance. Motivated by our results on human face alignment, we investigate CALE's performance on cat and dog face alignment. Although drawing a direct comparison is not possible, our results, both quantitative and qualitative (see Figs 3.5 and 3.7), show that CALE's performance on animal faces is not far from that on human faces.

Our Cats&Dogs dataset is a subset of the Oxford-IIIT-Pet dataset [72] which contains a rich variety of cats/dogs breeds, making the dataset particularly challenging. Our dataset contains 1511 images of cats and 1514 of dogs. For both animals, we kept 250 images for testing and used the rest for training. We used 22 landmarks similarly defined for both species (see 3.7). To measure performance, we used the same metric as the one used for AFLW-PIFA.

Fig. 3.6 and Table 3.6 summarise our results on 22 points. As we may observe, CALE literally produces the same fitting accuracy for both species.



(a) Cats. (b) Dogs.

Fig. 3.6 NME-based (%) performance on Cats&Dogs on 22 points.

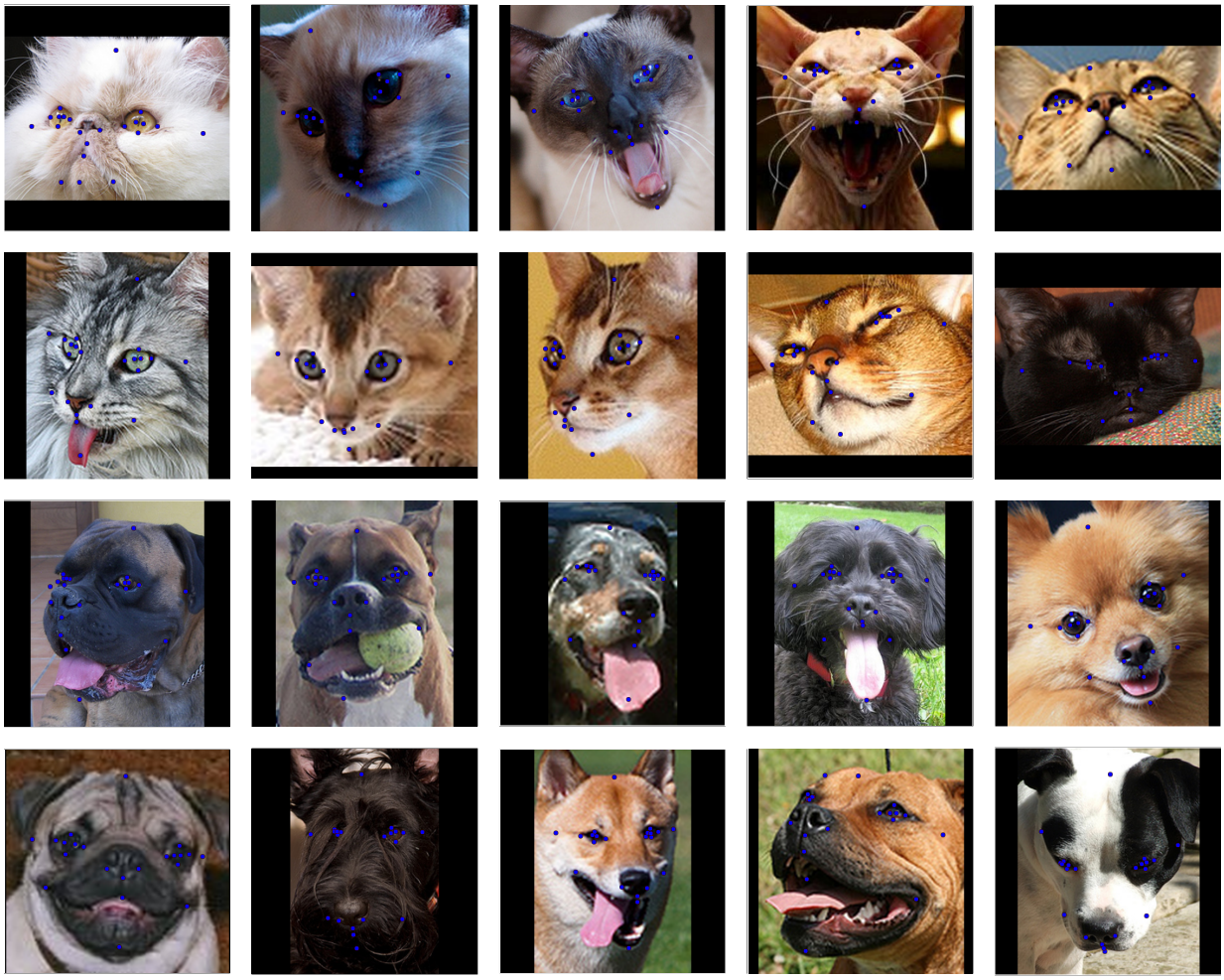


Fig. 3.7 Qualitative results produced by CALE on our Cats&Dogs dataset.

Chapter 4

Toward solving the 2D & 3D face alignment problem (and a dataset of 230.000 images)

This chapter investigates how far a very deep neural network is from attaining close to saturating performance on existing 2D and 3D face alignment datasets. To this end, we construct, for the first time, a very strong baseline by combining a state-of-the-art architecture for landmark localization with a state-of-the-art residual block introduced in Chapter 5, train it on a very large yet synthetically expanded 2D facial landmark dataset and finally evaluate it on *all other* 2D facial landmark datasets. We then create a guided by 2D landmarks network which converts 2D landmark annotations to 3D and unifies all existing datasets, leading to the creation of LS3D-W, the largest and most challenging 3D facial landmark dataset to date (~230,000 images). Following that, we train a neural network for 3D face alignment and evaluate it on the newly introduced LS3D-W. We further look into the effect of all "traditional" factors affecting face alignment performance like large pose, initialization and resolution, and introduce a "new" one, namely the size of the network. Finally, we show that both 2D and 3D face alignment networks achieve performance of remarkable accuracy which is probably close to saturating the datasets used.

The contributions of this Chapter have been published at ICCV 2017 in [11].

4.1 Datasets

In this Section, we provide a description of how existing 2D and 3D datasets were used for training and testing for the purposes of our experiments. We note that the 3D annotations preserve correspondence (i.e. their location corresponds to the actual one and the points are not simply placed at the visible edge of the face as in the 2D case, see also Fig. 1.4) across pose as opposed to the 2D ones and, in general, they should be preferred. We emphasize that the 3D annotations are actually the 2D projections of the 3D facial landmark coordinates but for simplicity we

will just call them 3D. In Section 4.8, we present a method for extending these annotations to full 3D. Finally, we emphasize that we performed cross-database experiments only. A detailed description of the datasets mentioned bellow can be found in Section 2.2.

4.1.1 Training datasets

For training and validation, we used 300-W-LP [132], a synthetically expanded version of 300-W [82]. 300-W-LP provides both 2D and 3D landmarks allowing for training models and conducting experiments using both types of annotations. For some 2D experiments, we also used the original 300-W dataset [82] and 300-VW dataset [87] for fine tuning, only. This is because the 2D landmarks of 300-W-LP are not entirely compatible with the 2D landmarks of the test sets used in our experiments (i.e. 300-W test set, [81], 300-VW [87] and Menpo [119]), but the original annotations from 300-W are. 10% of the training data was held out for validation.

4.1.2 Test datasets

While there is a large number of 2D datasets this type of annotations is problematic since for moderately large poses 2D landmarks lose correspondence. Currently, the only in-the-wild 3D test set is AFLW2000-3D [132] [1] We address this significant gap in 3D face alignment datasets in Section 4.5 by introducing a new testing dataset, LS3D-W-Balanced and re-annotate automatically the AFLW2000-3D dataset that is relatively noisy for some difficult cases. As such, in this chapter, we test the performance of our method on the original AFLW2000-3D dataset (containing 2,000 images), on its re-annotated version and on the newly introduced LS3D-W-Balanced dataset (consisting of 7,200 images).

4.1.3 Metrics

Traditionally, the metric used for face alignment is the point-to-point Euclidean distance normalized by the interocular distance [25, 82, 87]. However, as noted in [133], this error metric is biased for profile faces for which the interocular distance can be very small. Hence, we normalize by the bounding box size. In particular, we used the Normalized Mean Error defined as:

$$\text{NME} = \frac{1}{N}\sum_{i=1}^{N}\frac{\left\|\widetilde{L}_i - L_i\right\|_2}{d}, \tag{4.1}$$

where L_i denotes the ground truth landmarks for a given face, $\widetilde{L}_i$ the corresponding prediction and d is the square-root of the ground truth bounding box, computed as $d = \sqrt{w_{bbox} * h_{bbox}}$. Although we conducted both 2D and 3D experiments, we opted to use the same bounding box definition for both experiments; in particular we used the bounding box calculated from the 2D landmarks. This way, we can readily compare the accuracy achieved in 2D and 3D.

[1] The data from [49] includes mainly images collected in the lab and do not cover the full spectrum of facial poses.

4.2 Background

The residual block is the main building block of the Hourglass (HG) network, shown in Fig. 4.1, which is a state-of-the-art architecture for landmark localization that predicts a set of heatmaps (one for each landmark) in a fully convolutional fashion. The HG network is an extension of [66] allowing however for a more symmetric top-down and bottom-up processing. See also [69].

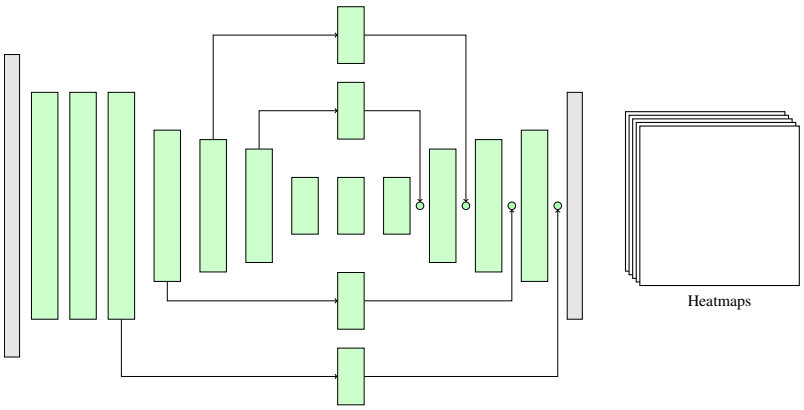


Fig. 4.1 The architecture of a single *Hour-Glass* (HG) network [69]. The network takes as input the features and the heatmaps produced at the $l-1$ stage and outputs a new set of heatmaps (predictions). Throughout this chapter the hourglass itself operates at a resolution of 64×64px .

4.3 Method

This Section describes the main idea of our method introducing all the architectural variations used to localise the 2D and 3D landmarks and to construct the very large scale 3D face alignment dataset (LS3D-W) containing more than 230,000 3D landmark annotations. Namely we present the following networks:

- **2D-FAN** The network consists of up to 4 HGs, taking as input a facial image and producing a set of heatmaps containing the 2D locations of the points. Note: the 2D points follow the visible boundary of the face and as such, for faces found in large poses, they lose the correspondence with the actual location, for more details please see Section 1.3.2.
- **3D-FAN** The network takes as input the facial image and outputs a set of heatmaps containing the 2D projection of the 3D points.
- **2D-to-3D-FAN** This network takes as input the heatmaps produced by 2D-FAN, stacks them alongside the facial image and produces a set of heatmaps that contain the 2D projection of the 3D points.
- **3D-FAN-full** This variant consists of a 3D-FAN network followed by a ResNet. As opposed to 3D-FAN, 3D-FAN-full also predicts the depth (i.e. the z coordinate) alongside (x, y) with the help of the ResNet subnetwork.

- **2D-to-3D-FAN-full** Combines 2D-to-3D-FAN with 3D-FAN-full. The network takes as input the 2D heatmaps produced by 2D-FAN alongside the facial image and outputs the (x, y, z) location of the points.

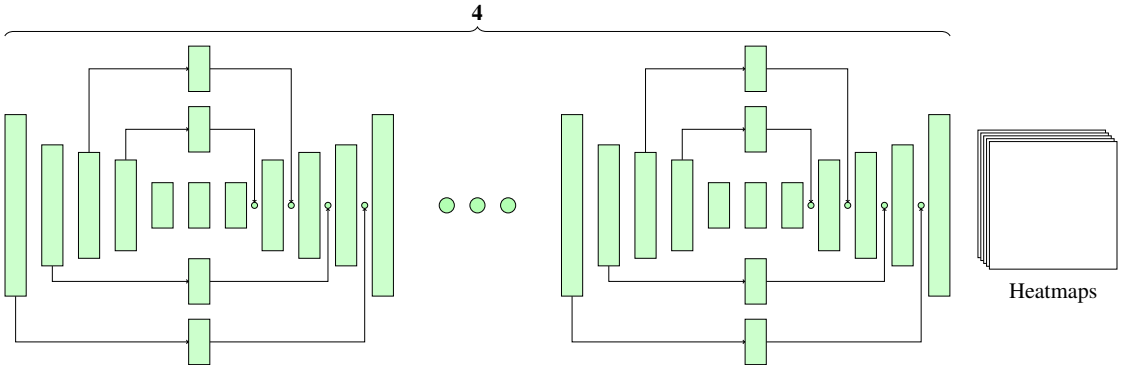


Fig. 4.2 The Face Alignment Network (FAN) constructed by stacking four HGs in which all bottleneck blocks (depicted as rectangles) were replaced with the hierarchical, parallel and multi-scale block of [10]. The network takes as input a facial image (at a resolution of 256×256px) and outputs a set of heatmaps, one for each landmark.

4.3.1 2D and 3D Face Alignment Networks

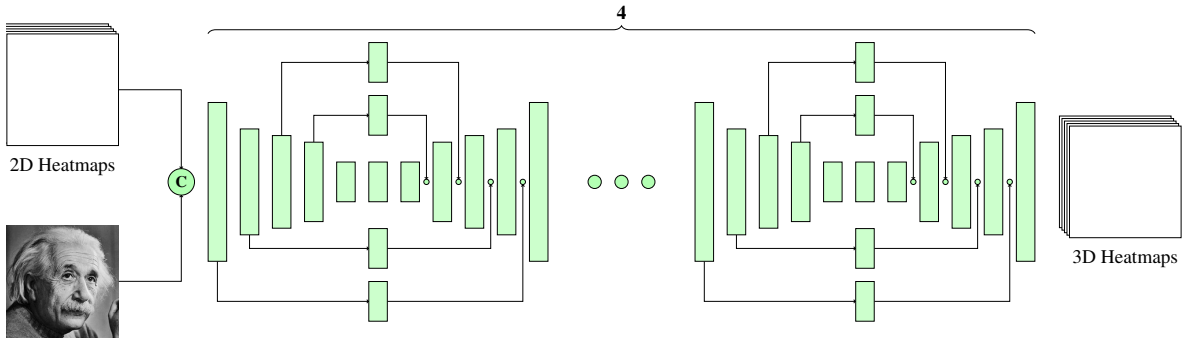


Fig. 4.3 The 2D-to-3D-FAN network used for the creation of the LS3D-W dataset. The network takes as input the RGB image and the 2D landmarks and outputs the corresponding 2D projections of the 3D landmarks. Note: '3D heatmaps' denotes the 2D projection of the 3D points represented using 2D heatmaps.

We coin the network used for our experiments simply Face Alignment Network (FAN). To our knowledge, it is the first time that such a powerful network (in terms of depth and capacity) is trained and evaluated for large scale 2D/3D face alignment experiments.

We construct FAN based on one of the state-of-the-art architectures for human pose estimation, namely the *Hour-Glass* (HG) network of [69]. In particularly, we used a stack of four HG networks (see Fig. 4.3). While [69] uses the bottleneck block of [38] as the main building block for the HG, we go one step further and replace the bottleneck block with our recently

introduced hierarchical, parallel and multi-scale block presented in Chapter 5. As we will show in Chapter 5, this block outperforms the original bottleneck of [38] when the same number of network parameter were used. Finally, we used 300-W-LP-2D and 300-W-LP-3D to train 2D-FAN and 3D-FAN, respectively.

Our aim is to create the very first very large scale dataset of 3D facial landmarks for which annotations are scarce. To this end, we followed a guided-based approach in which a FAN for predicting 3D landmarks is guided by 2D landmarks. In particular, we created a 3D-FAN in which the input RGB channels have been augmented with 68 additional channels, one for each 2D landmark, containing a 2D Gaussian with std = 1px centred at each landmark's location. We call this network 2D-to-3D FAN. Given the 2D facial landmarks for an image, 2D-to-3D FAN converts them to 3D. To train 2D-to-3D FAN, we used 300-W-LP which provides both 2D and 3D annotations for the same image. We emphasize again that the 3D annotations are actually the 2D projections of the 3D coordinates but for simplicity we call them 3D. Please see Section 4.8 for extending these annotations to full 3D.

4.3.2 Training

For all of our experiments, we independently trained three distinct networks on the 300-W-LP dataset (holding 10% of the data for validation): 2D-FAN, 3D-FAN, and 2D-to-3D-FAN. Note, that the networks where further finetuned on 300-W and 300-VW dataset when evaluated on 300-W-testset and on the testset 300-VW respectively. For the first two networks, we set the initial learning rate to 10^{-4} and used a minibatch of 10. During the process, we dropped the learning rate to 10^{-5} after 15 epochs and to 10^{-6} after another 15, training for a total of 40 epochs. We also applied random augmentation: flipping, rotation (from -50° to 50°), colour jittering, scale noise (from 0.8 to 1.2) and random occlusion. The 2D-to-3D-FAN model was trained by following a similar procedure increasing the amount of augmentation even further: rotation (from -70° to 70°) and scale (from 0.7 to 1.3). Additionally, the learning rate initially was set to 10^{-3}. All networks were implemented in Torch7 [20] and trained using rmsprop [97].

4.4 2D face alignment

This Section evaluates 2D-FAN (trained on 300-W-LP-2D), on 300-W test set, 300-VW (both training and test sets), and Menpo (frontal subset). Overall, 2D-FAN is evaluated on more than 220,000 images. Prior to reporting our results, the following points need to be emphasized:

1. 300-W-LP-2D contains a wide range of poses (yaw angles in $[-90^{\circ}, 90^{\circ}]$), yet it is still a synthetically generated dataset as this wide spectrum of poses were produced by warping the nearly frontal images of the 300-W dataset. It is evident that this lack of real data largely increases the difficulty of the experiment.

2. The 2D landmarks of 300-W-LP-2D that 2D-FAN was trained on are slightly different from the 2D landmarks of the 300-W test set, 300-VW and Menpo. To alleviate this,

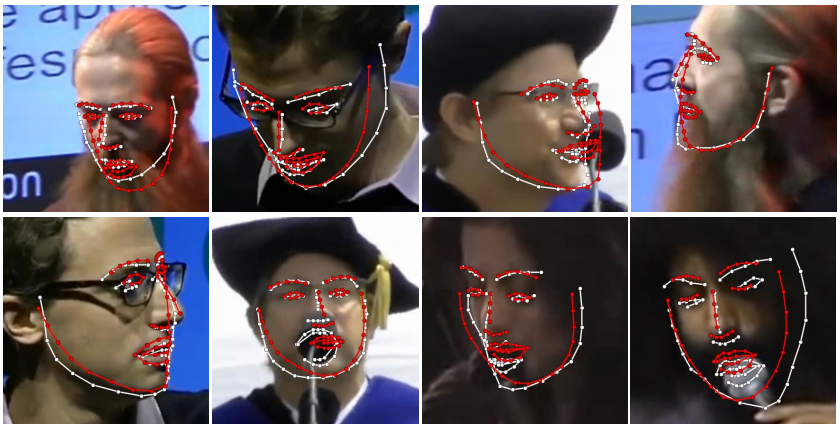


Fig. 4.4 Fittings with the highest error from 300-VW (NME 6.8-7%). Red: ground truth. White: our predictions. In most cases, our predictions are more accurate than the ground truth.

the 2D-FAN was further fine-tuned on the original 300-W training set for a few epochs. Although this seems to resolve the issue, this discrepancy obviously increases the difficulty of the experiment.

3. We compare the performance of 2D-FAN on all the aforementioned datasets with that of an unconventional baseline: the performance of a recent state-of-the-art method, namely MDM [101] on LFPW test set, initialized with the ground truth bounding boxes. We call this result MDM-on-LFPW. As there is very little performance progress made on the frontal dataset of LFPW over the past years, we assume that a state-of-the-art method like MDM (nearly) saturates it. Hence, we use the produced error curve to compare how well our method does on the much more challenging aforementioned test sets.

The cumulative error curves for our 2D experiments on 300-VW, 300-W test set and Menpo are shown in Fig. 4.8. We additionally report the performance of MDM on all datasets initialized by ground truth bounding boxes, ICCR, the state-of-the-art face tracker of [85], on 300-VW (the only tracking dataset), and our unconventional baseline (called MDM-on-LFPW). Comparison with a number of methods in terms of AUC are also provided in Table 4.1.

With the exception of Category C of 300-VW, it is evident that 2D-FAN achieves literally the same performance on all datasets, outperforming MDM and ICCR, and, notably, matching the performance of MDM-on-LFPW. Out of 7,200 images (from Menpo and 300-W test set), there are in total only 18 failure cases, which represent 0.25% of the images (we consider a failure a fitting with NME $> 7\%$). After removing these cases, the 8 fittings with the highest error for each dataset are shown in Fig. 4.5.

Regarding the Category C of 300-VW, we found that the main reason for this performance drop is the quality of the annotations which were obtained in a semi-automatic manner. After removing all failure cases (101 frames representing 0.38% of the total number of frames), Fig. 4.4 shows the quality of our predictions vs the ground truth landmarks for the 8 fittings with

Dataset	**2D-FAN(Ours)**	MDM[101]	iCCR[85]	TCDCN[124]	CFSS[127]
300-VW-A	**72.1%**	70.2 %	65.9%	-	-
300-VW-B	**71.2%**	67.9 %	65.5%	-	-
300-VW-C	**64.1%**	54.6%	58.1%	-	-
Menpo	**67.5%**	67.1%	-	47.9%	60.5%
300-W	**66.9%**	58.1%	-	41.7%	55.9%

Table 4.1 AUC (calculated for a threshold of 7%) on all major 2D face alignment datasets. MDM, CFSS and TCDCN were evaluated using ground truth bounding boxes and the openly available code.

Fig. 4.5 Fittings with the highest error from 300-W test set (first row) and Menpo (second row) (NME 6.5-7%). Red: ground truth. White: our predictions. In most cases, our predictions are more accurate than the ground truth.

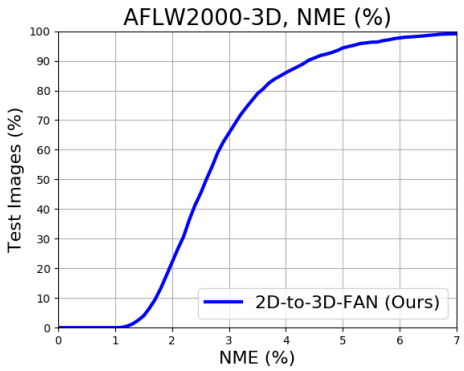


Fig. 4.6 NME on AFLW2000-3D, between the original annotations of [132] and the ones generated by 2D-to-3D-FAN. The error is mainly introduced by the automatic annotation process of [132]. See Fig. 4.7 for visual examples.

the highest error for this dataset. It is evident that in most cases our predictions are more accurate.

Conclusion: Given that 2D-FAN matches the performance of MDM-on-LFPW, we conclude that 2D-FAN achieves near saturating performance on the above 2D datasets. Notably, this result was obtained by training 2D-FAN primarily on synthetic data, and there was a mismatch between training and testing landmark annotations.

4.5 Large Scale 3D Faces in-the-Wild dataset

Motivated by the scarcity of 3D face alignment annotations and the remarkable performance of 2D-FAN, we opted to create a large scale 3D face alignment dataset by converting all existing 2D face alignment annotations to 3D. To this end, we trained a 2D-to-3D FAN as described in Subsection 4.3.2 and guided it using the predictions of 2D-FAN, creating 3D landmarks for: 300-W test set, 300-VW (both training and all 3 testing datasets), Menpo (the whole dataset).

Evaluating 2D-to-3D is difficult: the only available 3D face alignment in-the-wild dataset (not used for training) is AFLW2000-3D [132]. Hence, we applied our pipeline (consisting of applying 2D-FAN for producing the 2D landmarks and then 2D-to-3D FAN for converting them to 3D) on AFLW2000-3D and then calculated the error, shown in Fig. 4.6 (note that for normalization purposes, 2D bounding box annotations are still used). The results show that there is discrepancy between our 3D landmarks and the ones provided by [132]. After removing a few failure cases (19 in total, which represent 0.9% of the data), Fig. 4.7 shows 8 images with the highest error between our 3D landmarks and the ones of [132]. It is evident, that this discrepancy is mainly caused from the semi-automatic annotation pipeline of [132] which does not produce accurate landmarks especially for images with difficult poses.

By additionally including AFLW2000-3D into the aforementioned datasets, overall, ~230,000 images were annotated in terms of 3D landmarks leading to the creation of the Large Scale 3D Faces in-the-Wild dataset (LS3D-W), the largest 3D face alignment dataset to date.

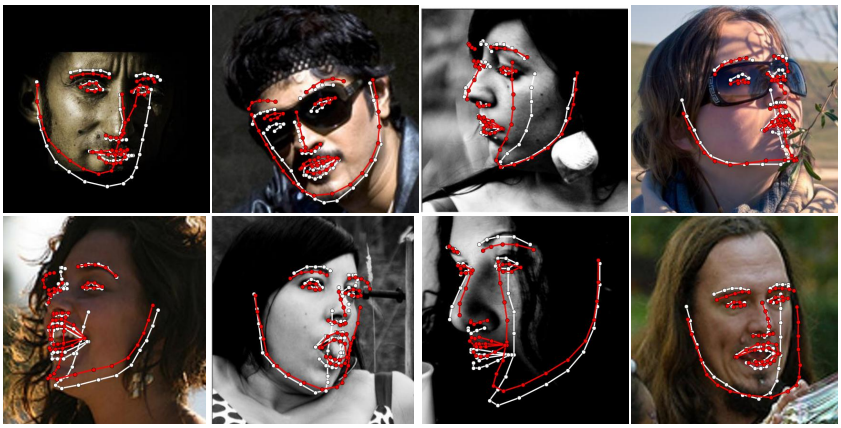


Fig. 4.7 Fittings with the highest error from AFLW2000-3D (NME 7-8%). Red: ground truth from [132]. White: predictions of 2D-to-3D-FAN. In most cases, our predictions are more accurate than the ground truth.

4.6 3D face alignment

This Section evaluates 3D-FAN trained on 300-W-LP-3D, on LS3D-W (described in the previous Section) i.e. on the 3D landmarks of the 300-W test set, 300-VW (both training and test sets), and Menpo (the whole dataset) and AFLW2000-3D (re-annotated). Overall, 3D-FAN is evaluated on ~230,000 images. Note that compared to the 2D experiments reported in Section 4.4, more images in large poses have been used as our 3D experiments also include AFLW2000-3D and the profile images of Menpo (~2000 more images in total).

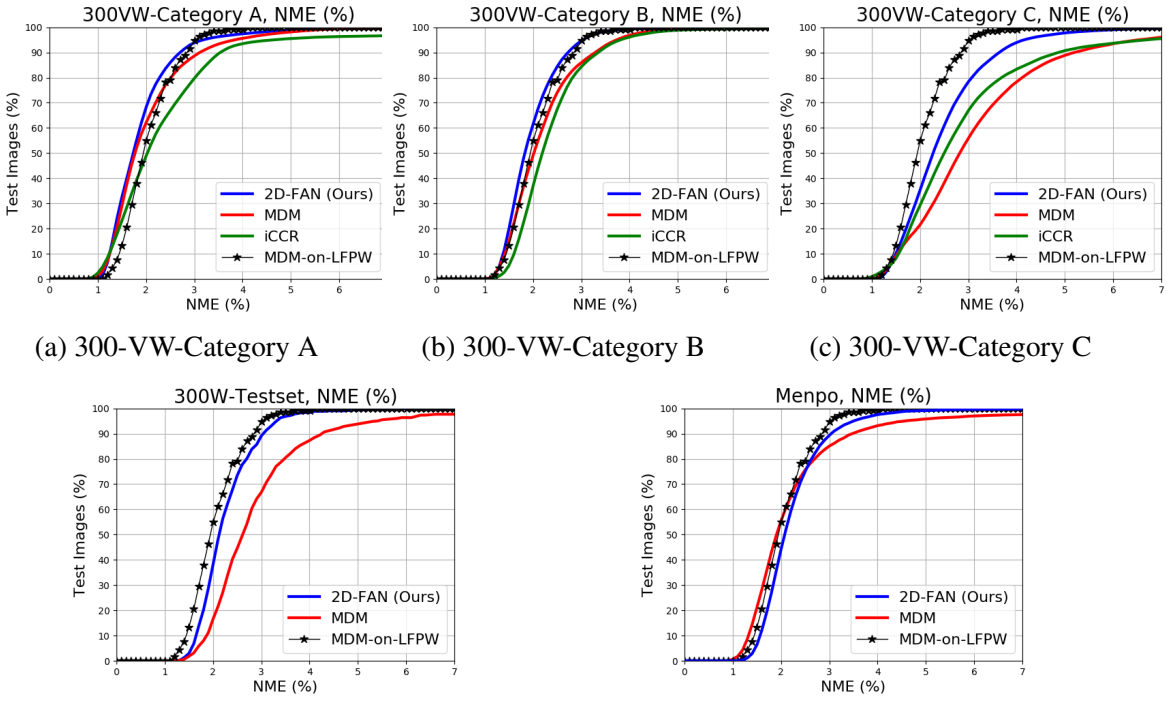
The results of our 3D face alignment experiments on 300-W test set, 300-VW, Menpo and AFLW2000-3D are shown in Fig. 4.9. We additionally report the performance of the state-of-the-art method of 3DDFA (trained on the same dataset as 3D-FAN) on all datasets.

Conclusion: 3D-FAN essentially produces the same accuracy on all datasets largely outperforming 3DDFA. This accuracy is slightly increased compared to the one achieved by 2D-FAN, especially for the part of the error curve for which the error is less than 2% something which is not surprising as now the training and testing datasets are annotated using the same mark-up.

4.7 Ablation studies

To further investigate the performance of 3D-FAN under challenging conditions, we firstly created a dataset of 7,200 images from LS3D-W so that there is an equal number of images in yaw angles $[0^\circ - 30^\circ]$, $[30^\circ - 60^\circ]$ and $[60^\circ - 90^\circ]$. We call this dataset LS3D-W Balanced. Then, we conducted the following experiments:

Performance across pose. We report the performance of 3D-FAN on LS3D-W Balanced for each pose separately in terms of the Area Under the Curve (AUC) (calculated for a threshold of 7%) in Table 4.2. We observe only a slight degradation of performance for very large poses ($[60^\circ - 90^\circ]$). We believe that this is to some extent to be expected as 3D-FAN was largely trained



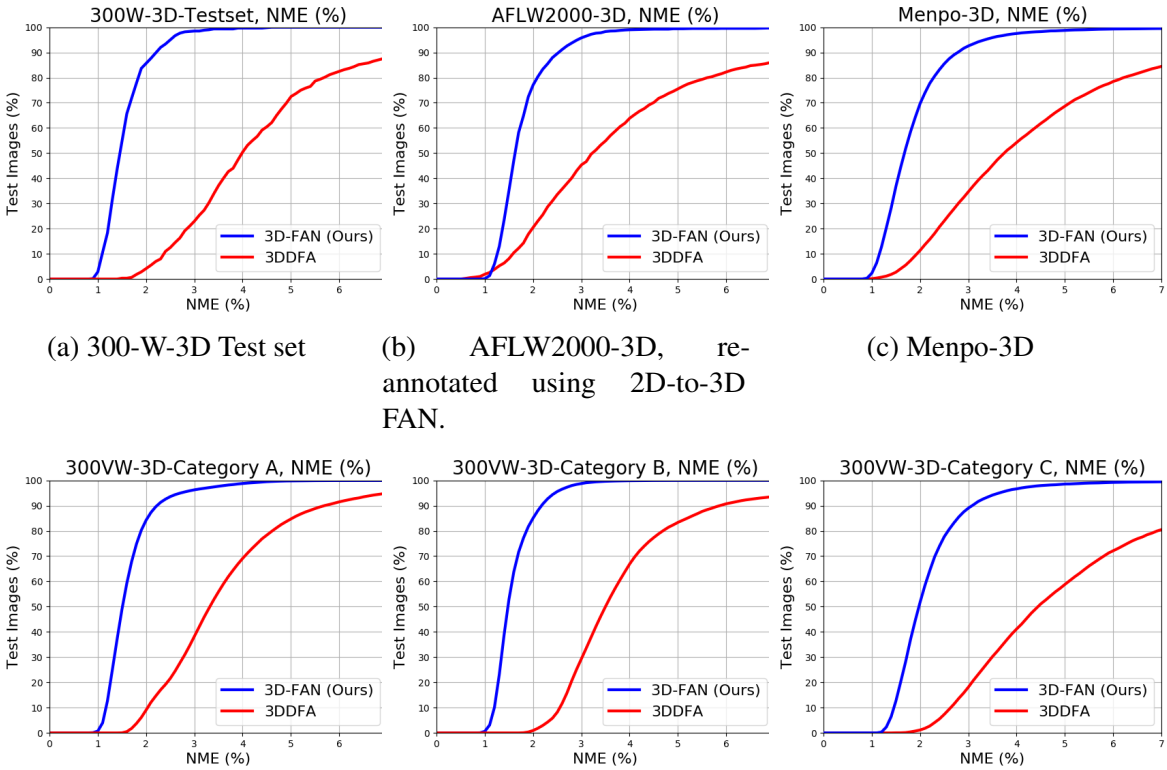
(a) 300-VW-Category A (b) 300-VW-Category B (c) 300-VW-Category C

(d) 300-W Testset (Indoor and Outdoor subset). (e) Menpo (on faces annotated with 68 points).

Fig. 4.8 2D face alignment experiments: NME (all 68 points used) on 300-VW (a-c), 300-W Testset (d) and Menpo (e). Our model is called 2D-FAN. MDM is initialized with ground truth bounding boxes. **Note: MDM-on-LFPW is not a method but the curve produced by running MDM on LFPW test set, initialized with the ground truth bounding boxes.**

Yaw	#images	3D-FAN (Ours)
$[0° - 30°]$	2400	73.5%
$[30° - 60°]$	2400	74.6%
$[60° - 90°]$	2400	68.8%

Table 4.2 AUC (calculated for a threshold of 7%) on the LS3D-W Balanced for different yaw angles.



(a) 300-W-3D Test set (b) AFLW2000-3D, re-annotated using 2D-to-3D FAN. (c) Menpo-3D

(d) 300-VW-3D Category A (e) 300-VW-3D Category B (f) 300-VW-3D Category C

Fig. 4.9 3D face alignment experiments: NME (all 68 points used) on the newly introduced LS3D-W dataset.

Noise	$[0° - 30°]$	$[30° - 60°]$	$[60° - 90°]$
0%	74.5%	75.2%	69.8%
10%	73.5%	74.6%	68.8%
20%	70.8%	71.7%	66.1%
30%	63.8%	63.5%	57.2%

Table 4.3 AUC on the LS3D-W Balanced for different levels of initialization noise. The network was trained with a noise level of up to 20% (the noise is drawn from a uniform distribution that perturbs the bounding box shape).

with synthetic data for these poses (300-W-LP-3D). This data was produced by warping frontal images (i.e. the ones of 300-W) to very large poses which causes face distortion especially for the face region close to the ears.

Conclusion: Facial pose is not a major issue for 3D-FAN.

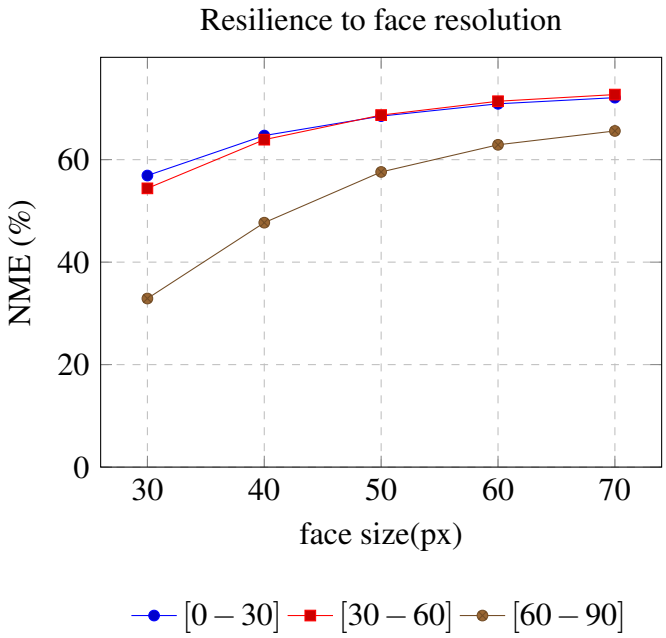


Fig. 4.10 AUC on the LS3D-W Balanced for different face resolutions. Up to 30px, performance remains high.

Performance across resolution. We repeated the previous experiment but for different face resolutions (resolution is reduced relative to the face size defined by the tight bounding box) and report the performance of 3D-FAN in terms of AUC in Fig. 4.10. Note that we did not retrain 3D-FAN to particularly work for such low resolutions. We observe significant performance drop for all poses only when the face size is as low as 30 pixels.

Conclusion: Resolution is not a major issue for 3D-FAN.

Performance across noisy initializations. For all reported results so far, we used 10% of noise added to the ground truth bounding boxes. Note that 3D-FAN was trained with noise level of 20%. Herein, we repeated the previous experiment but for different noise levels and report

#params	$[0° - 30°]$	$[30° - 60°]$	$[60° - 90°]$
2M	70.9%	69.9%	55.8%
4M	71.0%	70.5%	57.0%
6M	71.5%	71.1%	58.3%
12M	72.7%	72.7%	67.1%
18M	73.4%	74.2%	68.3%
24M	73.5%	74.6%	68.8%

Table 4.4 AUC on the LS3D-W Balanced for various network sizes. Between 12-24M parameters, performance remains almost the same.

the performance of 3D-FAN in terms of AUC in Table 4.3. We observe only small performance decrease for noise level equal to 30% which is greater than the level of noise that the network was trained with.

Conclusion: Initialization is not a major issue for 3D-FAN.

Performance across different network sizes. For all reported results so far, we used a very powerful 3D-FAN with 24M parameters. Herein, we repeated the previous experiment varying the number of network parameters and report the performance of 3D-FAN in terms of AUC in Table 4.4. The number of parameters is varied by firstly reducing the number of HG networks used from 4 to 1. Then, the number of parameters was dropped further by reducing the number of channels inside the building block. It is important to note that even then biggest network is able to run on 28-30 fps on a TitanX GPU while the smallest one can reach 150 fps. We observe that up to 12M, there is only a small performance drop and that the network's performance starts to drop significantly only when the number of parameters becomes as low as 6M.

Conclusion: There is a moderate performance drop vs the number of parameters of 3D-FAN. We believe that this is an interesting direction for future work.

4.8 Full 3D face alignment

In this Section, we present an extension of 2D-to-3D-FAN capable of additionally predicting the z coordinate of the facial landmarks.

Similarly to [9], we construct Full-2D-to-3D-FAN by introducing a second subnetwork for estimating the z coordinate (i.e. the depth of each keypoint) on top of 2D-to-3D-FAN. The input to the new subnetwork is the stacked heatmaps produced by 2D-to-3D-FAN alongside the RGB image. The heatmaps play a key role by showing the network where to "look" (i.e at which location should the depth be predicted) incorporating, at the same time, additional pose related information. The proposed subnetwork is based on a ResNet-152 [38] adapted to accept $3+N$ input channels and to output a vector $N \times 1$ instead of 1000×1. The network was trained using the L2 loss for 50 epochs and the same learning rates used for the rest of the networks. Fig. 4.12

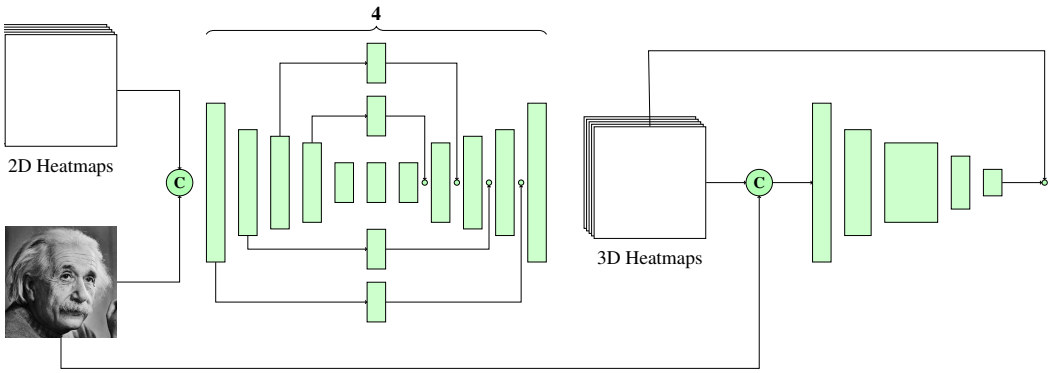


Fig. 4.11 The Full-2D-to-3D-FAN network used for the prediction of the x, y, z coordinates, where the z coordinate is the 1D vector produced by the ResNet subnetwork. The network takes as input an RGB image and the 2D landmarks and outputs the corresponding 3D landmarks.

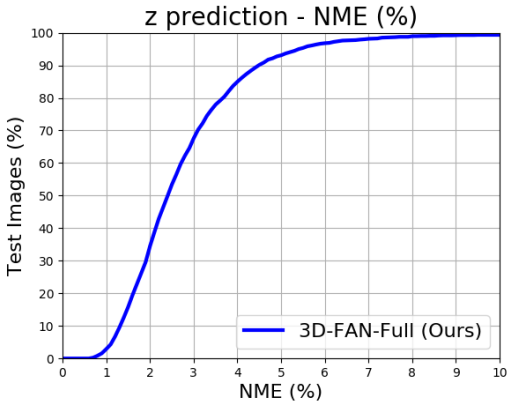


Fig. 4.12 NME on AFLW2000-3D, between the original annotations of [132] and the ones generated by 3D-FAN-Full for depth (z coordinate). Notice that FAN estimates both the depth (z) and x, y locations with similar accuracy, often generating in the process more accurate results.

reports the numerical error of Full-2D-to-3D-FAN on AFLW2000-3D. For visual results, see Fig. 4.15.

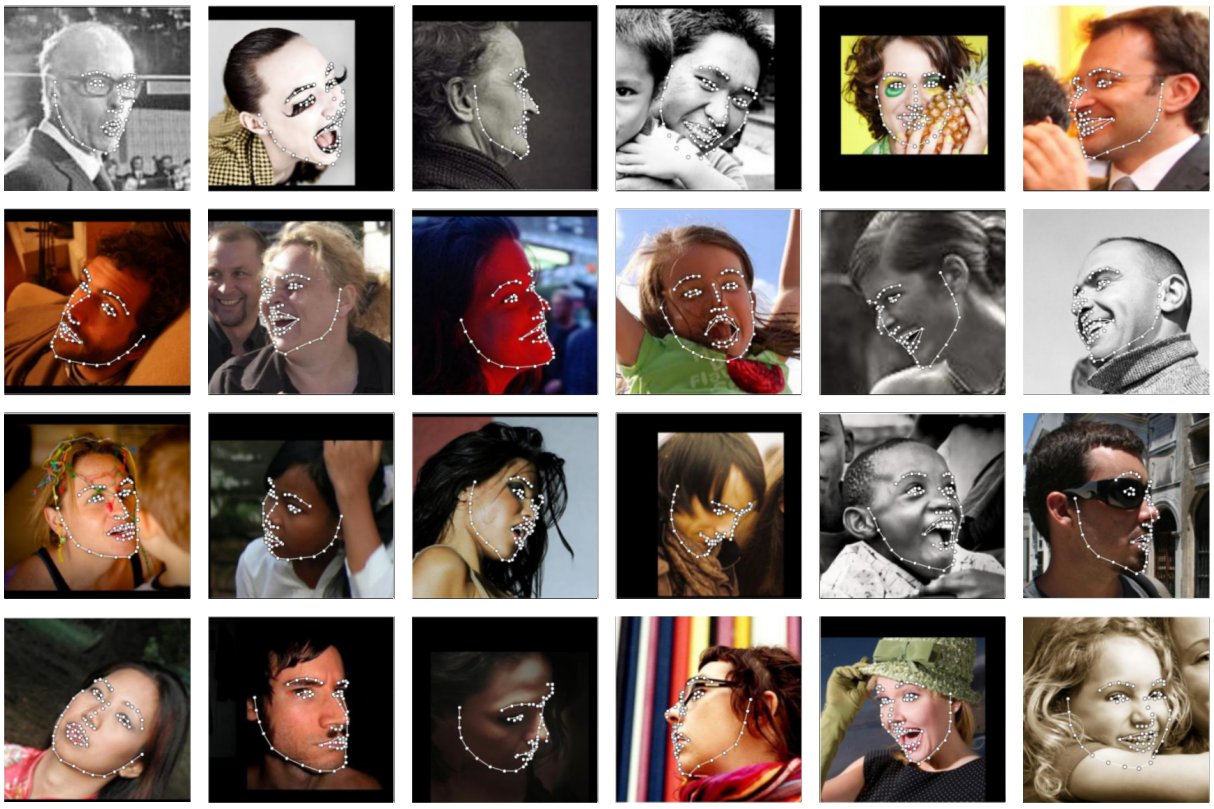


Fig. 4.13 Fitting examples produced by **2D-FAN** on LS3D-W balanced dataset.

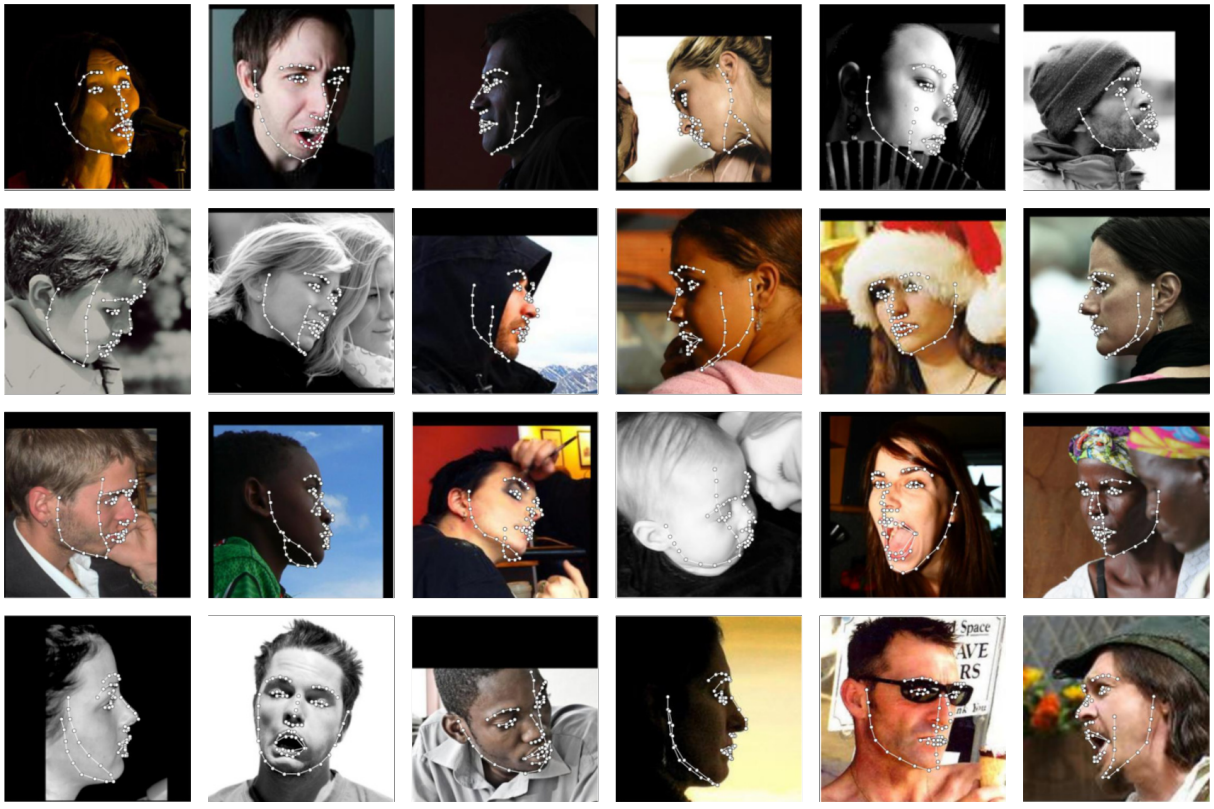


Fig. 4.14 Fitting examples produced by **3D-FAN** on LS3D-W balanced dataset.

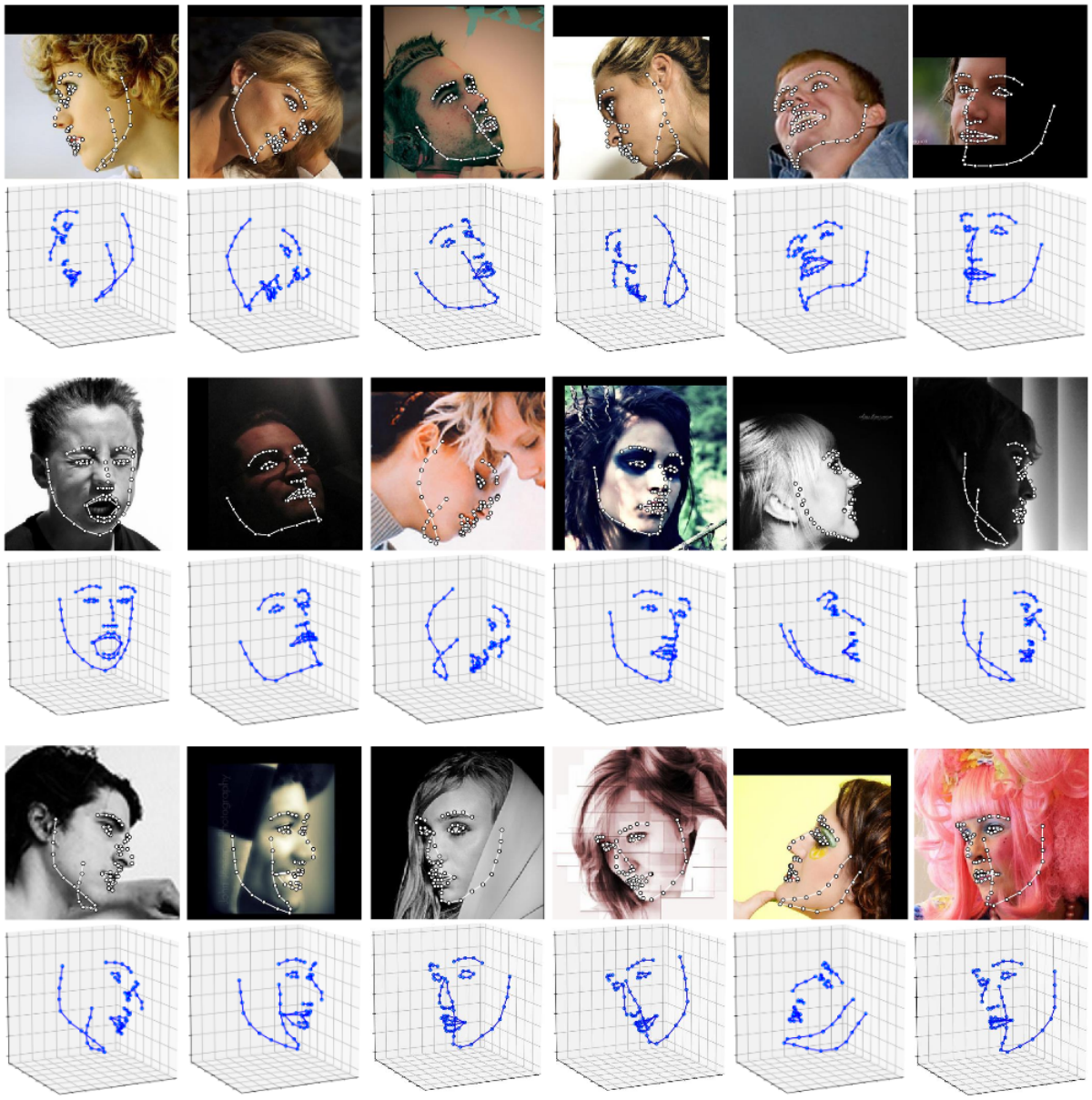


Fig. 4.15 Full 3D fitting examples produced by Full-2D-to-3D-FAN on AFLW2000-3D dataset.

Chapter 5

Hierarchical binary CNNs for landmark localization with limited resources

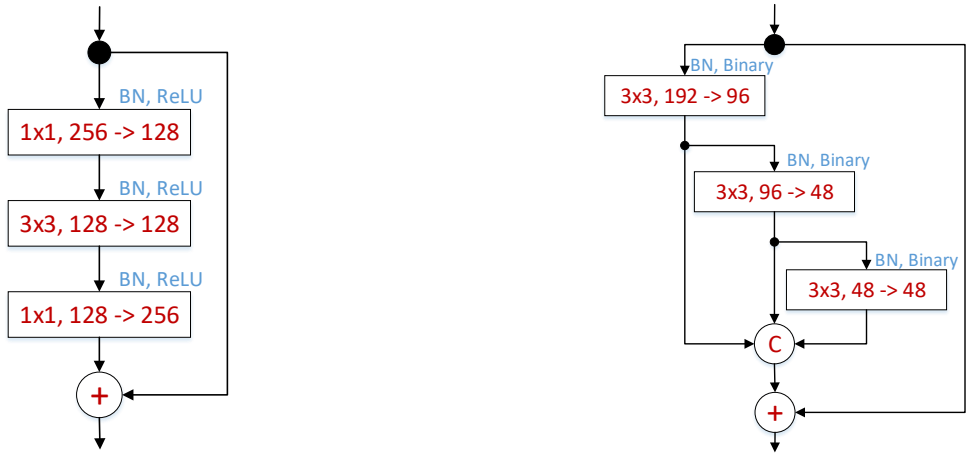
Very recently, work based on Convolutional Neural Networks (CNNs) has revolutionized landmark localization, demonstrating results of remarkable accuracy even on the most challenging datasets for human pose estimation [8, 69, 105] and face alignment [9] (see Chapter 3 and 4). However, deploying (and training) such methods is computationally expensive, requiring one or more high-end GPUs, while the learned models typically require hundreds of MBs, thus rendering them completely unsuitable for real-time or mobile applications. This chapter presents a highly accurate and robust yet efficient and lightweight method for landmark localization using binarized CNNs.

To this end, we study the effect of neural network binarization on localization tasks, focusing on face alignment, exhaustively evaluating various design choices, identifying performance bottlenecks, and more importantly proposing multiple orthogonal ways to boost performance. Based on our analysis, we then propose a novel hierarchical, parallel and multi-scale residual architecture that yields large performance improvement over the standard bottleneck block while having the same number of parameters, thus bridging the gap between the original network and its binarized counterpart. When evaluated on the most challenging datasets for face alignment, we report in many cases state-of-the-art performance.

The contributions of this Chapter have been published at ICCV 2017 (as an Oral) in [10].

5.1 Background

The ResNet consists of two types of blocks: *basic* and *bottleneck*. We are interested only in the latter one which was designed to reduce the number of parameters and keep the network memory footprint under control. We use the "pre-activation" version of [39], in which batch normalization [46] and the activation function precede the convolutional layer. Note that we used the version of bottleneck defined in [69] and shown in Fig. 5.1a the middle layer of which has 128 channels (vs 64 used in [39]). For details regarding HG see Section 4.2 or [69].



(a) original (b) proposed

Fig. 5.1 (a) The original bottleneck layer of [39]. (b) The proposed hierarchical parallel & multi-scale structure: our block increases the receptive field size, improves gradient flow, is specifically designed to have (almost) the same number of parameters as the original bottleneck, does not contain 1×1 convolutions, and in general is derived from the perspective of improving the performance and efficiency for binary networks. **Note:** a layer is depicted as a rectangular block containing: its filter size, the number of input and output channels; "C" - denotes concatenation and "+" an element-wise sum.

5.2 Method

Herein, we describe how we derive the proposed binary hierarchical, parallel and multi-scale block of Fig. 5.3e. In Section 5.3.1, by reducing the number of its parameters to match the ones of the original bottleneck, we further derive the block of Fig. 5.1b. This Section is organized as follows:

- We start by analyzing the performance of the binarized HG in Subsection 5.2.1 which provides the motivation as well as the baseline for our method.
- Then, we propose a series of architectural innovations in Subsections 5.2.2, 5.2.3, 5.2.4 and 5.2.5 (shown in Figs. 5.3b, 5.3c and 5.3d) each of which is evaluated and compared against the binarized residual block of Subsection 5.2.1.
- Finally, by combining ideas from these architectures, we propose the binary hierarchical, parallel and multi-scale block of Fig. 5.3e. Note that the proposed block is not a trivial combination of the aforementioned architectures but a completely new structure.

We note that all results for this Section were generated for the task of 3D face alignment using the newly introduced dataset from Chapter 4, LS3D-W-Balanced.

block type	AUC	# parameters
Bottleneck (real)	62.3%	3.5M
Bottleneck (binary)	46.2%	3.5M

Table 5.1 AUC@7% on LS3D-W-Balanced dataset for real-valued and binary bottleneck blocks within the HG network.

5.2.1 Binarized HG

The binarization is accomplished using:

$$\mathbf{I} * \mathbf{W} \approx (\text{sign}(\mathbf{I}) \circledast \text{sign}(\mathbf{W})) * \alpha, \tag{5.1}$$

where $\mathbf{I}$ is the input tensor, $\mathbf{W}$ represents the layer weights and $\circledast$ denotes the binary convolution operation which can be efficiently implemented with XNOR. $\alpha \in \mathbb{R}^+$ is a scaling factor computed as the average of the absolute weight values:

$$\alpha = \frac{1}{n} \|W\|_{\ell_1}, \tag{5.2}$$

We start from the original bottleneck blocks of the HG network and, following [79], we binarize them keeping only the first and last layers of the network real. See also Fig. 4.1. This is crucial, especially for the very last layer where higher precision is required for producing a dense output (heatmaps). Note that these layers account for less than 0.01% of the total number of parameters.

The performance of the original (real-valued) and the binarized HG networks can be seen in Table 5.1. We observe that binarization results in significant performance drop, noticing a large difference in performance which clearly indicates that the binary network has significant less representational power(in terms of number of unique values that it can represent). This drop in representational power is mostly caused by the limited number of unique 2D filters (2^{k^2}, where k is the kernel size). Since we make use of convolutional layers with a filter size of 1×1 and 3×3, the maximum number of unique combinations is $2^1 = 2$ and $2^9 = 512$ respectively. However, the actual filter is a 3D matrix. As such, assuming that we have have C_l filters in the l-th convolutional layer, we store in practice a 4D matrix of size $C_l \times C_{l-1} \times k \times k$, hence the number of total unique filters is $2^{k^2 C_{l-1}}$. Still, this is a significantly lower number of unique possibilities compared with the real valued case. We address this limitation and performance gap with a better architecture as detailed in the next four Subsections.

5.2.2 On the Width of Residual Blocks

The original bottleneck block of Fig. 5.3a is composed of 3 convolutional layers with a filter size of 1×1, 3×3 and 1×1, with the first layer having the role of limiting the width (i.e. the number of channels) of the second layer, thus greatly reducing the number of parameters inside

the module. However, it is unclear whether the idea of having a bottleneck structure will be also successful for the binary case, too. Due to the limited representational power of the binary layers, greatly reducing the number of channels might reduce the amount of information that can be passed from one layer to another, leading to lower performance.

To investigate this, we modify the bottleneck block by increasing the number of channels in the *thin* 3×3 layer from 128 to 256. By doing so, we match the number of channels from the first and last layer, effectively removing the "bottleneck", and increasing the amount of information that can be passed from one block to another. The resulting **wider** block is shown in Fig. 5.3b. Here, "wider"[1] refers to the increased number of channels over the initial *thin* layer.

As Table 5.2 illustrates, while this improves performance against the baseline, it also raises the memory requirements.

Conclusion: Widening the *thin* layer offers tangible performance improvement, however at a high computational cost.

5.2.3 On Multi-Scale Filtering

Small filters have been shown both effective and efficient [88, 96] with models being solely made up by a combination of convolutional layers with 3×3 and/or 1×1 filters [38, 39, 88]. For the case of real-valued networks, a large number of kernels can be learned. However, for the binary case, the number of possible unique convolutional kernels is limited to 2^k states only, where k is the size of the filter. Examples of such 3×3 learned filters are shown in Fig. 5.2.

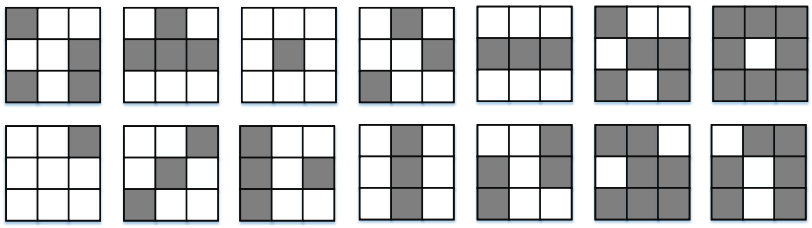


Fig. 5.2 Examples of learned 3×3 binary filters.

To address the limited representation power of 3×3 filters for the binary case, and similarly to [95], we largely depart from the block of Fig. 5.3b by proposing the multi-scale structure of Fig. 5.3c. Note that we implement our multi-scale approach using both larger filter sizes and max-pooling, which greatly increase the effective receptive field within the block. Also, because our goal is to analyze the impact of a multi-scale approach alone, we intentionally keep the number of parameters to a similar level to that of the original bottleneck block of Fig. 5.3a. To this end, we avoid a leap in the number of parameters, by (a) decomposing the 5×5 filters into two layers of 3×3 filters, and (b) by preserving the presence of *thin* layer(s) in the middle of the block.

[1] The term wider here strictly refers to a "moderate" increase in the number of channels in the *thin* layer (up to 256), effectively removing the "bottleneck". Except for the naming there is no other resemblance with [120] which performs a study of wide vs deep, using a different building block alongside a much higher number of channels (up to 2048) and without any form of quantization. A similar study falls outside the scope of our work.

Given the above, we split the input into two branches. The first (left) branch works at the same scale as the original bottleneck of Fig. 5.3a but has a 1×1 layer that projects the 256 channels into 64 (instead of 128) before going to the 3×3 one. The second (right) branch performs a multi-scale analysis by firstly passing the input through a max-pooling layer and then creating two branches, one using a 3×3 filter and a second one using a 5×5 decomposed into two 3×3. By concatenating the outputs of these two sub-branches, we obtain the remaining 64 channels (out of the 128 of the original bottleneck block). Finally, the two main branches are concatenated adding up to 128 channels, which are again back-projected to 256 with the help of a convolutional layer with 1×1 filters.

The accuracy of the proposed structure can be found in Table 5.2. We can observe a healthy performance improvement at little additional cost and similar computational requirements to the original bottleneck of Fig. 5.3a.

Conclusion: When designing binarized networks, multi-scale filters should be preferred.

5.2.4 On 1×1 Convolutions

In the previously proposed block of Fig. 5.3c, we opted to avoid an increase in the number of parameters, by retaining the two convolutional layers with 1×1 filters. In this Subsection, by relaxing this restriction, we analyze the influence of 1×1 filters on the overall network performance.

In particular, we remove all convolutional layers with 1×1 filters from the multi-scale block of Fig. 5.3c, leading to the structure of Fig. 5.3d. Our motivation to remove 1×1 convolutions for the binary case is the following: because 1×1 filters are limited to two states only (either 1 or -1) they have a very limited learning power. Due to their nature, they behave as simple filters deciding when a certain value should be passed or not. In practice, this allows the input to pass through the layer with little modifications, sometimes actually blocking "good features" and hurting the overall performance by a noticeable amount. This is particularly problematic for the task of landmark localization, where a high level of detail is required for successful localization.

Results reported in Table 5.2 show that by removing 1×1 convolutions, performance over the baseline is increased by almost 4%. Even more interestingly, the newly introduced block matches the performance of the one of Subsection 5.2.2, while having less parameters, which shows that the presence of 1×1 filters limits the performance of binarized CNNs.

Conclusion: The use of 1×1 convolutional filters on binarized CNNs has a detrimental effect on performance and should be avoided.

5.2.5 On Hierarchical, Parallel & Multi-Scale

Binary networks are even more sensitive to the problem of fading gradients [24, 79], and for our network we found that the gradients are up to 10 times smaller than those corresponding to its real-valued counterpart. To alleviate this, we design a new module which has the form of a hierarchical, parallel multi-scale structure allowing, for each resolution, the gradients to have 2

Block type	# params	AUC@7%
Bottleneck (original) (Fig. 5.3a)	3.5M	46.2%
Wider (Fig. 5.3b)	11.3M	52.9%
Multi-Scale (MS) (Fig. 5.3c)	4.0M	51.8%
MS without 1x1 filters (Fig. 5.3d)	9.3M	54.5%
Hierarchical, Parallel & MS (Ours, Final) (Fig. 5.3e)	**6.2M**	**54.6%**

Table 5.2 AUC-based comparison of different blocks on LS3D-W-Balanced dataset. # params refers to the number of parameters of the whole network.

different paths to follow, the shortest of them being always 1. The proposed block is depicted in Fig. 5.3e. Note that, in addition to better gradient flow, our design encompasses all the findings from the previous Subsections: (a) no convolutional layers with 1×1 filters should be used, (b) the block should preserve its width as much as possible (avoiding large drops in the number of channels), and (c) multi-scale filters should be used.

Contrary to the blocks described in Subsections 5.2.2 - 5.2.4, where the gradients may need to pass through two more layers before reaching the output of the block, in the newly proposed module, each convolutional layer has a direct path that links it to the output, so that at any given time and for all the layers within the module the shortest possible path is equal to 1. The presence of a hierarchical structure inside the module efficiently accommodates larger filters (up to 7×7), decomposed into convolutional layers with 3×3 filters. Furthermore, our design avoids the usage of an element-wise summation layer as for example in [106, 95], further improving the gradient flow and keeping the complexity under control.

As we can see in Table 5.2, the proposed block matches and even outperforms the block proposed in Section 5.2.3 having far less parameters.

Conclusion: Good gradient flow and hierarchical multi-scale filtering are crucial for high performance without excessive increase in the parameters of the binarized network.

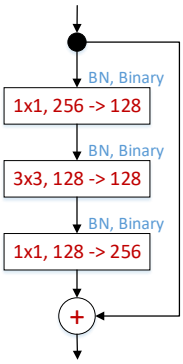
5.3 Proposed vs Bottleneck

In this Section, we attempt to make a fair comparison between the performance of the proposed block (**Ours, Final**, as in Fig. 5.3e) against that of the original bottleneck module (Fig. 5.3a) by taking two important factors into account:

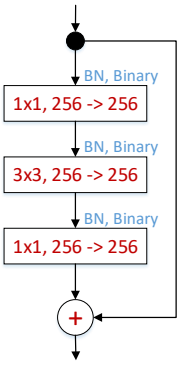
- Both blocks should have the same number of parameters.
- The two blocks should be compared for the case of binary but also real-valued networks.

With this in mind, in the following Sections, we show that:

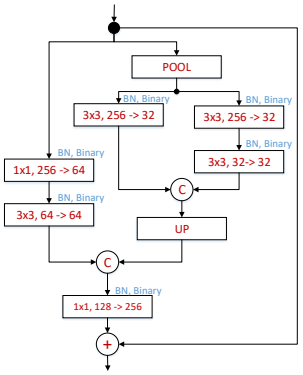
- The proposed block largely outperforms a bottleneck with the same number of parameters for the binary case.



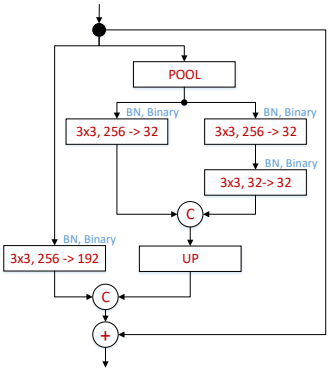
(a) The **Original Bottleneck** block with pre-activation, as defined in [39]. Its binarized version is described in Section 5.2.1.



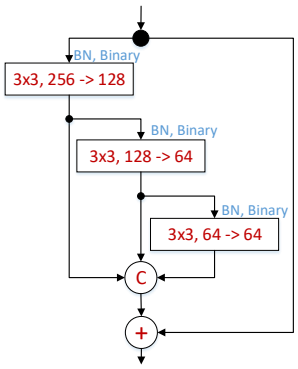
(b) The **Wider** version of (a) produced by increasing the number of filters in the second layer. See Subsection 5.2.2.



(c) Largely departing from (b), this block consists of **Multi-Scale (MS)** filters for analyzing the input at multiple scales. See Subsection 5.2.3.



(d) A variant of the MS block introduced in (c) after removing all convolutional layers with 1×1 filters (**MS Without** 1×1 **filters**). See Subsection 5.2.3.



(e) The proposed **Hierarchical, Parallel & MS** (denoted in this chapter as (**Ours, final**) block incorporates all ideas from (b), (c) and (d) with an improved gradient flow. See Subsection 5.2.5

Fig. 5.3 Different types of blocks described and evaluated. Our best performing block is shown in figure (e). A layer is depicted as a rectangular block containing: its filter size, number of input channels and the number of output channels). "C" - denotes concatenation operation and "+" an element-wise sum.

Layer type	# parameters	AUC
Bottleneck (Original) (Fig. 5.3a)	3.5M	46.2%
Wider (Fig. 5.3b)	11.3M	52.9%
Bottleneck (wider) + no 1×1	5.8M	51.8%
(Ours, Final) (Fig. 5.3e)	**6.2M**	**54.6%**

Table 5.3 AUC-based performance on LS3D-W-Balanced dataset for binary blocks: the # parameters of the original bottleneck are increased to match the # parameters of the proposed block. This firstly gives rise to the Wider block and its variant without the 1×1 Convolutions.

Layer type	# parameters	AUC
Bottleneck (original)	3.5M	46.2%
(Ours, Final) (Fig. 5.1b)	**4.0M**	**51.5%**

Table 5.4 AUC-based performance on LS3D-W-Balanced dataset for binary blocks: the # parameters of the proposed block are decreased to match the # parameters of the bottleneck.

- The proposed block also outperforms a bottleneck with the same number of parameters for the real case but in this case the performance difference is smaller.

We conclude that, for the real case, increasing the number of parameters (by increasing width) results in performance increase; however this is not the case for binary networks where a tailored design as the one proposed here is needed.

5.3.1 Binary

To match the number of parameters between the proposed and bottleneck block, we follow two paths. Firstly, we increase the number of parameters of the bottleneck: (a) a first way to do this is to make the block wider as described in Section 5.2.2. Note that in order to keep the number or input-output channels equal to 256, the resulting block of Fig. 5.3b has a far higher number of parameters than the proposed block. Despite this, the performance gain is only moderate (see Section 5.2.2 and Table 5.3). (b) Because we found that the 1×1 convolutional layers have detrimental effect to the performance of the Multi-Scale block of Fig. 5.3c, we opted to remove them from the bottleneck block, too. To this end, we modified the Wider module by (a) removing the 1×1 convolutions and (b) halving the number of parameters in order to match the number of parameters of the proposed block. The results in Table 5.3 clearly show that this modification is helpful but far from being close to the performance achieved by the proposed block.

Secondly, we decrease the number of parameters in the proposed block to match the number of parameters of the original bottleneck. This block is shown in Fig. 5.1b. To this end, we reduced the number of input-output channels of the proposed block from 256 to 192 so that the number of channels in the first layer are modified from [256 $\rightarrow$ 128, 3×3] to [192$\rightarrow$96, 3×3], in the second layer from [128$\rightarrow$64, 3×3] to [96$\rightarrow$48, 3×3] and in the third layer from

Layer type	# parameters	AUC
Bottleneck (wider)	7.0M	66.1%
(Ours, Final)	**6.2M**	**69.6%**

Table 5.5 AUC-based performance on LS3D-W-Balanced dataset for real-valued blocks: Our block is compared with a wider version of the original bottleneck so that both blocks have similar # parameters.

[64→64, 3×3] to [48→48, 3×3]. Notice, that even in this case, the proposed binarized module outperforms the original bottleneck block by more than 6% (in absolute terms) while both have very similar number of parameters (see Table 5.4).

5.3.2 Real

While the proposed block was derived from a binary perspective, Table 5.5 shows that a significant performance gain is also observed for the case of real-valued networks. In order to quantify this performance improvement and to allow for a fair comparison, we increase the number of channels inside the original bottleneck block so that both networks have the same depth and a similar number of parameters. Even in this case, our block outperforms the original block although the gain is smaller than that observed for the binary case. We conclude that for real-valued networks performance increase can be more easily obtained by simply increasing the number of parameters, but for the binary case a better design is needed as proposed in this work.

5.4 Ablation studies

In this Section, we present a series of other architectural variations and their effect on the performance of our binary network. All reported results are obtained using the proposed block of Fig. 5.3e coined **Ours, Final**. We focus on the effect of augmentation and different losses which are novel experiments not reported in [79], and then comment on the effect of pooling, ReLUs and performance speed-up.

Is Augmentation required? Recent works have suggested that binarization is an extreme case of regularization [23, 24, 67]. In light of this, one might wonder whether data augmentation is still required. Table 5.6 shows that in order to accommodate the presence of new poses and/or scale variations, data augmentation is very helpful providing a large increase (4%) in performance.

The effect of loss. We trained our binary network to predict a set of heatmaps, one for each landmark [99]. To this end, we experimented with two types of losses: the first one places a Gaussian around the correct location of each landmark and trains using a pixel-wise L2 loss [99]. However, the gradients generated by this loss are usually small even for the case of a real-valued network. Because binarized networks tend to amplify this problem, as an alternative, we also experimented with the Sigmoid cross-entropy pixel-wise loss typically used for detection

Layer type	# parameters	AUC
(Ours, Final) (No Aug.)	6.2M	52.3%
(Ours, Final) + Aug.	**6.2M**	**54.6%**

Table 5.6 The effect of using augmentation when training our binary network in terms of AUC-based performance on LS3D-W-Balanced dataset.

Layer type	# parameters	AUC
(Ours, Final) + L2	6.2M	53.7%
(Ours, Final) + Sigmoid	**6.2M**	**54.6%**

Table 5.7 The effect of using different losses (Sigmoid vs L2) when training our binary network in terms ofAUC-based performance on LS3D-W-Balanced dataset.

tasks [123]. We found that the use of the Sigmoid cross-entropy pixel-wise loss increased the gradients by 10-15x (when compared to the L2 loss), offering a 1% improvement (see Table 5.7), after being trained for the same number of epochs.

Pooling type. In the context of binary networks, and because the output is restricted to 1 and -1, max-pooling might result in outputs full of 1s only. To limit this effect, we placed the activation function before the convolutional layers as proposed in [39, 79]. Additionally, we opted to replace max-pooling with average pooling. However, this leads to slightly worse results (see Table 5.8). In practice, we found that the use of blocks with pre-activation suffices and that the ratio of 1 and -1 is close to 50% even after max-pooling.

With or without ReLU. Because during the binarization process all ReLU layers are replaced with the Sign function, one might wonder if ReLUs are still useful for the binary case. Our findings are in line with the ones reported in [79]. By adding a ReLU activation after each convolutional layer, we observe a 3% performance improvement (see Table 5.9), which can be attributed to the added non-linearity, particularly useful for training very deep architectures.

Performance. In theory, by replacing all floating-point multiplications with bitwise XOR and making use of the SWAR (Single instruction, multiple data within a register) [79, 24], the number of operations can be reduced up to 32x when compared against the multiplication-based convolution. However, in our tests, we observed speedups of up to 3.5x, when compared against cuBLAS, for matrix multiplications, a result being in accordance with those reported in [24]. We note that we did not conduct experiments on CPUs. However, given the fact that we used the

Layer type	# parameters	AUC
(Ours, Final) + Average	6.2M	52.8%
(Ours, Final) + Max	**6.2M**	**54.6%**

Table 5.8 The effect of using different pooling methods when training our binary network in terms of AUC-based performance on LS3D-W-Balanced dataset.

Layer type	# parameters	AUC
(Ours, Final)	6.2M	54.6%
(Ours, Final) + ReLU	**6.2M**	**57.1%**

Table 5.9 The effect of using ReLUs when training our binary network in terms of AUC-based performance on LS3D-W-Balanced dataset.

same method for binarization as in [79], similar improvements in terms of speed, of the order of 58x, are to be expected: as the real-valued network takes 0.67 seconds to do a forward pass on a i7-3820 using a single core, a speedup close to x58 will allow the system to run in real-time.

In terms of memory compression, by removing the biases, which have minimum impact (or no impact at all) on performance, and by grouping and storing every 32 weights in one variable, we can achieve a compression rate of 39x when compared against the single precision counterpart of Torch7. See also Fig. 5.4.

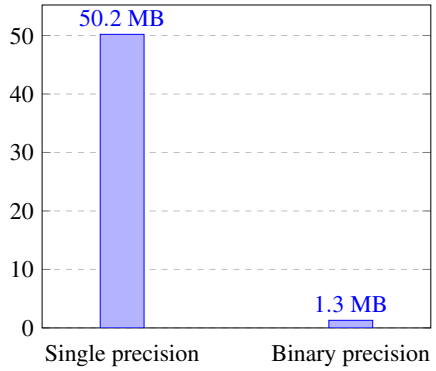


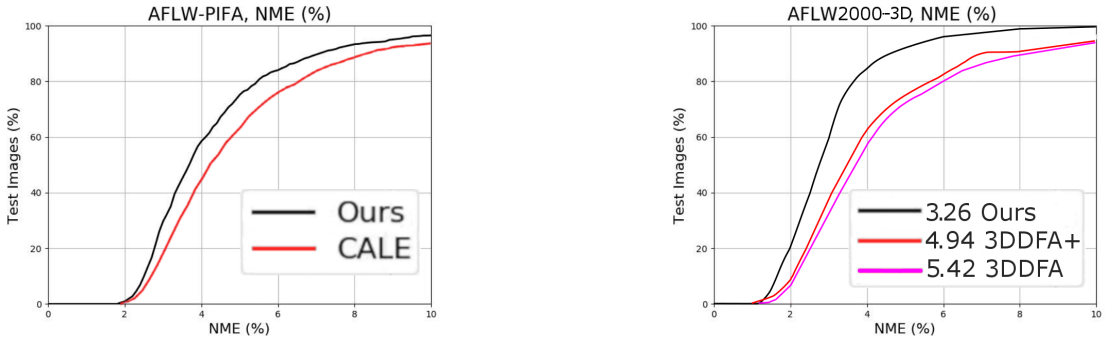
Fig. 5.4 Memory compression ratio. By binarizing the weights and removing the biases, we achieve a compression rate of 39x when compared against the single precision model.

5.5 Additional face alignment experiments

In addition to the experiments of the previous Section, in this Section, we compare our method against a few state-of-the-art methods for 3D face alignment. Our final system comprises a single HG network but replaces the real-valued bottleneck block used in [69] with the proposed binary, parallel, multi-scale block trained with the improvements detailed in Section 5.4.

We used three very challenging datasets for large pose face alignment, namely AFLW [57], AFLW-PIFA [53], and AFLW2000-3D [132]. The evaluation metric is the Normalized Mean Error (NME) [53].

AFLW is a large-scale face alignment dataset consisting of 25,993 faces annotated with up to 21 landmarks. The images are captured in arbitrary conditions exhibiting a large variety of poses and expressions. As Table 5.10 shows, our binarized network outperforms the state-of-the-art methods of [77] and [78], both of which use large real-valued CNNs.



(a) (b)

Fig. 5.5 Cumulative error curves (a) on AFLW-PIFA, evaluated on all 34 points (CALE is the method of [7]), (b) on AFLW2000-3D on all points computed on a random subset of 696 images equally represented in $[0°,30°]$, $[30°,60°]$, $[60°,90°]$ (see also [132]).

Method	[0°,30°]	[30°,60°]	[60°,90°]	mean
HyperFace [77]	3.93	4.14	4.71	4.26
AIO [78]	2.84	2.94	3.09	2.96
Ours	**2.77**	**2.86**	**2.90**	**2.85**

Table 5.10 NME-based (%) comparison on AFLW test set. The evaluation is done on the test set used in [78].

AFLW-PIFA [53] is a gray-scale subset of AFLW [57], consisting of 5,200 images (3,901 for training and 1,299 for testing) selected so that there is a balanced number of images for yaw angle in $[0°,30°]$, $[30°,60°]$ and $[60°,90°]$. All images are annotated with 34 points from a 3D perspective. Fig. 5.5a and Tables 5.11 and 5.12 show our results on AFLW-PIFA. When evaluated on both visible and occluded points, our method improves upon the current best result of [7] (which uses real weights) by more than 10%.

AFLW2000-3D is a subset of AFLW re-annotated by [132] from a 3D perspective with 68 points. We used this dataset only for evaluation. The training was done using the first 40,000 images from 300-W-LP [132]. As Fig. 5.5b shows, on AFLW2000-3D, the improvement over the state-of-the-art method of [132] (real-valued) is even larger. As further results in Fig. 5.13 show, while our method improves over the entire range of poses, the gain is noticeably higher for large poses ($[60° - 90°]$), where we outperform [132] by more than 40%.

PIFA [53]	RCPR [14]	PAWF [54]	CALE [7]	Ours
8.04	6.26	4.72	**2.96**	3.02

Table 5.11 NME-based (%) comparison on AFLW-PIFA evaluated on visible landmarks only. The results for PIFA, RCPR and PAWF are taken from [54].

CALE [7]	Ours
4.97	**4.47**

Table 5.12 NME-based (%) based comparison on AFLW-PIFA evaluated on all 34 points, both visible and occluded.

5.5.1 Training

All 3D face alignment models were trained on 300-W-LP (holding 10% of the data for validation) from scratch following the algorithm described in [79] and using rmsprop [97]. The initialization was done as in [38]. We randomly augmented the data with rotation (between -40^o and 40^o degrees), flipping and scale jittering (between 0.7 and 1.3). We trained the network for 55 epochs, dropping the learning rate four times, from 2.5e-4 to 5e-5. The input was normalized between 0 and 1 and all described networks were trained using the binary cross-entropy loss, defined as:

$$l = \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{W}\sum_{j=1}^{H} p_{ij}^{n}\log\widehat{p}_{ij}^{n} + (1-p_{ij}^{n})\log(1-\widehat{p}_{ij}^{n}), \tag{5.3}$$

where p_{ij}^{n} denotes the ground truth confidence map of the $n-$th landmark at the output pixel location (i, j) and $\widehat{p}_{ij}^{n}$ is the corresponding predicted output at the same location.

In terms of wall-clock training time, the real-valued network (with 6.2M parameters) takes around 12 hours to train on a single nVidia 1080Ti GPU. Due to the additional quantization operations, such as weight and input normalization and sign-based quantization, and the fact that the gradients are kept real, the binary network requires up to 50% more time to train.

The models were implemented with Torch7 [20].

Method	[0,30]	[30,60]	[60,90]	Mean
RCPR(300-W) [14]	4.16	9.88	22.58	12.21
RCPR(300-W-LP) [14]	4.26	5.96	13.18	7.80
ESR(300-W) [18]	4.38	10.47	20.31	11.72
ESR(300-W-LP) [18]	4.60	6.70	12.67	7.99
SDM(300-W) [107]	3.56	7.08	17.48	9.37
SDM(300-W-LP) [107]	3.67	4.94	9.76	6.12
3DDFA [132]	3.78	4.54	7.93	5.42
3DDFA+SDM [132]	3.43	4.24	7.17	4.94
Ours	**2.47**	**3.01**	**4.31**	**3.26**

Table 5.13 NME-based (%) based comparison on AFLW2000-3D evaluated on all 68 points, both visible and occluded. The results for RCPR, ESR and SDM are taken from [132].

5.6 Advanced block architectures

In this section, we explore the effectiveness of two architectural changes applied to our best performing block (**Ours, final**), namely varying its depth and its cardinality. Again, we used the standard training-validation partition of LS3D-W-Balanced.

5.6.1 On the depth of the proposed block

To further explore the importance of the multi-scale component in the overall structure of the proposed block, we gradually increase its depth and as a result, the number of its layers, as shown in Fig. 5.7b. The advantage of doing this is twofold: (a) it increases the receptive field within the block, and (b) it analyses the input simultaneously at multiple scales. We ensure that by doing so the number of parameters remains (approximately) constant. To this end, we halve the number of channels of the last layer at each stage. In the most extreme case, the last layer will have a single channel. Because, the representational power of such a small layer is insignificant, in practice we stop at a minimum of 4, which corresponds to a depth equal to 8. The results, reported in Fig. 5.7b, show that the performance gradually improves up to 55.1% for a depth equal to 6, and then, further on, it saturates and eventually gradually degrades as the depth increases.

Conclusion: The depth of the multi-scale component is an important factor on the overall module performance. Increasing it, up to a certain point, is beneficial and can further improve the performance at no additional cost.

5.6.2 On the cardinality of the proposed block

Inspired by the recent innovations of [106] for real-valued networks, in this section we explore the behavior of an increased cardinality (defined as in [106] as the size of the set of transformations) when applied to our binary hierarchical, parallel & multi-scale block.

Starting again from our block of Fig. 5.3e, we replicate its structure C times making the following adjustments in the process: (1) While the number of input channels of the first layer remains the same, the output and the input of the subsequent layers are reduced by a factor of C, and (2) the output of the replicated blocks is recombined via concatenation. The final module structure is depicted in Fig. 5.8a.

The full results with respect to the network size and the block cardinality (ranging from 1 to 16) are shown in Fig. 5.8b. Our findings are that increasing the block cardinality, provides good improvement also for the case of binary networks for the task of face alignment. In particular, when incorporated into the structure of our block with a similar number of parameters, the module out-performs by 1-2% compared to the block of similat size. As the number of parameters decreases, the performance gain tend to increase.

Conclusion: For the binary case, further increasing the block cardinality can help especially in the low number of parameters regime.

5.7 Improved network architectures

In all previous sections, we investigated the performance of the various blocks by incorporating them into a single hourglass network, i.e. by keeping the network architecture fixed. In this section, we explore a series of architectural changes applied to the overall network structure. First, inspired by [80], we simplify the HG model, improving its performance without sacrificing accuracy for the binary case. Then, we study the effect of stacking multiple networks together and analyze their behavior.

5.7.1 Improved HG architecture

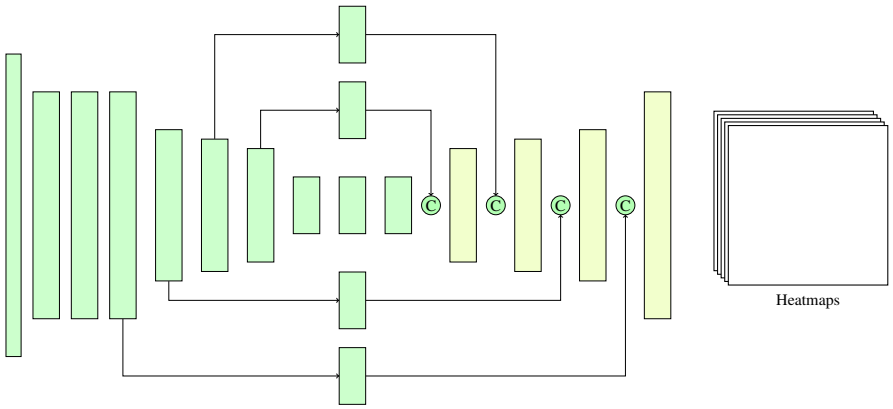


Fig. 5.6 Improved, U-Net inspired, HG architecture. The dark-green modules were left unchanged, while for the light-green ones we doubled the number of their input channels from 256 to 512.

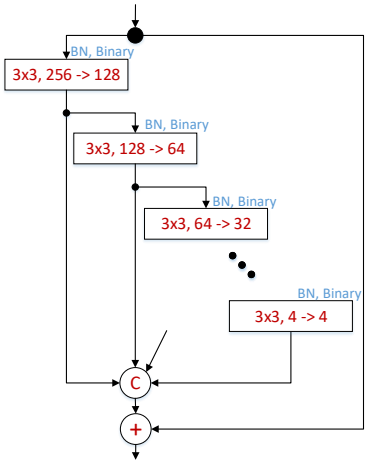
Motivated by the findings of Subsection 5.2.5 that shed light on the importance of the gradient flow and suggested that skip connections with shorter paths should be used where possible, we adopt a similar approach to the overall HG architecture.

In particular, to improve the overall gradient flow, we removed the residual blocks in the upsampling branches that are tasked with the "injection" of high resolution information into the later stages of the network. To adjust to that change, the number of input channels of the first layer from the modules that are immediately after the point where the branch is merged via *concatenation* is increased by two times (to accommodate to the increase in the number of channels). The resulting architecture is depicted in Fig. 5.6. A similar approach, with up-sampling skip connections, was used in the U-Net architecture [80].

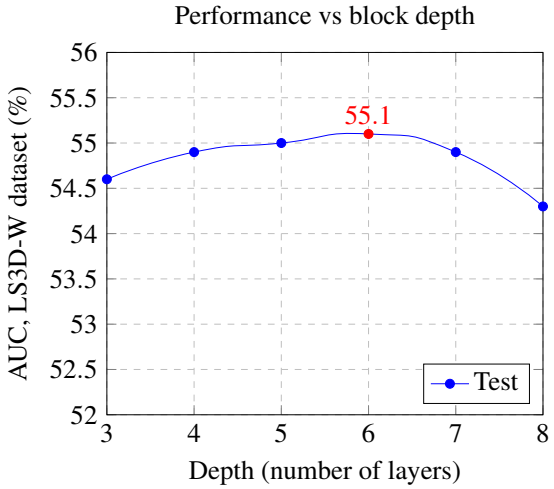
The results, reported in Table 5.14, show that by removing the residual blocks from the upsampling branches, the performance, over the baseline HG is increased by 0.5%, further solidifying the importance of the gradient flow in the performance of binary networks. Furthermore, due to the decrease in the number of layers and parameters, an up to 20% speedup is observed. The network is trained using the same procedure described previously, for 55 epochs.

Network architecture	# parameters	AUC
HG (Fig. 4.1)	6.2M	54.6%
Improved HG (Fig. 5.6)	**5.8M**	**55.1%**

Table 5.14 Comparison between HG and Improved HG on the LS3D-W dataset. Both networks are built with our proposed binarized block.

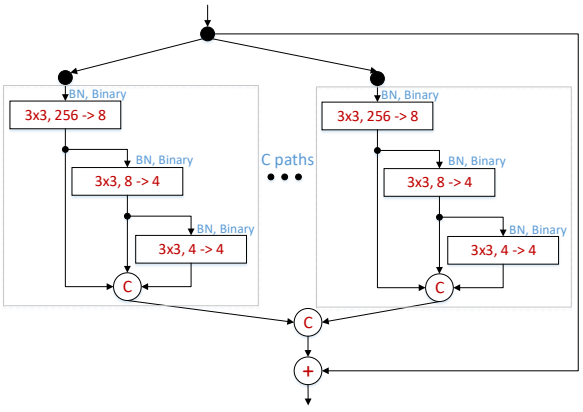


(a) Depth vs AUC-based performance on the LS2D-W dataset.

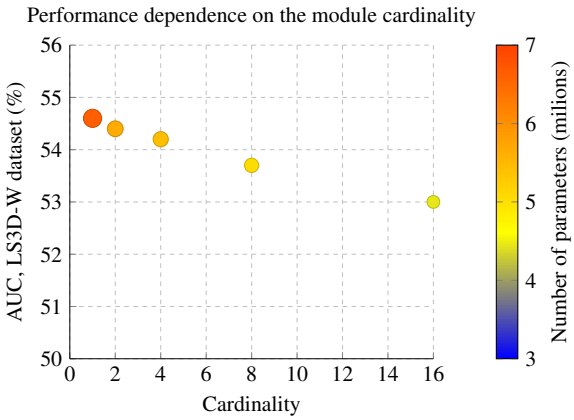


(b) Depth vs AUC-based performance on the LS2D-W dataset.

Fig. 5.7 The effect of varying the depth of the proposed binary block on performance. While in general fewer weights result in a faster network, due to the the introduction of additional layers, in practice, the network experiences a small slowdown.



(a) ResNetXt-like extension of **(Ours, final)** binary block. C represents the cardinality of the block. See also Subsection 5.6.2.



(b) Cardinality vs AUC-based performance on the LS2D-W dataset. Notice how the efficiency (the ratio between the number of parameters and AUC) decreases as we increase the block cardinality.

Fig. 5.8 The effect of varying the cardinality of the proposed binary block on performance.

# stacks	# parameters	AUC
1	6.2M	54.6%
2	11.0M	61.0%
3	17.8M	63.9%

Table 5.15 Accuracy of stacked networks on LS3D-W dataset. All networks are built with our proposed binarized block.

5.7.2 Stacked Binarized HG networks

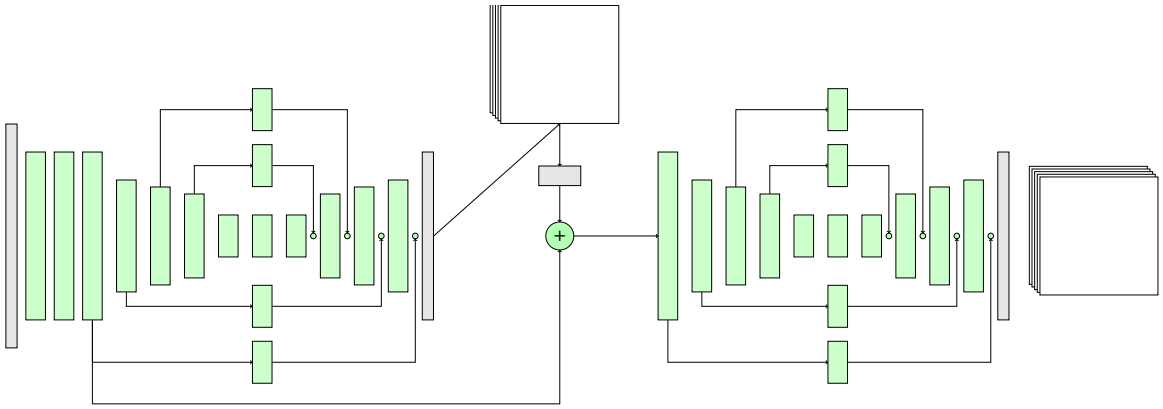


Fig. 5.9 A two-stack binarized HG. All blocks are binarized, except for the very first and last layers showed in red colour. The network takes as input an RGB facial image at a resolution of 256×256px and produces k heatmaps, one for each predicted keypoint, at a resolution of 64×64px.

Network stacking has been shown to be beneficial in terms of performance (see Chapter 4) when real-valued models are used. In this subsection, we explore whether the same holds for the binary case.

Following [69], we stack and interconnect the networks as follows: The first network takes as input the RGB image and outputs a set of N heatmaps. The next network in the stack takes as input the sum of: (1) the input to the previous network, (2) the projection of the previously predicted heatmaps, and (3) the output of the last but one block from the previous level. The resulting network for a stack of two is shown in Fig. 5.9.

As the results of Table 5.15 show, network stacking for the binary case behaves to some extent similarly to the real-valued case, however the gains from one stage to another are smaller, and performance seems to saturate faster. We believe that the main reason for this is that for the case of binary networks, activations are noisier especially for the last layers of the network. As such the feature maps for the binary case are more noisy and blurry as we move on to the last layers of the network. As network stacking relies on features from the earlier networks of the cascade and as these are noisy, we conclude that this has a negative impact on the overall network's performance.

Training. To speedup the training process, we trained the stacked version in a sequential manner. First, we trained the first network until convergence, then we added the second one on top of it, freezing its weights and training the second one. The process is repeated until all networks are added. Finally, the entire stack is trained jointly for 50 epochs.

5.8 Additional experiments

In this section, we further show that the proposed block generalizes well producing consistent results across various datasets and tasks. To this end, we report results on the task of face parsing, also known as semantic facial part segmentation, which is the problem of assigning a categorical label to every pixel in a facial image. We constructed a dataset for facial part segmentation by joining together the 68 ground truth landmarks (originally provided for face alignment) to fully enclose each facial component. In total, we created seven classes: skin, lower lip, upper lip, inner mouth, eyes, nose and background. Fig. 5.10 shows an example of a ground truth mask. We trained the network on the 300-W dataset (approximately 3,000 images) and tested it on the 300-W competition test set, both Indoor&Outdoor subsets (600 images), using the same procedure described in Section 7.

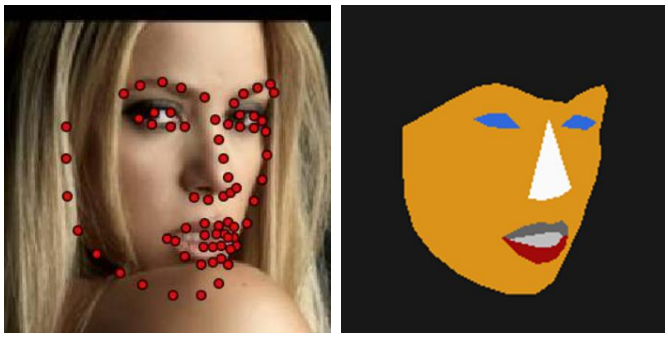


Fig. 5.10 Example of a ground truth mask (right) produced by joining the 68 ground truth landmarks (left). Each colour denotes one of the seven classes.

Architecture. We reused the same architecture for landmark localization, changing only the last layer in order to accommodate the different number of output channels (from 68 to 7). We report results for three different networks of interest: (a) a real-valued network using the original bottleneck block (called "Real, Bottleneck"), (b) a binary network using the original bottleneck block (called "Binary, Bottleneck"), and (c) a binary network using the proposed block (called "Binary, Ours"). To allow for a fair comparison, all networks have a similar number of parameters and depth. For training the networks, we used the Log-Softmax loss [66].

Results. Table 5.16 shows the obtained results reported in terms of pixel accuracy, mean accuracy and mean IU, defined as:

$$pixel\ accuracy : \frac{\sum_i n_{ii}}{\sum_i t_i} \tag{5.4}$$

Network type	pixel acc.	mean acc.	mean IU
Real, bottleneck	97.98%	77.23%	69.29%
Binary, bottleneck	97.41%	70.35%	62.49%
Binary, Ours	97.91%	76.02%	68.05%

Table 5.16 Results on 300-W (Indoor&Outdoor). The pixel acc., mean acc. and mean IU are computed as in [66].

$$mean\ accuracy : \frac{1}{n_{cl}} \sum_i \frac{n_{ii}}{t_i} \tag{5.5}$$

$$mean\ IU : \frac{1}{n_{cl}} \sum_i \frac{n_{ii}}{(t_i + \sum_j n_{ji} - n_{ii})}, \tag{5.6}$$

where n_{ij} is the number of pixels of class i predicted to belong to class j, n_{cl} represents the number of classes present in the ground truth and $t_i = \sum_j n_{ij}$ is the total number of pixels of class i.

Similarly to our face alignment experiments, we observe that the binarized network based on the proposed block significantly outperforms a similar-sized network constructed using the original bottleneck block, almost matching the performance of the real-valued network. Most of the performance improvement is due to the higher representation/learning capacity of our block, which is particularly evident for difficult cases like unusual poses, occlusions or challenging lighting conditions. For visual comparison, see Fig. 5.12.

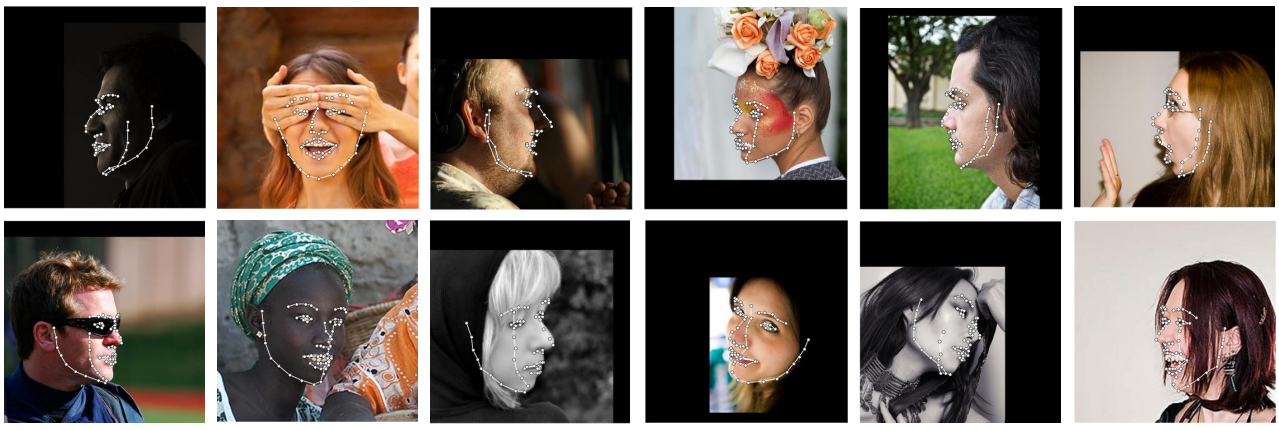


Fig. 5.11 Fitting examples produced by our binary network on AFLW2000-3D dataset. Notice that our method copes well with extreme poses, facial expressions and lighting conditions.

Fig. 5.12 Qualitative results on 300-W (Indoor&Outdoor). Observe that the proposed binarized network significantly outperforms the original binary one, almost matching the performance of the real-valued network.

Chapter 6

Super-FAN: Integrated facial landmark localization and super-resolution of real-world low resolution faces in arbitrary poses with GANs

Fig. 6.1 A few examples of visual results produced by our system on real-world low resolution faces from WiderFace.

In this Chapter we describe a novel method that simultaneously addresses two challenging tasks: (a) improving the quality of low resolution facial images and (b) accurately locating the facial landmarks on such poor resolution images. Attempting to address both tasks simultaneously is really a chicken-and-egg problem: On one hand, being able to detect the facial landmarks has already been shown beneficial for face super-resolution [131, 110]; however how to accomplish this for low resolution faces in arbitrary poses is still an open problem (see Chapter 4). On the other hand, if one could effectively super-resolve low quality and low resolution faces across the whole spectrum of facial poses, then facial landmarks can be localized with high accuracy.

To this end, we propose Super-FAN: the very first end-to-end system that addresses both tasks simultaneously, i.e. both improves face resolution and detects the facial landmarks. The novelty or Super-FAN lies in incorporating structural information in a GAN-based super-resolution algorithm via integrating a sub-network for face alignment through heatmap regression and

optimizing a novel heatmap loss. We also illustrate the benefit of training the two networks jointly by reporting good results not only on frontal images (as in prior work) but on the whole spectrum of facial poses, and not only on synthetic low resolution images (as in prior work) but also on real-world images. Finally, both quantitatively and qualitatively we show large improvement over the state-of-the-art for both face super-resolution and alignment, showing for the first time good results on real-world low resolution images like the ones of Fig. 6.1.

The contributions of this Chapter have been published at CVPR 2018 in [12].

6.1 Datasets

In this section we briefly describe the train and test split used for the experiments reported in the current chapter. For a in-depth description of the datasets see Section 2.3.

In order to systematically evaluate face super-resolution across pose, the training dataset was constructed from 300-W-LP [132], AFLW [57], Celeb-A [65] and a portion of LS3D-W balanced [11] (10% of the data was held out for validation). For testing, we used the remaining images from LS3D-W balanced, in which each pose range ($[0° - 30°], [30° - 60°], [60° - 90°]$) is equally represented.

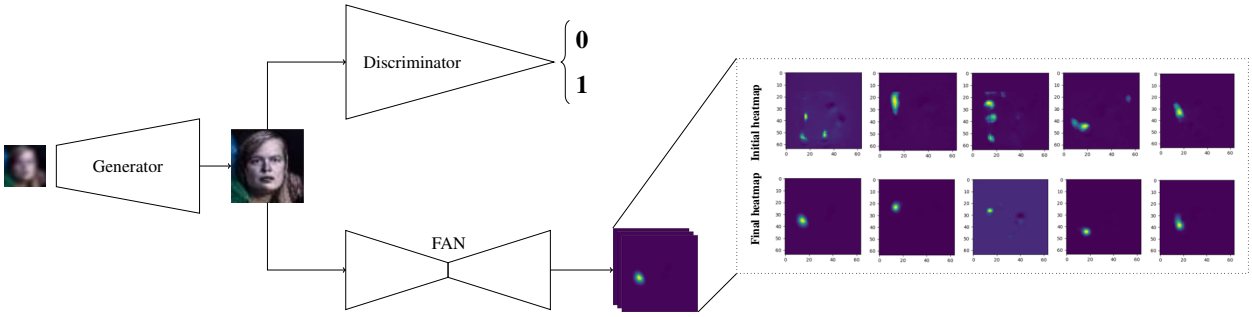


Fig. 6.2 The proposed Super-FAN architecture comprises three connected networks: the first network is a newly proposed Super-resolution network (see sub-section 6.2.1). The second network is a WGAN-based discriminator used to distinguish between the super-resolved and the original HR image (see sub-section 6.2.2). The third network is FAN, a face alignment network for localizing the facial landmarks on the super-resolved facial image and improving super-resolution through a newly-introduced heatmap loss (see sub-section 6.2.3).

6.2 Method

In this section, we describe the proposed architecture comprising of three connected networks: the first network is a Super-resolution network used to super-resolve the LR images. The second network is a discriminator used to distinguish between the super-resolved and the original HR images. The third network is FAN: the face alignment network for localizing the facial landmarks on the super-resolved facial images. Note that at test time the discriminator is not used. Overall, we call our network Super-FAN. See Fig. 6.2

Notably, for super-resolution, we propose a new architecture, shown in Fig. 6.3a, and detailed, along with the loss functions to train it, in sub-section 6.2.1. Our discriminator, based on Wasserstein GANs [2], is described in sub-section 6.2.2. Our integrated FAN along with our newly-introduced heatmap regression loss for super-resolution is described in sub-section 6.2.3. Sub-section 6.2.4 provides the overall loss for training Super-FAN. Finally, sub-section 6.2.5 describes the complete training procedure.

6.2.1 Super-resolution network

In this section, we propose a new residual-based architecture for super-resolution, inspired by [61], and provide the intuition and motivation behind our design choices. Our network as well as the one of [61] are shown in Figs. 6.3a and 6.3b, respectively. Their differences are detailed below. Following recent work [118, 117], the input and output resolutions are 16×16 and 64×64, respectively.

Per-block layer distribution. The architecture of [61], shown in Fig. 6.3b, uses 16, 1 and 1 blocks (layers) operating at the original, twice the original, and 4 times the original resolution, respectively; in particular, 16 blocks operate at a resolution 16×16, 1 at 32×32 and another 1 at 64×64. Let us denote this architecture as $16-1-1$. We propose a generalized architecture of the form $N_1 - N_2 - N_3$, where N_1, N_2 and N_3 are the number of blocks used at the original, twice the original, and 4 times the original resolution, respectively. As opposed to the architecture of [61] where most of the blocks (i.e. 16) work at the input resolution, we opted for a more balanced distribution: 12-3-2, shown in Fig. 6.3a. Our motivation behind this change is as follows: since the main goal of the network is to super-resolve its input via hallucination, using only a single block at higher resolutions (as in [61]) is insufficient for the generation of sharp details, especially for images found in challenging scenarios (e.g. Fig. 6.1).

Building block architecture. While we experimented with a few variants of residual blocks [38, 39], similarly to [51, 61], we used the one proposed in [36]. The block contains two 3×3 convolutional layers, each of them followed by a batch normalization layer [46]. While [61] uses a PReLU activation function, in our experiments, we noticed no improvements compared to ReLU, therefore we used ReLUs throughout the network. See Fig. 6.3a.

On the "long" skip connection. The SR-ResNet of [61] groups its 16 modules operating at the original resolution in a large block, equipped with a skip connection that links the first and the last block, in an attempt to improve the gradient flow. We argue that the resolution increase is a gradual process in which each layer should improve upon the representation of the previous one, thus the infusion of lower level futures will have a small impact on the overall performance. In practice, and at least for our network, we found very small gains when using it.

Pixel and perceptual losses

Pixel loss. Given a low resolution image I^{LR} (of resolution 16×16px) and the corresponding high resolution image I^{HR} (of resolution 64×64px), we train a generator network G_{θ_G} parameterized

by $\theta_G = \{W_{1:L}; b_{1:L}\}$ where W_L and b_L denotes the weights and respectively the biases of the Lth layer. We used the pixel-wise MSE loss to minimize the distance between the high resolution and the super-resolved image. It is defined as follows:

$$l_{pixel} = \frac{1}{r^2WH} \sum_{x=1}^{rW} \sum_{y=1}^{rH} (I_{x,y}^{HR} - G_{\theta_G}(I^{LR})_{x,y})^2, \tag{6.1}$$

where W and H denote the size of I^{LR} and r is the upsampling factor (set to 4 in our case).

Perceptual loss. While the pixel-wise MSE loss achieves high peak signal-to-noise ratio (PSNR) values, it often results in images which lack fine details, are blurry and unrealistic (see Fig. 6.4). To address this, in [51, 61], a perceptual loss is proposed in which the super-resolved image and the original image must also be close in feature space. While [61] defines this loss over the activations of layer 5_4 (the one just before the FC layers) of VGG-19 [88], we instead used a combination of low, middle and high level features computed after the B1, B2 and B3 blocks of ResNet-50 [38]. The loss over the ResNet features at a given level i is defined as:

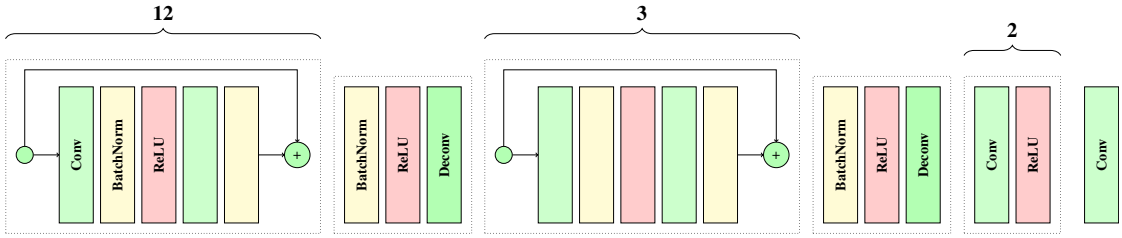
$$\begin{aligned} l_{feature/i} = \frac{1}{W_iH_i} \sum_{x=1}^{W_i} \sum_{y=1}^{H_i} (\phi_i(I^{HR})_{x,y} \\ - \phi_i(G_{\theta_G}(I^{LR}))_{x,y})^2, \end{aligned} \tag{6.2}$$

where ϕ_i denotes the feature map obtained after the last convolutional layer of the $i-$th block and W_i, H_i its size.

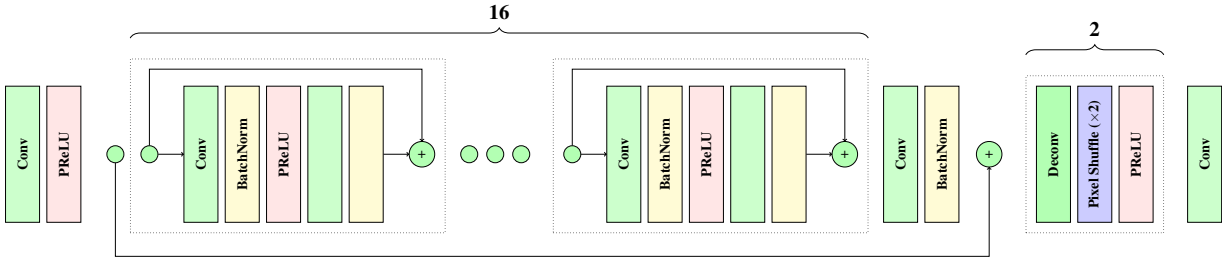
6.2.2 Adversarial network

The idea of using a GAN [34] for face super-resolution is straightforward: the generator G in this case is the super-resolution network which via a discriminator D and an adversarial loss is enforced to produce more realistic super-resolved images lying in the manifold of facial images. Prior work in image super-resolution [61] used the GAN formulation of [76]. While in our work, we do not make an attempt to improve the GAN formulation per se, we are the first to make use of recent advances within super-resolution and replace [76] with the Wasserstein GAN of (WGAN) [2], as also improved in [37] (see also Eq. (6.3)).

We emphasize that our finding is that the improvement over [76] is only with respect to the stability and easiness of training and not with the quality of the super-resolved facial images: while training from scratch with the GAN loss of [76] is tricky and often leads to an unsatisfactory solution, by using a WGAN loss, we stabilized the training and allowed for the introduction of the GAN loss at earlier stages in the training process, thus reducing the overall training time. Finally, in terms of network architecture, we used the DCGAN [76] without batch normalization.



(a) Ours



(b) SR-ResNet

Fig. 6.3 A comparison between the proposed super-resolution architecture (left) and the one described in [61] (right). See also sub-section 6.2.1.

Adversarial loss

Following [2] and [37], the WGAN loss employed in our face super-resolution network is defined as:

$$\begin{aligned} l_{WGAN} = & \mathop{\mathbb{E}}_{\hat{I}\sim\mathbb{P}_g}[D(\hat{I})] - \mathop{\mathbb{E}}_{I\sim\mathbb{P}_r}[D(I^{HR})] \\ & + \lambda \mathop{\mathbb{E}}_{\hat{I}\sim\mathbb{P}_{\hat{I}}}[\,(\left\|\nabla_{\hat{I}} D(\hat{I})\right\|_2 - 1)^2\,], \end{aligned} \tag{6.3}$$

where $\mathbb{P}_r$ is the data distribution and $\mathbb{P}_g$ is the generator G distribution defined by $\hat{I} = G(I^{LR})$ (the input I_{LR} is randomly sampled from the set of low resolution facial images). $\mathbb{P}_{\hat{I}}$ is implicitly defines uniformly sampling along straight lines between pairs of samples from the data distribution $\mathbb{P}_r$ and the generator ones $\mathbb{P}_g$.

6.2.3 Face Alignment Network

The losses defined above (pixel, perceptual and adversarial) have been used in general purpose super-resolution and although alone do provide descent results for facial super-resolution, they also fail to incorporate information related to the structure of the human face into the super-resolution process. We have observed that when these losses are used alone pose or expression related details may be missing or facial parts maybe incorrectly located (see Fig. 6.4).

To alleviate this, we propose to enforce facial structural consistency between the low and the high resolution image via integrating a network for facial landmark localization through heatmap regression into the super-resolution process and optimizing an appropriate heatmap loss.

To this end, we propose to use the super-resolved image as input to a FAN and train it so that it produces the same output as that of another FAN applied on the original high resolution image. We note that FAN uses the concept of heatmap regression to localize the landmarks: rather than training a network to regress a 68×2 vector of x and y coordinates, each landmark is represented by an output channel containing a 2D Gaussian centered at the landmark's location, and then the network is trained to regress the 2D Gaussians, also known as heatmaps. As a number of works have shown (e.g. [8]), these heatmaps capture shape information (e.g. pose and expression), spatial context and structural part relationships. Enforcing the super-resolved and the corresponding HR image to yield the same heatmaps via minimization of their distance is a key element of our approach: not only are we able to localize the facial landmarks but actually we impose these two images to have similar facial structure. In terms of architecture, we simply used the pretrained FAN with 2 Hourglass modules introduced in Chapter 4.

Heatmap loss

Based on the above discussion, we propose to enforce structural consistency between the super-resolved and the corresponding HR facial image via a heatmap loss defined as:

$$l_{heatmap} = \frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{W} \sum_{j=1}^{H} \left\| \hat{p}_{i,j}^{n} - p_{i,j}^{n} \right\|^2, \tag{6.4}$$

where $\hat{p}_{i,j}^{n}$ and $p_{i,j}^{n}$ represent theconfidence maps at pixel location (i, j) for the nth landmark produced by running the FAN integrated into our super-resolution network on the super-resolved image $\hat{I}_{HR}$ and the heatmap obtained by running another FAN on the original image I_{HR}, respectively.

Another key feature of our heatmap loss is that its optimization does not require having access to ground truth landmark annotations just access to a pre-trained FAN. This allows us to train the entire super-resolution network in a weakly supervised manner which is necessary since for some of the datasets used for training (e.g. CelebA) ground truth landmark annotations are not available, anyway.

6.2.4 Overall training loss

The overall loss used for training Super-FAN is:

$$l^{SR} = \alpha l_{pixel} + \beta l_{feature} + \gamma l_{heatmap} + \zeta l_{WGAN}, \tag{6.5}$$

where $\alpha = 0.5, \beta = 0.5, \gamma = 0.1$ and $\zeta = 0.1$ are the corresponding weights.

6.2.5 Training

All images were cropped based on the bounding box such that the face height is 50 px. Input and output resolutions were 16×16 px and 64×64 px, respectively. To avoid overfitting, we performed random image flipping, scaling (between 0.85 and 1.15), rotation (between $-30°$ and $30°$), colour, brightness and contrast jittering. All models, except for the one trained with the GAN loss, were trained for 60 epochs, during which the learning rate was gradually decreased from 2.5e-4 to 1e-5. The model trained with the GAN loss was based on a previously trained model which was fine-tuned for 5 more epochs. The ratio between running the generator and the discriminator was kept to 1. Finally, for end-to-end training of the final model (i.e. Super-FAN), all networks (super-resolution, discriminator and FAN) were trained jointly for 5 epochs with a learning rate of 2.5e-4. All models, implemented in PyTorch [73], were trained using rmsprop [97].

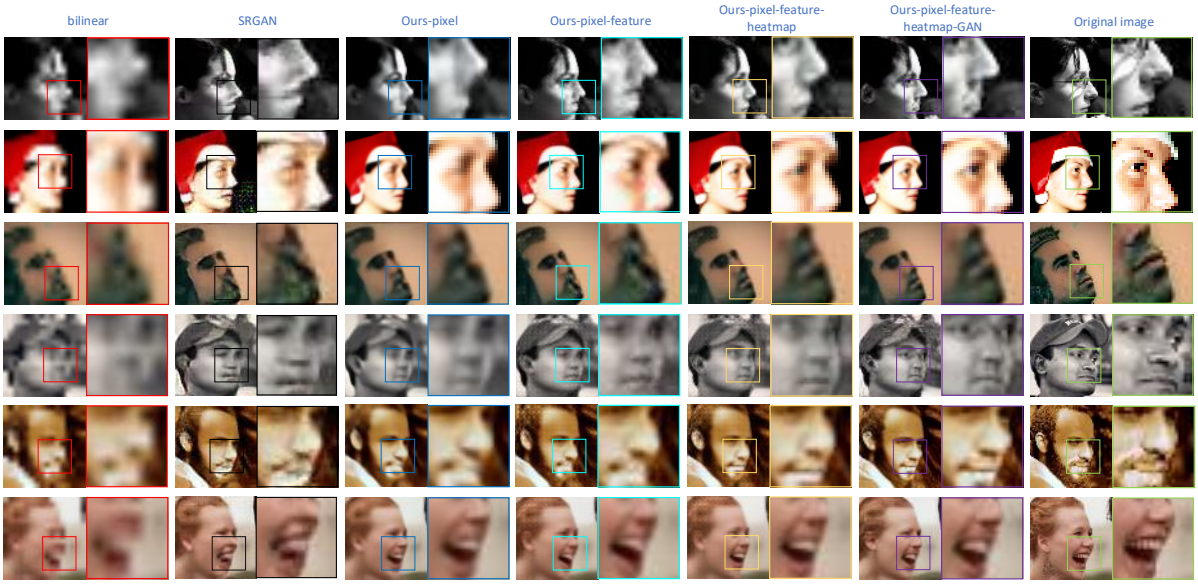


Fig. 6.4 Visual results on LS3D-W. Notice that: (a) The proposed Ours-pixel-feature already provides better results than those of SR-GAN [61]. (b) By additionally adding the newly proposed heatmap loss (Ours-pixel-feature-heatmap) the generated faces are better structured and look far more realistic. Ours-pixel-feature-heatmap-GAN is Super-FAN which improves upon Ours-pixel-feature-heatmap by adding the GAN loss and by end-to-end training. Best viewed in electronic format.

6.3 Experiments

In this section, we evaluate the performance of Super-FAN. The details of our experiments are as follows:

Training/Testing. Unless otherwise stated, all methods, including [61], were trained on the training sets of section 6.1. We report quantitative and qualitative results on the subset of LS3D-W balanced consisting of 3,000 images, with each pose range being *equally* represented. We

report qualitative results for more than 200 images from WiderFace.

Performance metrics. In sub-section 6.3.1, we report results using the standard super-resolution metrics, namely the SSIM [104] and PSNR defined as:

$$PSNR = 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10}(MSE), \tag{6.6}$$

where MAX_I is the maximum possible value of the image and the $MSE = \frac{1}{WH}\sum_{i=1}^{W}\sum_{j=1}^{H}[I(i,j) - \hat{I}(i,j)]^2$ is computed between the ground truth image I and the super-resolved one $\hat{I}$.

In the process we confirm [61] that both of them are a poor measure of the perceived image quality. In sub-section 6.3.2, we report results on facial landmark localization accuracy. To alleviate the issues with PSNR and SSIM, we also propose another indirect way to assess super-resolution quality based on facial landmarks: in particular, we trained a FAN on high resolution images and then used it to localize the landmarks on the super-resolved images produced by each method. As our test set (LS3D-W balanced) provides the ground truth landmarks, we can use landmark localization accuracy to assess the quality of the super-resolved images: the rationale is that, the better the quality of the super-resolved image, the higher the localization accuracy will be, as the FAN used saw only real high resolution images during training. The metric used to quantify performance is the Area Under the Curve (AUC) [11].

Variants compared. In section 5.1, we presented a number of networks and losses for super-resolution which are all evaluated herein. These methods are named as follows:

- Ours-pixel: this is the super-resolution network of sub-section 6.2.1 trained with the pixel loss of Eq. (6.1).
- Ours-pixel-feature: this is the super-resolution network of sub-section 6.2.1 trained with the pixel loss of Eq. (6.1) and the perceptual loss of Eq. (6.2).
- Ours-pixel-feature-heatmap: this is the super-resolution network of sub-section 6.2.1 trained with the pixel loss of Eq. (6.1), the perceptual loss of Eq. (6.2), and the newly proposed heatmap loss of Eq. (6.4).
- Ours-Super-FAN: this improves upon ours-pixel-feature-heatmap by additionally training with the GAN loss of Eq. (6.3) and by end-to-end training.

Comparison with the state-of-the-art. We report results for the method of [61], implemented with and without the GAN loss, called SR-GAN and SR-ResNet, respectively, and for the standard baseline based on bilinear interpolation. We also show visual results on WiderFace by running the code from [131][1].

[1]It is hard in general to compare with [131] because the provided code pre-processes the facial images very differently to our method.

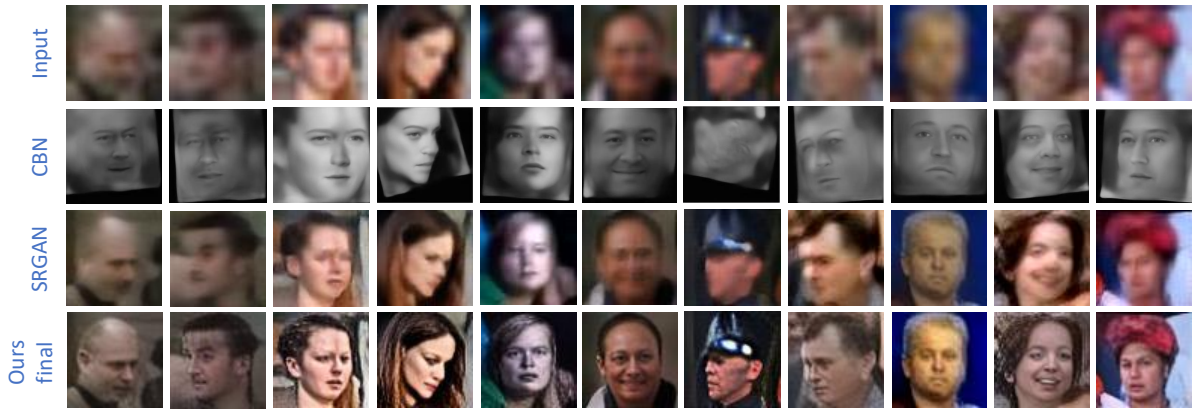


Fig. 6.5 Results produced by our system, SR-GAN [61] and CBN [131] on real-world low resolution faces from WiderFace.

6.3.1 Super-resolution results

Our quantitative results on LS3D-W across all facial poses are shown in Table 6.1. In terms of PSNR, the best results are achieved by Ours-pixel-feature-heatmap. In terms of SSIM, the best performing method seems to be Ours-pixel. From these numbers, it is hard to safely conclude which method is the best. Visually inspecting the super-resolved images though in Fig. 6.4 clearly shows that the sharper and more detailed facial images are by far produced by Ours-pixel-feature-heatmap and Ours-Super-FAN. Notably, Ours-pixel achieves top performance in terms of SSIM, yet the images generated by it are blurry and unrealistic (see Fig. 6.4), and are arguably less visually appealing than the ones produced by incorporating the other loss terms. We confirm the findings of [61] that these metrics can sometimes be misleading.

6.3.2 Facial landmark localization results

Herein, we present facial landmark localization results (on LS3D-W), also in light of our proposed way to evaluate super-resolution based on the accuracy of a pre-trained FAN on the super-resolved images (see **Performance metrics**). We report results for the following methods:

- FAN-bilinear: this method upsamples the LR image using bilinear interpolation and then runs FAN on it.
- Retrained FAN-bilinear: this is the same as FAN-bilinear. However, FAN was re-trained to work exclusively with bilinearly upsampled LR images.
- FAN-SR-ResNet: the LR image is super-resolved using SR-ResNet [61] and then FAN is run on it.
- FAN-SR-GAN: the LR image is super-resolved using using SR-GAN [61] and then FAN is run on it.

Method	PSNR			SSIM		
	30	60	90	30	60	90
bilinear upsample (baseline)	20.25	21.45	22.10	0.7248	0.7618	0.7829
SR-ResNet	21.21	22.23	22.83	0.7764	0.7962	0.8077
SR-GAN	20.01	20.94	21.48	0.7269	0.7465	0.7586
Ours-pixel	**21.55**	22.45	23.05	**0.8001**	**0.8127**	**0.8240**
Ours-pixel-feature	21.50	22.51	23.10	0.7950	0.7970	0.8205
Ours-pixel-feature-heatmap	**21.55**	**22.55**	**23.17**	0.7960	0.8105	0.8210
Ours-Super-FAN	20.85	21.67	22.24	0.7745	0.7921	0.8025

Table 6.1 PSNR- and SSIM-based super-resolution performance on LS3D-W balanced dataset across pose (higher is better). The results are not indicative of visual quality. See Fig. 6.4.

- FAN-Ours-pixel: the LR image is super-resolved using Ours-pixel and then FAN is run on it.
- FAN-Ours-pixel-feature: the LR image is super-resolved using Ours-pixel-feature and then FAN is run on it.
- FAN-Ours-pixel-feature-heatmap-GAN: the LR image is super-resolved using Ours-pixel-feature-heatmap-GAN and then FAN is run on it. The FAN is **not trained** with the rest of the super-resolution network i.e. the same FAN as above was used. This variant is included to highlight the importance of jointly training the face alignment and super-resolution networks as proposed in this work.
- Super-FAN: this is the same as above however, this time, FAN is **jointly trained** with the rest of the network.
- FAN-HR images: this method uses directly the original HR images as input to FAN. This method provides an upper bound in performance.

The results are summarized in Fig. 6.4 and Table 6.2. See Fig. 6.6 for examples showing the landmark localization accuracy. From the results, we conclude that:

1. Super-FAN is by far the best performing method being the only method attaining performance close to the upper performance bound provided by FAN-HR images.
2. Jointly training the face alignment and super-resolution networks is necessary to obtain high performance: Super-FAN largely outperforms FAN-Ours-pixel-feature-heatmap-GAN (second best method).
3. The performance drop of Super-FAN for large poses ($> 60^\circ$) is almost twice as much as that of FAN-HR images. This indicates that facial pose is still an issue in face super-resolution.

4. Even a FAN trained exclusively to work with bilinearly upsampled images (Retrained FAN-Bilinear), clearly an unrealistic scenario, produces moderate results, and far inferior to the ones produced by Super-FAN.

5. FAN-Ours-pixel-feature outperforms both FAN-SR-GAN and FAN-SR-ResNet. This shows that the proposed super-resolution network of section 6.2.1 (which does not use heatmap or WGAN losses) already outperforms the state-of-the-art.

6. From FAN-Ours-pixel to Super-FAN, each of the losses added improves performance which is in accordance to the produced visual results of Fig. 6.4. This validates our approach to evaluate super-resolution performance indirectly using facial landmark localization accuracy.

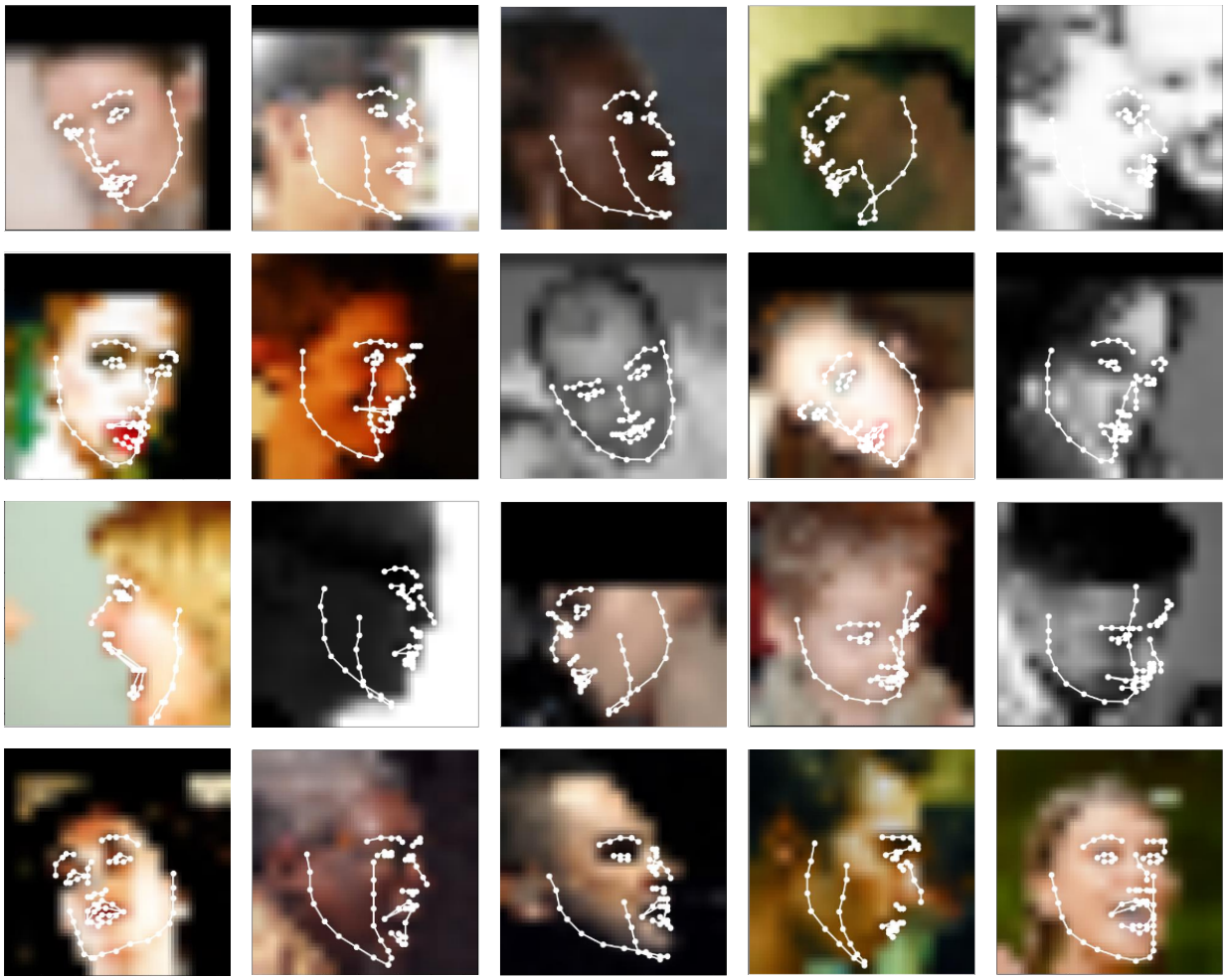


Fig. 6.6 Fitting examples produced by Super-FAN on a few images from LS3D-W. The predictions are plotted over the original low-resolution images. Notice that our method works well for faces found in challenging conditions such as large poses or extreme illumination conditions despite the poor image quality.

Method	[0-30]	[30-60]	[60-90]
FAN-bilinear	10.7%	6.9%	2.3%
FAN-SR-ResNet	48.9%	38.9%	21.4%
FAN-SR-GAN	47.1%	36.5%	19.6%
Retrained FAN-bilinear	55.9%	49.2%	37.8%
FAN-Ours-pixel	52.3%	45.3%	28.3%
FAN-Ours-pixel-feature	57.0%	50.2%	34.9%
FAN-Ours-pixel-feature-heatmap	61.0%	55.6%	42.3%
Super-FAN	**67.0%**	**63.0%**	**52.5%**
FAN-HR images	75.3%	72.7%	68.2%

Table 6.2 AUC across pose (calculated for a threshold of 10%; see [11]) on our LS3D-W balanced test set. The results, in this case, are indicative of visual quality. See Fig. 6.4.

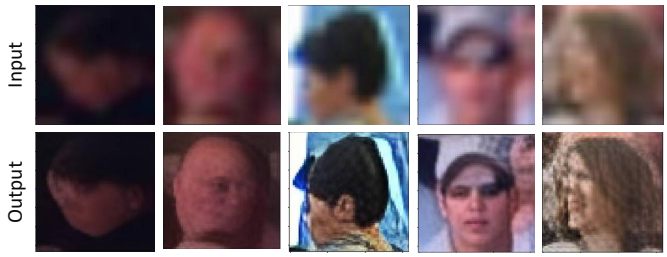


Fig. 6.7 Failure cases of our method on WiderFace. Typically, these include extreme facial poses, large occlusions and heavy blurring.

6.3.3 Comparison on real-world images

Most face super-resolution methods show results on synthetically generated LR images. While these results are valuable for assessing performance, a critical aspect of any system is its performance on real-world data captured in unconstrained conditions. To address this, in this section we provide visual results by running our system on more than 200 low resolution blurry images taken from the WiderFace and compare its performance with that of SR-GAN [61] and CBN [131].

Initially, we found that the performance of our method on real images, when trained on artificially downsampled images, was sub-optimal, with the super-resolved images often lacking sharp details. However, retraining Super-FAN by applying additionally random Gaussian blur (of kernel size between 3 and 7 px) to the input images, and simulating jpeg artefacts and colour distortion, seems to largely alleviate the problem. Results of our method, SR-GAN (also retrained in the same way as our method) and CBN can be seen in Figs. 6.1 and 6.5.

Our method provides the sharper and more detailed results performing well across all poses. SR-GAN fails to produce sharp results. CBN produces unrealistic results especially for the images that landmark localization was poor.

A few failure cases of our method are shown in Fig. 6.7; mainly cases of extreme poses, large occlusions and heavy blurring. With respect to the latter, although our augmentation strategy seems effective, it is certainly far from optimal. Enhancing it is left for interesting future work.

6.4 Ablation studies

This section describes a series of experiments, further analysing the importance of particular components on the overall performance. It also provides additional qualitative results.

On the pixel loss. In this section, we compare the effect of replacing the L2 loss of Eq. 6.1 with the L1 loss. While the L1 loss is known to be more robust in the presence of outliers, we found no improvement of using it over the L2 loss. The results are shown in Table 6.3.

On the heatmap loss. Similarly to the above experiment, we also replaced the L2 heatmap loss of Eq. 6.4 with the L1 loss. The results are shown in Table 6.5, showing descent improvement for large poses.

On the importance of the skip connection. Herein, we analysed the impact of the long-skip connections to the overall performance of the generator. The results, shown in Table 6.4, show no improvement.

On network speed. Besides accuracy, another important aspect of network performance is speed. Compared with SR-GAN [61], our generator is only 10% slower, being able to process 1,000 images in 4.6s (vs. 4.3s required by SR-GAN) on an NVIDIA Titan-X GPU.

6.4.1 Additional qualitative results

Fig. 6.9 shows the face size distribution Notice that our method copes well with pose variation and challenging illumination conditions. There were a few failure cases, but in most of these cases, it is impossible to tell whether the low-resolution image was actually a face.

Fig. 6.6 shows a few fitting results produced by Super-FAN on the LS3D-W Balanced dataset. The predictions were plotted on top of the low resolution input images. We observe that our method is capable of producing accurate results even for faces found in arbitrary poses exhibiting various facial expressions.

We also tested our system on images from the Surveillance Cameras Face dataset (SC-face) [35]. The dataset contains 4,160 images of 130 unique subjects taken with different cameras from different distances. Fig. 6.8 shows a few qualitative results from this dataset.

Fig. 6.8 Qualitative results on the SCface dataset [35].

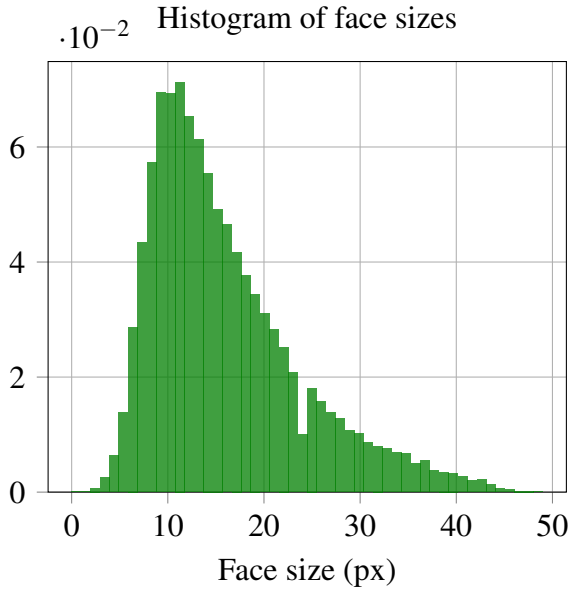


Fig. 6.9 Face size (defined as $\max(width, height)$) distribution of the selected subset of low resolution images from WiderFace.

Method	PSNR			SSIM		
	30	60	90	30	60	90
Ours-pixel (L2)	21.55	22.45	23.05	0.8001	0.8127	0.8240
Ours-pixel (L1)	21.47	22.40	23.00	0.7988	0.8120	0.8229

Table 6.3 PSNR and SSIM when training our generator with L2 and L1 pixel-losses.

Method	PSNR			SSIM		
	30	60	90	30	60	90
Ours-pixel (no-skip)	21.55	22.45	23.05	0.8001	0.8127	0.8240
Ours-pixel (with skip)	21.56	22.45	23.04	0.8021	0.8132	0.8241

Table 6.4 PSNR and SSIM for "no-skip" and "with skip" versions. The "no-skip" version indicates the absence of the long skip connection (the network depicted in Fig. 6.3a), while the "with skip" version adds two new long skip connections, similarly to [38].

Method	[0-30]	[30-60]	[60-90]
FAN-Ours-pixel-feature-heatmap (L2)	61.0%	55.6%	42.3%
FAN-Ours-pixel-feature-heatmap (L1)	61.1%	55.4%	42.0%

Table 6.5 AUC across pose (on our LS3D-W balanced test set) for L2 and L1 heatmap losses.

Chapter 7

Conclusions

The aim of this thesis was to address a series of challenges in the area of 2D and 3D face alignment, significantly advancing the state-of-the-art and proposing in the process novel deep learning-based architectures and methodologies. Mainly, we address: (a) the problem of fitting faces in very large poses, in the $-90^\circ - 90^\circ$ range (Chapter 3), (b) in both 2D and 3D space (Chapter 4), creating simultaneously (c) the largest "in-the-wild" large pose 3D face alignment dataset - LS3D-W (Chapter 4). Additionally, we study and address a new challenge: that of (d) fitting landmarks in very low resolution faces (Chapter 6). From a performance perspective, we propose (e) a novel residual block specially tailored for binarized neural networks that significantly improves the speed while maintaining a similar or competitive accuracy (Chapter 5). The results presented through the thesis set the new state-of-the-art on 2D & 3D face alignment as well as on face super-resolution.

This findings suggest that a carefully designed fully convolutional neural network architecture that follows a top-down approach can achieve near saturation results, comparable with a human annotator, for the case of 2D and 3D face alignment, even for the case of binarized neural networks. Futhermore, we have shown the importance of facial landmarks for guiding other tasks, in particular, we show that the keypoints help preserve the overall facial structure of super-resolved facial images.

Alongside the publications mentioned in Section 1.4, all of our code is available on Github at https://github.com/1adrianb under a BSD3-clause license (at the time of writing 8 unique repositories totalling thousands of downloads and github stars). The dataset introduced, LS3D-W was open sourced and can be downloaded from https://www.adrianbulat.com under the same permissive BSD3-clause license. Finally, a live demo of our 3D face alignment method is available at https://www.adrianbulat.com/face-alignment-demo.

In the beginning of this thesis, we emphasized that face alignment is one of the key steps of a plethora of computer vision applications. Currently, our 2D and 3D face alignment approach is widely used as part of various research tasks and topics such as: 3D face reconstruction, either as initialization or preprocessing step [47, 64], facial expression synthesis [92], face super-resolution [12, 115], face completion and editing [91, 109], hard examples mining [89], talking face image generation [125], emotion recognition [31, 63], facial attribute transfer [113],

selfie video stabilization [114], abnormal behavior detection [42], deception detection [70], gaze estimation [71], age estimation [27], facial palsy detection [41] or face frontalization [15].

7.1 Future work

The main focus of this thesis has been on addressing the key challenges for the problem of face alignment using novel, deep learning based methods. While significant progress was made (see Fig. 1.1 for examples of "before" and "after"), there is still room for further improvement. As such, the work conducted in this thesis, in addition to the above mentioned contributions, provides the basis for future work on several topics. At least four such topics were identified:

- Face tracking
- Multi-person face alignment
- Unconstrained low resolution face alignment
- Multi-task face analysis

The following sections discuss each of the proposed directions.

7.1.1 Face tracking

Although in Chapter 4 we conducted face tracking experiments on 300-VW, the work presented in this thesis focuses on static images, treating each frame from the video independently. A direct extension is to adapt the network to continuous data streams, exploiting the temporal relationship that naturally occurs in a video. This can both improve the accuracy and the stability of the method removing at the same time the requirement of running a face detector at each frame. A possible approach toward this will be to take in consideration the previous predictions when making the new one, exploit the optical flow or make joint predictions of multiple frames at once with a sliding window.

7.1.2 Multi-person face alignment

The current methods proposed throughout this thesis, in Chapters 3-6, can process a single face at one given moment (considering a batch size of 1). Therefore, if an image contains N faces, we either need to increase the batch size to N (the efficient way) or do N forward passes. However this can more elegantly be addressed inside the Fast-RCNN [32] framework formulation. Not only will this speed-up the fitting process itself, but it will also incorporate the face detection stage, previously done by a separate network.

7.1.3 Unconstrained low resolution face alignment

In Chapter 6, we made an initial attempt to address the problem of joint super-resolution and face alignment in real-world low resolution images (note that all previous work use artificially bilinearly downsampled images) by running an experiment on a subset of images from WiderFace. While we did show improvement on certain cases, there is still significant room for further performance gains.

7.1.4 Multi-task face analysis

Finally, all of the above methods, especially the work from Chapter 6 use different neural networks if the problem is different. However, one can use a single network with multiple heads, where each head will provide results for a specific task (e.g. super-resolution, 2D face alignment, etc). This was previously shown (for other tasks) to both improve the performance of individual tasks as well as reducing the computational requirements.

References

[1] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, 2014. 9

[2] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017. 65, 66, 67

[3] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic. Robust discriminative response map fitting with constrained local models. In *CVPR*, 2013. 10

[4] A. Asthana, S. Zafeiriou, G. Tzimiropoulos, S. Cheng, and M. Pantic. From pixels to response maps: Discriminative image filtering for face alignment in the wild. *IEEE TPAMI*, 37(6):1312–1320, 2015. 10

[5] V. Belagiannis, C. Rupprecht, G. Carneiro, and N. Navab. Robust optimization for deep regression. In *ICCV*, 2015. 9

[6] P. Belhumeur, D. Jacobs, D. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. In *CVPR*, 2011. 10, 12, 22

[7] A. Bulat and G. Tzimiropoulos. Convolutional aggregation of local evidence for large pose face alignment. In *British Machine Vision Conference (BMVC)*, 2016. ix, 18, 54, 55

[8] A. Bulat and G. Tzimiropoulos. Human pose estimation via convolutional part heatmap regression. In *European Conference on Computer Vision*, pages 717–732. Springer, 2016. 2, 9, 11, 43, 68

[9] A. Bulat and G. Tzimiropoulos. Two-stage convolutional part heatmap regression for the 1st 3d face alignment in the wild (3dfaw) challenge. In *European Conference on Computer Vision (ECCV)*, pages 616–624. Springer, 2016. 11, 39, 43

[10] A. Bulat and G. Tzimiropoulos. Binarized convolutional landmark localizers for human pose estimation and face alignment with limited resources. In *International Conference on Computer Vision (ICCV)*, 2017. viii, 30, 43

[11] A. Bulat and G. 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 (ICCV), 2017. xii, 16, 27, 64, 70, 74

[12] A. Bulat and G. Tzimiropoulos. Super-fan: Integrated facial landmark localization and super-resolution of real-world low resolution faces in arbitrary poses with gans. In *IEEE Conference on ComputerVision and Pattern Recognition (CVPR)*, 2018. 64, 78

[13] A. Bulat and Y. Tzimiropoulos. Hierarchical binary cnns for landmark localization with limited resources. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[14] X. P. Burgos-Artizzu, P. Perona, and P. Dollár. Robust face landmark estimation under occlusion. In *ICCV*, 2013. 24, 54, 55

[15] J. Cao, Y. Hu, H. Zhang, R. He, and Z. Sun. Learning a high fidelity pose invariant model for high-resolution face frontalization. In *Advances in Neural Information Processing Systems*, pages 2872–2882, 2018. 79

[16] Q. Cao, L. Lin, Y. Shi, X. Liang, and G. Li. Attention-aware face hallucination via deep reinforcement learning. In *CVPR*, 2017. 16

[17] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. In *CVPR*, 2012. 3, 10

[18] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. *IJVC*, 2014. 55

[19] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik. Human pose estimation with iterative error feedback. *arXiv preprint arXiv:1507.06550*, 2015. 9

[20] R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A matlab-like environment for machine learning. In *NIPS-W*, 2011. 31, 55

[21] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *TPAMI*, 23(6):681–685, 2001. 2

[22] M. Courbariaux, Y. Bengio, and J.-P. David. Training deep neural networks with low precision multiplications. *arXiv*, 2014. 14

[23] M. Courbariaux, Y. Bengio, and J.-P. David. Binaryconnect: Training deep neural networks with binary weights during propagations. In *NIPS*, 2015. 14, 51

[24] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio. Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. *arXiv*, 2016. 14, 47, 51, 52

[25] D. Cristinacce and T. Cootes. Feature detection and tracking with constrained local models. In *BMVC*, 2006. 28

[26] D. Cristinacce and T. Cootes. Automatic feature localisation with constrained local models. *Pattern Recognition*, 41(10):3054–3067, 2008. 10

[27] P. V. de Castro. Age estimation using deep learning on 3d facial features. 2018. 79

[28] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 21

[29] P. Dollár, P. Welinder, and P. Perona. Cascaded pose regression. In *CVPR*, 2010. 2, 3

[30] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *IEEE TPAMI*, 38(2):295–307, 2016. 15

[31] P. M. Ferreira, F. Marques, J. S. Cardoso, and A. Rebelo. Physiological inspired deep neural networks for emotion recognition. *IEEE Access*, 6:53930–53943, 2018. 78

[32] R. Girshick. Fast r-cnn. In *ICCV*, 2015. 9, 79

[33] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016. 2

[34] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 15, 66

[35] M. Grgic, K. Delac, and S. Grgic. Scface–surveillance cameras face database. *Multimedia tools and applications*, 51(3):863–879, 2011. x, 75, 76

[36] S. Gross and M. Wilber. Training and investigating residual nets. http://torch.ch/blog/2016/02/04/resnets.html. 65

[37] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017. 66, 67

[38] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. xiii, 10, 15, 30, 31, 39, 46, 55, 65, 66, 77

[39] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *ECCV*, 2016. ix, 11, 15, 43, 44, 46, 49, 52, 65

[40] J. L. Holi and J.-N. Hwang. Finite precision error analysis of neural network hardware implementations. *IEEE Transactions on Computers*, 42(3):281–290, 1993. 14

[41] G.-S. J. Hsu, W.-F. Huang, and J.-H. Kang. Hierarchical network for facial palsy detection. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 693–6936. IEEE, 2018. 79

[42] S. Hu, X. Jia, and Y. Fu. Research on abnormal behavior detection of online examination based on image information. In *2018 10th International Conference on Intelligent Human-*

Machine Systems and Cybernetics (IHMSC), volume 2, pages 88–91. IEEE, 2018. 79

[43] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, University of Massachusetts, Amherst, 2007. 11

[44] H. Huang, R. He, Z. Sun, and T. Tan. Wavelet-srnet: A wavelet-based cnn for multi-scale face super resolution. In *ICCV*, 2017. 16

[45] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. *arXiv*, 2016. 10

[46] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv*, 2015. 43, 65

[47] A. S. Jackson, A. Bulat, V. Argyriou, and G. Tzimiropoulos. Large pose 3d face reconstruction from a single image via direct volumetric cnn regression. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1031–1039. IEEE, 2017. 78

[48] V. Jain and E. G. Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. *UMass Amherst Technical Report*, 2010. 13

[49] L. A. Jeni, S. Tulyakov, L. Yin, N. Sebe, and J. F. Cohn. The first 3d face alignment in the wild (3dfaw) challenge. In *ECCV*, 2016. 11, 28

[50] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 22

[51] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016. 15, 65, 66

[52] S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *BMVC*, 2010. 9

[53] A. Jourabloo and X. Liu. Pose-invariant 3d face alignment. In *ICCV*, 2015. 10, 11, 13, 22, 24, 53, 54

[54] A. Jourabloo and X. Liu. Large-pose face alignment via cnn-based dense 3d model fitting. In *CVPR*, 2016. xi, xii, 10, 11, 22, 23, 24, 54

[55] V. Kazemi and S. Josephine. One millisecond face alignment with an ensemble of regression trees. In *CVPR*, 2014. 24

[56] J. Kim, J. Kwon Lee, and K. Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, 2016. 15

[57] M. Köstinger, P. Wohlhart, P. M. Roth, and H. Bischof. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *ICCV-W*, 2011. 11, 13, 22, 53, 54, 64

[58] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. 2009. 14

[59] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 365–372. IEEE, 2009. 16

[60] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang. Interactive facial feature localization. In *ECCV*, 2012. 10, 12, 16, 22

[61] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017. x, 15, 16, 65, 66, 67, 69, 70, 71, 74, 75

[62] D. D. Lin, S. S. Talathi, and V. S. Annapureddy. Fixed point quantization of deep convolutional networks. *arXiv*, 2015. 14

[63] C. Liu, T. Tang, K. Lv, and M. Wang. Multi-feature based emotion recognition for video clips. In *Proceedings of the 2018 on International Conference on Multimodal Interaction*, pages 630–634. ACM, 2018. 78

[64] F. Liu, R. Zhu, D. Zeng, Q. Zhao, and X. Liu. Disentangling features in 3d face shapes for joint face reconstruction and recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5216–5225, 2018. 78

[65] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. 16, 64

[66] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. xii, 9, 10, 19, 21, 29, 60, 61

[67] P. Merolla, R. Appuswamy, J. Arthur, S. K. Esser, and D. Modha. Deep neural networks are robust to weight binarization and other non-linear distortions. *arXiv*, 2016. 51

[68] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre. Xm2vtsdb: The extended m2vts database. In *2nd international conference on audio and video-based biometric person authentication*, volume 964, pages 965–966, 1999. 12

[69] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016. viii, 2, 10, 29, 30, 43, 53, 59

[70] M. Ngô, B. Mandira, S. F. Yılmaz, W. Heij, S. Karaoglu, H. Bouma, H. Dibeklioglu, and T. Gevers. Deception detection by 2d-to-3d face reconstruction from videos. *arXiv preprint arXiv:1812.10558*, 2018. 79

[71] C. Palmero, J. Selva, M. A. Bagheri, and S. Escalera. Recurrent cnn for 3d gaze estimation using appearance and shape cues. *arXiv preprint arXiv:1805.03064*, 2018. 79

[72] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. Jawahar. Cats and dogs. In *CVPR*, 2012. 25

[73] A. Paszke, S. Gross, and S. Chintal. Pytorch. http://github.com/pytorch/pytorch. 69

[74] T. Pfister, J. Charles, and A. Zisserman. Flowing convnets for human pose estimation in videos. In *ICCV*, 2015. 9, 19, 20

[75] T. Pfister, K. Simonyan, J. Charles, and A. Zisserman. Deep convolutional neural networks for efficient pose estimation in gesture videos. In *ACCV*. 2014. 2

[76] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 66

[77] R. Ranjan, V. M. Patel, and R. Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *arXiv preprint arXiv:1603.01249*, 2016. 53, 54

[78] R. Ranjan, S. Sankaranarayanan, C. D. Castillo, and R. Chellappa. An all-in-one convolutional neural network for face analysis. In *IEEE Face & Gesture*, 2017. xii, 53, 54

[79] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *ECCV*, 2016. 14, 45, 47, 51, 52, 53, 55

[80] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015. 57

[81] C. Sagonas, E. Antonakos, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: Database and results. *IVC*, 47:3–18, 2016. 11, 12, 28

[82] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *CVPR*, 2013. 5, 11, 12, 13, 28

[83] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. A semi-automatic methodology for facial landmark annotation. In *CVPR-W*, 2013. 10, 22

[84] M. S. Sajjadi, B. Schölkopf, and M. Hirsch. Enhancenet: Single image super-resolution through automated texture synthesis. *ICCV*, 2017. 15

[85] E. Sánchez-Lozano, B. Martinez, G. Tzimiropoulos, and M. Valstar. Cascaded continuous regression for real-time incremental face tracking. In *ECCV*, 2016. 32, 33

[86] J. Saragih, S. Lucey, and J. Cohn. Deformable model fitting by regularized landmark mean-shift. *IJCV*, 91(2):200–215, 2011. 10

[87] J. Shen, S. Zafeiriou, G. G. Chrysos, J. Kossaifi, G. Tzimiropoulos, and M. Pantic. The first facial landmark tracking in-the-wild challenge: Benchmark and results. In *ICCV-W*, 2015. 5, 11, 12, 28

[88] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 9, 19, 46, 66

[89] E. Smirnov, A. Melnikov, A. Oleinik, E. Ivanova, I. Kalinovskiy, and E. Lukyanets. Hard example mining with auxiliary embeddings. In *CVPR Workshop on Disguised Faces in the Wild*, volume 4, 2018. 78

[90] B. M. Smith and L. Zhang. Collaborative facial landmark localization for transferring annotations across datasets. In *ECCV*, 2014. 11

[91] L. Song, J. Cao, L. Song, Y. Hu, and R. He. Geometry-aware face completion and editing. *arXiv preprint arXiv:1809.02967*, 2018. 78

[92] L. Song, Z. Lu, R. He, Z. Sun, and T. Tan. Geometry guided adversarial facial expression synthesis. In *2018 ACM Multimedia Conference on Multimedia Conference*, pages 627–635. ACM, 2018. 78

[93] D. Soudry, I. Hubara, and R. Meir. Expectation backpropagation: Parameter-free training of multilayer neural networks with continuous or discrete weights. In *NIPS*, pages 963–971, 2014. 14

[94] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In *CVPR*, 2013. 10

[95] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, pages 4278–4284, 2017. 15, 46, 48

[96] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 15, 46

[97] T. Tieleman and G. Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2), 2012.

31, 55, 69

[98] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler. Efficient object localization using convolutional networks. In *CVPR*, 2015. 19

[99] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *NIPS*, 2014. 9, 19, 20, 51

[100] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *CVPR*, 2014. 9

[101] 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*, 2016. 10, 32, 33

[102] G. Tzimiropoulos. Project-out cascaded regression with an application to face alignment. In *CVPR*, 2015. 2, 3, 10

[103] G. Tzimiropoulos and M. Pantic. Optimization problems for fast aam fitting in-the-wild. In *ICCV*, 2013. 2

[104] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 70

[105] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. *arXiv preprint arXiv:1602.00134*, 2016. 9, 43

[106] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. *arXiv*, 2016. 15, 48, 56

[107] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 532–539, 2013. 3, 55

[108] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *CVPR*, 2013. 10, 24

[109] N. Xue, J. Deng, S. Cheng, Y. Panagakis, and S. Zafeiriou. Side information for face completion: a robust pca approach. *arXiv preprint arXiv:1801.07580*, 2018. 78

[110] C.-Y. Yang, S. Liu, and M.-H. Yang. Structured face hallucination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1099–1106, 2013. 16, 63

[111] H. Yang, X. Jia, C. C. Loy, and P. Robinson. An empirical study of recent face alignment methods. *arXiv preprint arXiv:1511.05049*, 2015. 10

[112] S. Yang, P. Luo, C. C. Loy, and X. Tang. Wider face: A face detection benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 16

[113] W. Yin, Z. Liu, and C. C. Loy. Instance-level facial attributes transfer with geometry-aware flow. *arXiv preprint arXiv:1811.12670*, 2018. 78

[114] J. Yu and R. Ramamoorthi. Selfie video stabilization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 551–566, 2018. 79

[115] X. Yu, B. Fernando, B. Ghanem, F. Porikli, and R. Hartley. Face super-resolution guided by facial component heatmaps. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 217–233, 2018. 78

[116] X. Yu, J. Huang, S. Zhang, W. Yan, and D. N. Metaxas. Pose-free facial landmark fitting via optimized part mixtures and cascaded deformable shape model. In *CVPR*, 2013. 24

[117] X. Yu and F. Porikli. Ultra-resolving face images by discriminative generative networks. In *ECCV*, 2016. 16, 65

[118] X. Yu and F. Porikli. Hallucinating very low-resolution unaligned and noisy face images by transformative discriminative autoencoders. In *CVPR*, 2017. 16, 65

[119] S. Zaferiou. The menpo facial landmark localisation challenge. In *CVPR-W*, 2017. 13, 28

[120] S. Zagoruyko and N. Komodakis. Wide residual networks. *arXiv*, 2016. 46

[121] M. D. Zeiler, G. W. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *2011 International Conference on Computer Vision*, pages 2018–2025. IEEE, 2011. 10

[122] J. Zhang, M. Kan, S. Shan, and X. Chen. Leveraging datasets with varying annotations for face alignment via deep regression network. In *ICCV*, 2015. 11

[123] N. Zhang, E. Shelhamer, Y. Gao, and T. Darrell. Fine-grained pose prediction, normalization, and recognition. *arXiv preprint arXiv:1511.07063*, 2015. 19, 52

[124] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Facial landmark detection by deep multi-task learning. In *ECCV*. 2014. 10, 33

[125] H. Zhou, Y. Liu, Z. Liu, P. Luo, and X. Wang. Talking face generation by adversarially disentangled audio-visual representation. *arXiv preprint arXiv:1807.07860*, 2018. 78

[126] S. Zhou, Y. Wu, Z. Ni, X. Zhou, H. Wen, and Y. Zou. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv*, 2016. 14

[127] S. Zhu, C. Li, C. Change Loy, and X. Tang. Face alignment by coarse-to-fine shape searching. In *CVPR*, 2015. 24, 33

[128] S. Zhu, C. Li, C. C. Loy, and X. Tang. Transferring landmark annotations for cross-dataset face alignment. *arXiv*, 2014. 11

[129] S. Zhu, C. Li, C. C. Loy, and X. Tang. Face alignment by coarse-to-fine shape searching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4998–5006, 2015. 10

[130] S. Zhu, C. Li, C. C. Loy, and X. Tang. Unconstrained face alignment via cascaded compositional learning. In *CVPR*, 2016. xi, 22, 23, 24

[131] S. Zhu, S. Liu, C. C. Loy, and X. Tang. Deep cascaded bi-network for face hallucination. In *ECCV*, 2016. x, 16, 63, 70, 71, 74

[132] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li. Face alignment across large poses: A 3d solution. *CVPR*, 2016. viii, ix, xii, 5, 10, 11, 13, 22, 28, 34, 35, 40, 53, 54, 55, 64

[133] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, 2012. 12, 28