Escuela Técnica Superior de Ingenieros Informáticos

Universidad Politécnica de Madrid

Multi-task Shape Preserving Face Alignment

Tesis Doctoral

Grado de Doctor en Inteligencia Artificial

AUTOR: Roberto Valle Fernández
DIRECTOR/ES: Luis Baumela Molina y
José Miguel Buenaposada Biencinto

Madrid, 2019

Multi-task shape preserving face alignment

Roberto Valle Fernández

Agradecimientos

Durante el desarrollo de la tesis doctoral me han acompañado numerosas personas e instituciones a las que me gustaría dedicar unas palabras de gratitud.

- En primer lugar, necesito agradecer encarecidamente a mis dos directores de tesis, José Miguel Buenaposada y Luis Baumela, la oportunidad brindada para trabajar a su lado. Para mí ha sido un auténtico privilegio el poder compartir todo este tiempo con vosotros. Siempre es un honor trabajar junto a dos personas a las que admiras tanto. José Miguel todo lo que he aprendido te lo debo a ti. Luis ojalá algún día yo pueda ser tan excelente profesor como tú eres. Colaborar contigo en las asignaturas de *Reconocimiento de formas* y *Deep leaning* ha sido un placer.

- Quiero destacar la colaboración de Antonio Valdés, profesor de la Universidad Complutense de Madrid. Las extraordinarias reuniones semanales que hemos mantenido durante todos estos años, así como su perspectiva matemática del problema, han sido parte fundamental de la tesis. Muchas gracias Antonio por permitirme ayudar, como tutor externo, a David González en el desarrollo de su TFG.

- Agradecer al Ministerio de Economía y Competitividad de España la financiación recibida de parte de los proyectos de investigación SPACES-UPM (TIN2013-47630-C2-2-R) y HEIMDAL-UPM (TIN2016-75982-C2-2-R) asociados al desarrollo de esta tesis. Además, quiero destacar la coordinación de ambos proyectos llevada a cabo por el departamento de electrónica de la Universidad de Alcalá.

- Gracias al EIT Digital Doctoral School por los cursos de emprendimiento recibidos, por fomentar el desarrollo industrial de la tesis y por financiar la estancia de tres meses realizada en la Universidad de Nottingham, bajo la supervisión de Georgios Tzimiropoulos a quien he tenido el privilegio de conocer. Gracias Yorgos por aceptar mi propuesta y hacerme formar parte del excelente equipo que tienes.

- Agradecer así al Centro de Supercomputación y Visualización de Madrid (CeSViMa) el permitirme hacer uso de sus recursos de cómputo para los experimentos de la tesis.

- No quiero olvidar mencionar a los que han sido mis compañeros durante estos años en el laboratorio PCR. Kendrick Cetina, Iago Suárez, Enrique Muñoz, Liangjin Zhao, Juan Bekios, Antonio Fernández, Ghesn Sfeir, Apurba Gorai, David Jurado, Nestor Audante y Chuiwen Ma. Además es motivo de orgullo el haber podido ayudar, como tutor externo, en la realización de su TFG o TFM a Alejandro Rabadán, Alejandro José Povo, Elvira Amador y Pedro Diego López. Mis mejores deseos para todos.

- En el plano personal, debo agradecer a mis amigos de toda la vida todos los grandes momentos que hemos pasado juntos durante el desarrollo de la tesis y que considero han sido importantes para poder encarar cada momento con mucha más alegría.

- Mención especial por supuesto para mi novia, Rocío Melo. Ella ha sido la dueña de mi corazón y mi mayor apoyo durante todos estos años. La persona con la que más tiempo paso y con la que quiero compartir el resto de mi vida. También agradecer a toda su familia el cariño que me muestran en todo momento.

- Por último, como primer miembro de la familia que realiza un doctorado, necesitaba dedicarle este trabajo y el título de Doctor en Inteligencia Artificial a toda mi familia, por creer en mis posibilidades siempre. En particular, destacar a mi padre, mi madre y mi hermano. No hace falta explicar por qué. Sin ellos nada hubiera sido igual.

Gracias a todos de corazón.

Resumen

Esta tesis aborda el problema de la estimación robusta y precisa de la configuración de rostros humanos en imágenes. Esto implica la localización en la imagen de un conjunto de puntos de referencia fiduciales que representan la combinación de los componentes rígidos y no rígidos de la configuración de un rostro. Este es un problema relevante y abierto en el campo de la visión por computador, cuando analizamos rostros sin restricciones en su captura; es decir, incluyendo poses extremas, expresiones faciales exageradas, iluminación arbitraria, oclusiones parciales, baja resolución, etc. Para este fin, empleamos un enfoque basado en una cascada de regresores que refinan iterativamente sus estimaciones hasta llegar a una solución final.

Abordamos primeramente el problema de estimar la orientación de la cabeza, porque es el más simple y, además, porque, una vez que se conoce el componente rígido del rostro, la deformación de la cara es más fácil de calcular. Es muy difícil establecer el estado del arte en este área, pues no hay una metodología comúnmente acordada para la evaluación del rendimiento. En la tesis presentamos un conjunto de datos, unos algoritmos y unos resultados de base para realizar dicha evaluación.

Para estimar la deformación no rígida de la cara empleamos un esquema de regresores en cascada. Primero consideramos la estrategia tradicional, basada en el entrenamiento de una secuencia de árboles de regresión mediante un algoritmo de "Gradient Boosting" (ERT). Aquí presentamos una nueva arquitectura de refinado progresivo que aborda la explosión combinatoria que se produce al considerar las posibles deformaciones de todas las partes de la cara. También evaluamos el enfoque más habitual en este momento, que consiste en la construcción de una cascada de redes de neuronas convolucionales (CNNs). Introducimos una solución basada en combinar dos CNNs con una nueva capa final para estimar las coordenadas de los puntos de referencia. Finalmente, presentamos una nueva solución híbrida basada en la combinación de una CNN y un ERT de refinado progresivo. Nuestro enfoque funciona en presencia de puntos de referencia ocluidos o no etiquetados en el conjunto de entrenamiento. Esto nos ha permitido realizar experimentos cruzando bases de datos, que revelan la existencia de un sesgo significativo en los conjuntos de datos de entrenamiento. Lo cual, sin duda, limita la capacidad de generalización de los regresores entrenados con dichas bases de datos. Hasta donde sabemos, ésta es la primera vez que se ha planteado este problema en el contexto de la *alineación del rostro*.

En nuestra propuesta final presentamos un enfoque unificado para inferir la orientación de la cabeza, los puntos de referencia del rostro y sus visibilidades. Dicha propuesta está basada en la combinación de una red neuronal multi-tarea (MNN) que simultáneamente estima la orientación de la cabeza, la posición de los puntos de referencia y sus visibilidades, junto con un ERT de refinado progresivo. La arquitectura de la MNN, cómo la entrenamos, y cómo combinamos sus predicciones con el ERT son novedosos.

En los experimentos realizados evaluamos el rendimiento de nuestras propuestas y las comparamos con los mejores algoritmos que existen en la literatura, utilizando las bases de datos más relevantes.

Abstract

This thesis deals with the problem of accurately and robustly estimating the pose of human faces in images. This involves the location in the image of a set of facial fiducial points or landmarks that represent the combination of the rigid and non-rigid components of face pose. This is a relevant and open problem in computer vision when we capture faces under "in-the-wild" conditions, *i.e.*, those including extreme rotations, exaggerated facial expressions, arbitrary illumination, partial occlusions, blurriness, and so forth. We adopt a cascade approach in which a sequence of regressors iteratively refines their estimations to reach a final solution.

We first consider the problem of estimating the head orientation because it is simple. Also, because, once the rigid component of a face is known, the deformation is easier to compute. It is very difficult to determine the state-of-the-art in this area because there is no agreed upon methodology in the literature. In the thesis we introduce a head pose benchmark together with a set of baseline results supported by one traditional algorithm based on ensemble learning and several recent CNN architectures.

To estimate the non-rigid deformation of the face we adopt a cascade scheme. First we consider the traditional approach based on Gradient Boosting to learn a sequence of tree regressors (ERT). Here, we introduce a novel coarse-to-fine architecture that addresses the combinatorial explosion of combinations of face part deformations. We also evaluate the present mainstream approach consisting on cascading a set of Convolutional Neural Networks (CNNs). We introduce a solution based on a pair of CNNs with a new final layer to estimate the landmark coordinates. Finally, we introduce a novel hybrid solution based the combination of a CNN and a coarse-to-fine ERT. Our approach may also be trained in presence of missing or occluded landmarks in the training set. This has enabled us to perform cross-dataset experiments that reveal the existence of significant data set bias that may limit the generalization capabilities of regressors trained on present data sets. To the best of our knowledge, this is the first time such a problem has been raised in the context of *face alignment*.

In our final proposal we present a unified approach to infer head pose, facial landmark location and visibility estimation. It is based on the combination of a Multi-task Neural Network (MNN) that simultaneously estimates head pose, landmarks position and their visibilities, together with a coarse-to-fine ERT. The architecture of the MNN, the way we train it, and the way we combine its predictions with the ERT are all new.

In the experiments we evaluate the performance of our proposals and compare them with the top performing algorithms in the literature using the most relevant "in-the-wild" benchmarks.

Contents

Chapter 1

Introduction

The perception of human faces is an essential and effortless neurological mechanism for us to visually distinguish other people. From birth, newborns possess primitive facial processing capabilities and show particular interest in real faces. Thereafter, adults readily recognize faces and infer at glance different information such as age, gender, race, mood, etc. The analysis of this large set of facial traits has been one of the most active research areas in the field of computer vision during the past decades, because of its applications in human-computer interaction, video surveillance, social interactions analysis and biometric security, to name a few.

Henceforth, we denote as *facial analysis problems* the following tasks: facial expression recognition: estimates mood and emotions, *i.e.*, happiness, anger, disgust, fear, sadness or surprise [137, 46, 64, 115]; the detection of facial action units: deals with the movements of some muscles corresponding to an emotion [142, 121, 94]; facial attributes classification: extracts some physical biometric traits associated with the face, including demographic data (*e.g.*, gender, age and race), hair style, presence of beard, glasses, make-up, and so forth [7, 69, 141, 44, 43]; identity recognition: describes the ability to identify a person and prevents face spoofing [106, 74, 99]; face reenactment: modifies the appearance of a person by transferring the expression of another subject [118, 82]; face verification: determines whether two face images belong to the same individual [62, 103]; face reconstruction: generates a 3D mesh model from a 2D image [34, 9]; face hallucination: refers to the task of recovering high resolution face images from corresponding low resolution inputs [15].

Over time, an extensive literature such as the one presented above, has raised the importance of *face alignment* as a pre-processing step to improve the performance of most facial analysis problems that require knowledge of the location of each face part of interest, *e.g.*, earlobes, eyes, nose, mouth, chin, and so forth. For example, we require an accurate mouth detection to identify whether a person is smiling or not, and we also demand a good eye location for determining the presence of glasses. Face alignment has been an active topic in the computer vision field for over 20 years [52]. It represents the problem of matching a 3D face model with a 2D face image, such that the geometric structure defining the 3D model, accurately describes the shape of the face projected onto the image. In general, object alignment enables the establishment of correspondences between the model and the image plane, which helps us obtain a normalized rotation, translation, and scale representation of the object. As a result, we can apply this transformation to each input image to align the face with a single reference coordinate system (see Fig. 1.1), which is also referred to as image registration. However, the errors that occur in the detection of facial parts, as well as the variability of faces, often result in unstable registration, *e.g.*, the last face in Fig. 1.1. These misalignments may have an impact on the quality of facial analysis results in unconstrained scenarios, because these wrong normalizations may induce errors in the extracted features. In our proposal, we focus on the face alignment problem, ignoring any image registration step to normalize the faces.

(a) Original cropped face images

(b) Registered face images using Zhu *et al.* [146] face alignment approach

Figure 1.1: Comparison between the original and registered face images from the Adience data set [32] using a standard face alignment procedure.

A robust and accurate face alignment algorithm should be able to handle both rigid and non-rigid face deformation in uncontrolled conditions (so-called "in-the-wild"). However, although many efforts have been devoted to the face alignment problem during the past decades [53], it still remains a very challenging task in unrestricted scenarios due to various conditions such as arbitrary illumination changes, exaggerated facial expressions, presence of occlusions, extreme poses, ethnic diversity, etc. The face alignment task becomes even harder when many of these challenges are present simultaneously. In Fig. 2.6a, we display some representative examples of face images acquired under laboratory conditions, where some volunteers were asked to pose in a controlled scenario. In comparison with Fig. 2.6b, it is visually noticeable that any face analysis task would be harder using those in-the-wild face images collected from the Internet, and exposed to multiple nuisance factors. At this point, we confirm that recent development of challenging benchmarks have contributed to realize that some approaches have become obsolete because they offer poor accuracy in realistic scenarios, thus we need the progress of current face alignment data sets to inspire novel ideas and reduce the gap between research and real world requirements. The most promising moving direction of this field might be the 3D face alignment benchmarks [133].

Once discussed the difficulties arising in accurately aligning faces in-the-wild, we first address this task by learning the rigid and non-rigid face deformation as two independent problems. On the one hand, we roughly compute the rigid pose of a face as the estimation of the head position and 3D orientation with respect to the camera coordinate system. The head pose is usually modelled as a rigid transformation, described by a translation and a rotation. It is traditionally used as a preliminary step to reduce the variability in other facial analysis problems due to the appearance changes produced by extreme poses. On the other hand, we infer the non-rigid face deformation through the detection of multiple fiducial points of interest, denoted as "landmarks", which define the projection of the face shape onto the image plane. These key points delimit the deformation produced by facial expressions in most visually noticeable face parts, *e.g.*, earlobes, eyes, nose, mouth, chin, etc. However, the accurate location of these landmarks from scratch is quite challenging because it implies the estimation of both rigid and non-rigid deformation together. In Fig. 1.2 we show an example of head pose estimation and facial landmark location under a realistic scenario.

In this thesis we depart from the known location of each face in the image, which is also referred to as "bounding box" face detection. Most in-the-wild data sets include this bounding box ground truth in their annotations. It is usually acquired using face detectors robust to variations in head pose, illumination, expression, scale, skin colour, occlusion, make-up, and so forth. In Fig. 1.2 we draw a cyan rectangle around each detected face using the SSD detector [67]. Here, we also notice that there is a person wearing sunglasses whose bounding box does not fit properly around the chin. In fact, the face detection task is still far from being solved in challenging scenarios, since the facial boundaries are sometimes not discernible. Although a discussion of how to perform the detection of these faces under in-the-wild conditions falls outside the scope of this thesis, we provide in the next section an overview to *face detection*, given its relevance.

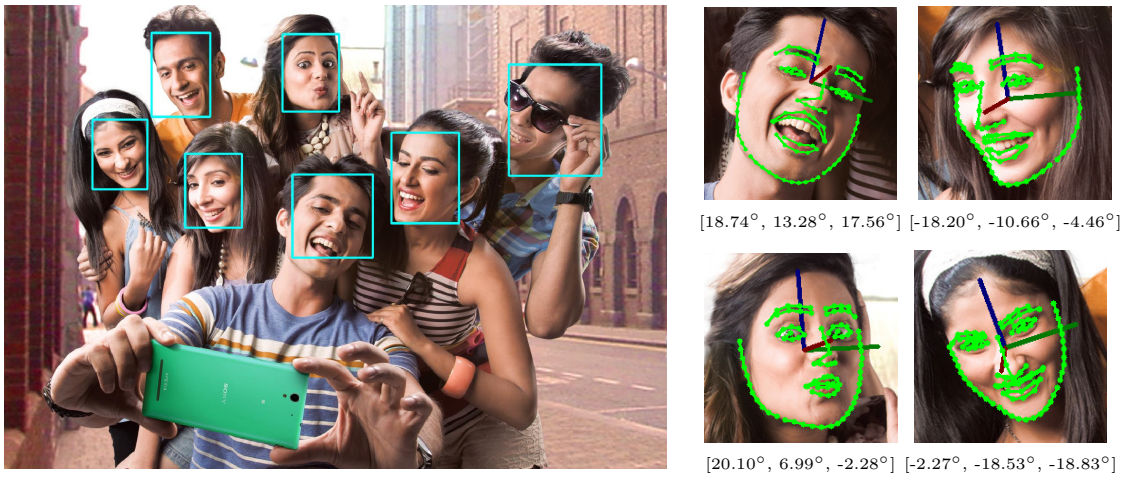


Figure 1.2: Head pose and facial landmark estimation example under a realistic situation. The axis orientation represents the three rotation angles. Green coloured coordinates show landmarks predictions. We locate cyan bounding boxes using the SSD detector [67].

1.1 Face alignment requirements

In the past decades, face detection has attracted a lot of attention within the computer vision field [135] due to the multitude of applications that result from it. Face detection is the process related to the acquisition of human faces from digital images, and it is a critical step for any application that requires face inspection because its accuracy greatly affects the performance of the face analysis that follows. As a result, it provides a bounding box for each human face, being these bounding boxes the coordinates of the rectangle that fully encloses each face in the image. Nowadays, face detectors have achieved impressive results on large and medium-sized faces, however, the performance on small faces is still far from satisfactory.

The traditional methodology to perform facial analysis depends on two preliminary steps: face detection and face alignment [52]. At this moment, there is also a considerable amount of literature that learn simultaneously face detection along with some of these subsequent analysis tasks, *e.g.*, face alignment [146, 35] or attributes recognition [87, 86].

In this thesis we assume that faces have already been detected. We always perform experiments using the bounding boxes annotated in each benchmark. The adoption of a face detection algorithm is a first step required to apply our contribution in a practical

framework, similar to the majority of commercial software packages, *e.g.*, Google Cloud Vision AI®, Microsoft Azure®, Amazon Rekognition®, IBM Watson Visual Recognition API®, Face++®, Affectiva®, Facesoft®, Kairos®, and so forth.

To this end, we recommend some classic sliding-window implementations trained with Boosting algorithms to achieve real-time performance around semifrontal faces, such as Viola *et al.* [114] or Dalal *et al.* [25] that benefit from handcrafted Haar-like or HOG features respectively. The detector proposed by Mathias *et al.* [75] (HeadHunter) is computationally expensive, but reports excellent results with human faces under in-the-wild conditions, using rigid templates and robust Integral Channel features [29]. On the other hand, deep learning methods provide the state-of-the-art based on the single shot strategy, *e.g.*, YOLO [88], SSD [67] or Retina-Net [65], although their computational complexity requires a dedicated GPU to achieve real-time performance[1].

1.2 Motivation

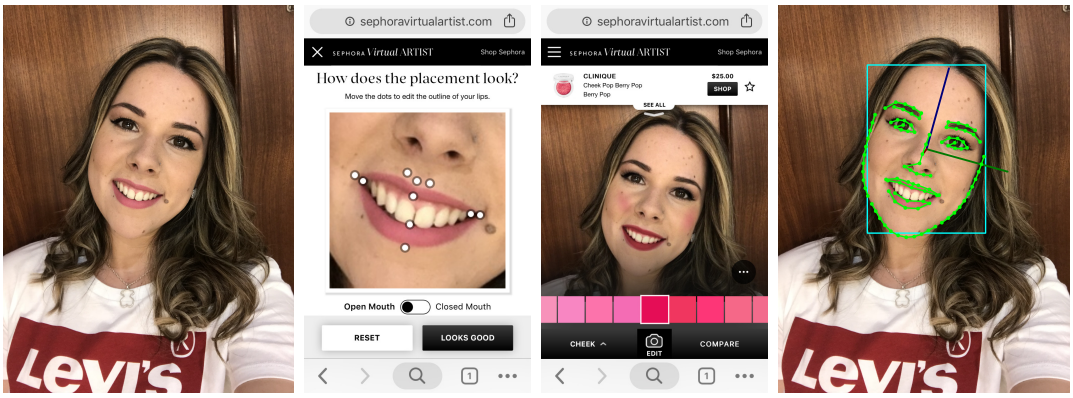
This thesis is the first step towards the construction of a full system that semantically describes people and their activities, by detecting different anomalies in their behaviour. In this case, a fundamental component is the extraction of information from the human face to estimate visual semantic attributes (*e.g.*, young woman wearing glasses is looking right). This research is part of two projects, SPACES-UPM (TIN2013-47630-C2-2-R) and HEIMDAL-UPM (TIN2016-75982-C2-2-R), funded by the Spanish Ministry of Economy and Competitiveness.

As stated in the introduction, head pose estimation and facial landmark detection are two crucial steps to successfully perform facial analysis. In fact, both tasks have gained major interest due to their potential applications through augmented reality technology. For example, companies such as Michael Kors® or Sephora® promote certain accessories or cosmetics using mobile apps that allow the customers to model and try-on sunglasses, lipsticks or eye shadows, in search for one that suits them (see Fig. 1.3a). Another face alignment application of interest is in popular social networks filters that animate a face in real-time with different looks, *e.g.*, Snapchat®, Instagram®, MSQRD® or FaceApp® (see Fig. 1.3b). However, these apps typically require locating manually the landmarks around each facial part being modelled. They generally lack precision under challenging situations such as near-profile face orientation, faces partially occluded or multiple faces located in the scene, to name a few.

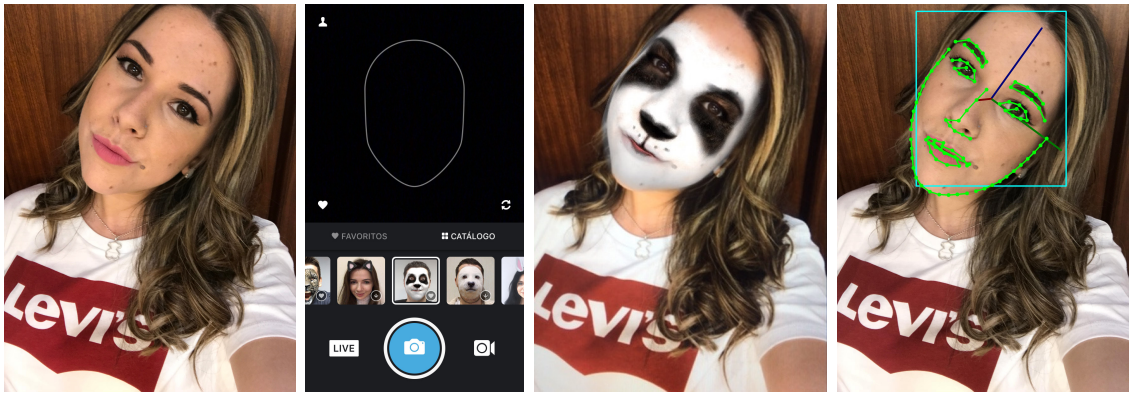
A careful analysis of commercial software based on current face alignment algorithms, reveals the existence of a knowledge gap in how to accurately address these in-the-wild challenges. The major difficulties in the face alignment field are the estimation of both rigid and non-rigid face deformation tasks, together with the development of a model with enough learning capacity to fit facial expressions, while preserving face shape under severe occlusions. Consequently, face alignment must balance the production of consistent face shape predictions robust to occluded landmarks, and the ability to be successful under extreme non-rigid face deformations.

Most literature only focuses on one of these problems, and as a result, they prioritize either an algorithm better suited to handle occlusions (*i.e.*, keeping a valid face shape) [139, 138, 54], or an algorithm that adapts to exaggerated facial expressions (*i.e.*, enabling a high degree of deformation) [42, 104, 28]. Our main motivation is the development of

[1]We suggest OpenCV library (https://github.com/opencv/opencv/tree/master/samples/dnn).



(a) Sephora® simulates virtual make-up using manually annotated landmarks



(b) MSQRD® overlays live filters over semifrontal faces

Figure 1.3: Realistic mobile phone applications combining face alignment and augmented reality for commercial advertising and social media. On the right we show our main result on simultaneous head pose and landmark estimation (see Chapter 4).

an efficient algorithm that works in realistic conditions, reaching a compromise between both problems, *i.e.*, partial occlusion robustness vs capacity to model facial expressions. Hereafter, our research is based on the following hypotheses:

- Most face alignment literature implicitly incorporates the rigid estimation task in the localization of facial landmarks. Our first assumption is that we would outperform head pose estimation by optimizing this problem using a specific regressor.

- Face alignment algorithms based on deep learning techniques are very robust in the majority of in-the-wild conditions and achieve impressive results across extreme head poses and facial expressions, however they struggle in presence of severe occlusions, and ambiguous facial configurations, such as for example when more than one nearby face appears in the image. This is due to the lack of a global face shape constraint. For this reason, we assume that their predictions would be further improved by implementing a module to preserve the face shape.

- Face alignment approaches that employ traditional machine learning techniques [48] (*e.g.*, Random Forest, Boosting, etc.) are efficient and achieve accurate results that let us implicitly impose a prior face shape model, when they are properly initialized. However, the limited representation of their handcrafted features, and the lack of an initialization close to the ground truth, restricts their performance, particularly

in uncontrolled settings. We hypothesize that these methods would improve if they use the discriminative features generated by deep networks, and they are properly initialized, but it remains unclear how to combine both strategies.

- Head pose estimation and facial landmark detection tasks are correlated. Thus, we would boost the performance of both if they were jointly modelled from a multi-task perspective.

- The performance of facial landmark detection would improve by taking into account the estimated visibility of each landmark.

1.3 Objectives

In this thesis our main goal is the development of an efficient face alignment algorithm able to outperform published state-of-the-art results. Relevant work that analyzes how to deal with face alignment [53, 122] focuses primarily on the detection of facial landmarks, which implicitly solves rigid and non-rigid estimation tasks together. Consequently, these approaches require a sequence of two or more models that iteratively refines the shape of interest (*i.e.*, first stages minimize the rigid alignment error, whereas last stages handle non-rigid deformation), at the expense of an increase in the computational time demanded to process such a cascade of regressors. In fact, the computational time required is another issue that we would like to alleviate in our proposal.

First, we pay attention to how to infer the rigid face deformation. We hypothesize that the estimation of the landmark location will improve notably if we train a face alignment model specialized only in the regression of the non-rigid face deformation. Since the rigid transformation of interest represents the 3D orientation and translation of the head with respect to the camera, our initial objective is to outperform the most relevant head pose estimation methods.

Simultaneously, we study the performance of recent work based on cascades of regressors that progressively estimate the location of these landmarks of interest from scratch. It is worth mentioning that the number of regressors depends on, whether they employ a good initialization (*i.e.*, fewer regressors when the rigid deformation has been solved), and whether they compute a discriminative feature representation (*i.e.*, robust features results in less regressors). Moreover, we notice that state-of-the-art results lack of a global head model to enforce face shape consistency in their predictions, which is an important issue for locating non-visible landmarks. In this thesis, our priority indeed is to preserve a valid face shape in our solutions.

Finally, our last objective is to combine all these tasks of interest (*i.e.*, head pose estimation, facial landmark location and their visibility estimation) in the same regressor using a multi-task strategy, where we infer rigid and non-rigid face deformation together, but optimizing each problem separately. We also require that all tasks share common features to produce better generalization.

To this end, we initially explore different traditional machine learning techniques based on ensemble learning [48] to perform head pose estimation and landmark detection. These algorithms became very popular before the irruption of deep learning in the face alignment field [100]. However, during the development of the thesis we reach the conclusion that, to achieve top performance, we require the use of deep networks, which have established the state-of-the-art in the face alignment task since 2016 [141].

1.4 Contributions

The research presented in this thesis resulted in different publications [111, 3, 112, 70, 110, 113]. The following points summarize our main contributions to the face alignment field using in-the-wild face images:

- In Chapter 2 we present a real-time classifier based on a Random Forest scheme [111] to infer discrete face orientation. We determine the head pose angles by combining handcrafted local features extracted randomly around the face. We also analyze the performance of Convolutional Neural Networks (CNNs) [3] to estimate the head pose angles from a holistic perspective. Here, we introduce a new benchmark to make fair evaluations among the state-of-the-art and compare the performance of both local and holistic methods. These two approaches produced the following two publications:

  R. Valle, J. M. Buenaposada, A. Valdés, and L. Baumela. Head-pose estimation in-the-wild using a random forest. In *Proc. Articulated Motion and Deformable Objects*, pages 24-33, 2016.

  E. Amador, R. Valle, J. M. Buenaposada, and L. Baumela. Benchmarking head pose estimation in-the-wild. In *Proc. Iberoamerican Congress on Pattern Recognition*, pages 45-52, 2017.

- In Chapter 3 we present a two-stage regressor formed by a sequence of deep networks (CNN+CNN) to perform the detection of different facial landmarks. This proposal produces a set of heatmaps, one heatmap per landmark, where the mode of each one determines the landmark positions [70]. In this work we focus on the lack of a face shape constraint. To this end, we train these networks using synthetic random occlusions. We also delete heatmaps between both CNNs, to learn how to detect these landmarks under occlusion using information of their neighbours. This work received the "best student paper award" at the 23rd Iberoamerican Congress on Pattern Recognition (CIARP 2018).

  Additionally, we extend our previous work, by incorporating a novel regression layer that automatically computes the landmark coordinates of interest from the feature maps produced by the last CNN [110]. Both approaches produced the following two publications:

  P. D. López, R. Valle, and L. Baumela. Facial landmarks detection using a cascade of recombinator networks. In *Proc. Iberoamerican Congress on Pattern Recognition*, pages 575-583, 2018.

  R. Valle, J. M. Buenaposada, and L. Baumela. Cascade of encoder-decoder CNNs with learned coordinates. *Pattern Recognition Letters*, (in press), 2019.

- In Chapter 3 we also present a two-stage hybrid scheme formed by a single CNN, which provides robust feature maps, and an ensemble of regression trees (ERT) that implicitly preserves the face shape when it is properly initialized [112]. As far as we know, this is the first algorithm that combines successfully both traditional machine learning and deep leaning methods in a single framework (CNN+ERT).

At this point, we introduce a coarse-to-fine ERT scheme that gradually refines the location of landmarks, avoiding the combinatorial explosion of parts deformation.

In addition, we extend the previous hybrid framework, by including a new initialization procedure robust under partial occlusions [113]. We also perform cross-dataset experiments that reveal the existence of significant data set bias that may limit the generalization capabilities of regressors trained on present data sets. To the best of our knowledge, this is the first time such a problem has been raised in the field. These algorithms result in the following two publications:

R. Valle, J. M. Buenaposada, A. Valdés, and L. Baumela. A deeply-initialized coarse-to-fine ensemble of regression trees for face alignment. In *Proc. European Conference on Computer Vision*, pages 609-624, 2018.

R. Valle, J. M. Buenaposada, A. Valdés, and L. Baumela. Face alignment using a 3D deeply-initialized ensemble of regression trees. *Computer Vision and Image Understanding*, 189, 2019.

- In Chapter 4 we present an original multi-task CNN architecture to simultaneously estimate head pose, localize landmarks, and predict their visibilities, which shares common features among tasks at earlier layers. We follow again a two-stage strategy (CNN+ERT), where a coarse-to-fine ERT that implicitly preserves the face shape, incorporates the predicted visibilities to ignore the refinement obtained from those regressors whose features are extracted around occluded or self-occluded landmarks.

Chapter 2

Head pose estimation

The human capacity to approximately estimate the head pose is a standard ability that we perform effortlessly. People typically use the orientation of the head to convey rich information during social interactions. For example, people nod their heads to indicate that they understand what is being said, or shift the head towards a specific direction when there is an object of interest.

In the context of computer vision, head pose estimation is the process of inferring the orientation of the head in the camera reference system [80]. By estimating the head pose, we mean predicting the relative orientation between the viewer and the target head. It is usually parametrized by the yaw, pitch and roll angles of the head (see Fig. 2.1).

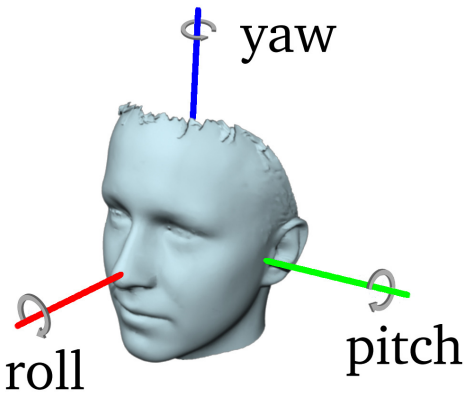


Figure 2.1: Yaw, pitch and roll rotation angles describe the pose of the head.

Head pose estimation is a relevant pre-processing step for many *facial analysis tasks*, since the performance of these problems is affected by the facial appearance variability, and most of it depends on the face orientation (*i.e.*, head pose texture variability may be even greater than a change in the person's identity). It has been used in multiple facial analysis problems, such as facial landmark detection [26, 126, 54, 61], identity recognition [106, 74], attributes estimation (*e.g.*, age, gender or race) [7], expression recognition [137], action units detection [142], attention detection [19] or identifying social interactions [73]. Traditionally a extreme head pose has been one of the major causes of failure among pose estimation methods.

As stated in Section 1.4, in this thesis we consider head pose estimation as a key step to perform *face alignment* in-the-wild. Our main goal is to accurately infer it, as a preliminary stage to alleviate the computation of the non-rigid face deformation. However, head pose estimation is attracting widespread interest in computer vision by itself, because it is also intrinsically linked with visual gaze estimation [45, 109] used in human-computer interaction, and driver monitoring [12] since the perception of visual attention away from road is a critical indicator of accident risk.

This chapter is structured as follows. In Section 2.1 we categorize existing literature and review the most prominent methods. In Section 2.2 we present a real-time algorithm to estimate the head pose using a Random Forest classifier, where local patches cast votes for angles. In Section 2.3 we propose an alternative regressor by means of a Convolutional Neural Network model that extracts holistic features from the whole face image. Finally, in Section 2.4 we introduce a new benchmark and compare our results with the literature.

2.1 Related work

The head pose estimation task, also termed 3D rotation, may be categorized according to their application domain. The majority of existing techniques rely on high resolution RGB or RGB-D face images (*i.e.*, RGB-D is a combination of RGB and depth images, where pixels intensities represent the distance between the camera and the corresponding objects). In this chapter, we do not deal with RGB-D information because it requires active sensors, which are expensive and inaccurate in outdoor scenes, limiting their applicability to only laboratory conditions [33]. Thus, we focus on previous literature based on RGB images using faces captured in realistic in-the-wild scenarios. We follow [80] to broadly organize head pose estimation approaches in three groups:

- **Manifold embedding approaches** assume that the head pose is the largest source of variance in facial appearance and propose different algorithms to reduce the high dimensionality of face images. These approaches are based on the assumption that each face image can be projected onto a low dimensional manifold parametrized by the yaw, pitch and roll rotation angles.

  Dimensionality reduction techniques have attracted a lot of attention due to their high accuracy using face images acquired under laboratory conditions, where face appearance is more likely to correlate with the head pose variability. Sundararajan *et al.* [101] (AVM) proposed SIFT keypoints [71] with Geometric Blur descriptors to compute identity-invariant features, and the Nyström approximation method to obtain a low dimensional embedding. Peng *et al.* [85] also modelled the geometry representation of the pose variations, and decouple pose-unrelated factors (identity, expression and illumination) using an instance parametric space where 3D head pose can be estimated.

  Unfortunately, in all these approaches there is no guarantee that, under in-the-wild conditions, the primary subspace components will relate to head orientation rather than to illumination, expression, gender, age, etc.

- **Geometric approaches** detect the location of a set of fiducial points, also termed landmarks, in the image plane (*e.g.*, eye/mouth corners, nostrils, earlobes, the edge of the eyebrows, etc.) and establish correspondences between an explicit deformable mean 3D face shape model and those 2D landmarks (see Chapter 3) [20, 24, 146]. There are many geometric algorithms in the literature that estimate the camera pose from a set of correspondences, referred to as Perspective-n-Point (PnP) problem [38].

  An open source software implementation of this algorithm can be found in the well-known OpenCV library (*solvePnP* function). It employs a Direct Linear Transform (DLT) method to algebraically solve the system of equations determined by the 3D landmarks and their projections onto the image plane. Accordingly, DLT provides

an initial projection matrix, which is iteratively refined using Levenberg-Marquardt optimization to find the parameters that minimize the landmarks reprojection error.

Alternatively, Pose from Orthography and Scaling Iteratively [27] (POSIT) initially approximates the projection matrix with an orthographic projection (*i.e.*, it requires a set of 4 correspondences obtained from non-coplanar points), and then iterates by shifting the landmarks to enhance the perspective model until it reaches convergence. POSIT is more efficient and does not require an initial guess to obtain good accuracy on head pose estimation [4][1].

Geometric methods are prone to struggle in the presence of outliers within the set of landmark correspondences. In this case, RANSAC [38] can be used in conjunction with any of the aforementioned methods to make the final solution more robust to outliers. However, these methods are constrained by the accuracy of the landmark detection, and the similarity of the mean 3D face shape modelling an specific person.

- **Regression approaches** have become more popular because of their robustness to facial appearance changes under in-the-wild conditions. These methods infer a non-linear function that directly maps face image features into a discrete or continuous head pose estimation. Here, we also categorize them into two subgroups according to whether they use traditional machine learning or deep learning strategies.

  The former is usually represented by voting algorithms based on multiple regression trees. Dantone *et al.* [26, 33] exploited the capacity of Random Forests in mapping local features within patches densely extracted from both RGB and RGB-D images. These patches cast votes for the corresponding 3D rotation angles, the detection of the nose tip (allowing to group patches with similar location), and their visibility prediction (ignoring patches extracted from occluded face regions). Hara *et al.* [47] proposed a novel splitting method that allows regression tree nodes to have more than two child nodes, which also outperforms pose estimation results.

  The latter is related to head pose regression through Convolutional Neural Networks (CNNs) using as input the whole face image. These approaches outperform previous work, which suggests that a direct holistic method would be more convenient to deal with this problem. Yang *et al.* [126] uses a simple CNN with 3 convolutional layers, 3 pooling layers and 2 fully connected layers for the regression of yaw, pitch and roll angles to assist face alignment. Patacchiola *et al.* [84] studied the use of LeNet and AlexNet [59] baseline architectures trained with an Adam optimizer and RMSE regression loss. Ruiz *et al.* [92] (HopeNet) highlighted the performance of ResNet-50 [49] baseline pre-trained on ImageNet, by adding two loss functions for both head pose classification and regression. They concluded that their combination performs better than regression solely. Hsu *et al.* [51] (QuatNet) also emphasized the results obtained using GoogLeNet [102] baseline pre-trained on ImageNet, where a novel multi-regression loss function has been added through quaternions instead of Euler angles. Gao *et al.* [39] and Liu *et al.* [68] (DLDL, GLDL) proposed a scheme based on label distribution learning [40] to alleviate the lack of training data for extreme poses, using pre-trained models on VGG-Face [83] (VGG-16 [98]) and ImageNet (ResNet-50 [49]) respectively. Lastly, Yang *et al.* [130] (FSA-Caps-Fusion) proposed a novel fine-grained hierarchical classification of the discretized head pose into a coarse-to-fine scheme followed by a soft regression stage inspired by DEX [91].

[1] In this thesis we follow the implementation in [56] to estimate pose from the annotated landmarks.

As we noted previously, head pose estimation is closely related to other facial analysis problems. In fact, face orientation is often estimated as an auxiliary task in face alignment. Recent work [142, 146, 119, 87, 60, 140, 86] also demonstrates that learning simultaneously another facial correlated problem (*e.g.*, action units, face detection, landmark location, and so forth) with the head pose allows them to improve their performance. In Chapter 4 we discuss these multi-task approaches and introduce a final solution based on a multi-task scheme that establishes a new state-of-the-art on head pose estimation.

At this point, we follow our hypothesis from Section 1.2, which assumes that most face alignment algorithms struggle when they attempt to simultaneously solve both rigid and non-rigid deformation, and consider the problem of estimating only the head orientation. To this end, our principal objective in this chapter is to analyze the performance of two different regression approaches based on the aforementioned traditional machine learning and deep learning methodologies. Thus, we focus on the estimation of 3D rotation angles that parametrize the rigid pose of the head, which allows us to evaluate the accuracy of current methods.

2.2 Head pose classification using Random Forest

In this section, we introduce the first contribution of this thesis [111]. The objetive is to evaluate the results of a head pose regressor based on local features through RGB face images acquired under in-the-wild conditions, similar to Dantone *et al.* [26]. We present a real-time algorithm based on a Random Forest, where patches extracted randomly from the face cast votes for the head pose angles of interest.

The Random Forest is a well-known machine learning algorithm formed by an ensemble of T decision trees g, whose prediction is determined by combining the outputs from all the trees. This technique has been successfully used in a variety of computer vision problems, such as classification, regression and probability density estimation [23]. Moreover, it is a widely used algorithm because it may be trained with a moderately low amount of data and the resultant ensemble can perform in real-time.

Algorithm 1 considers the problem of inferring the discretized yaw angle from a face image by choosing the most probable class according to the predictions of several decision trees $\{g_t\}_{t=1}^{T}$, instead of computing a continuous head pose like [26].

Algorithm 1 Training a Random Forest

Input: $\mathtt{I}$, $\tilde{\mathbf{h}}$, T
  for $t = 1$ **to** T **do**
    // Extract random subset of images $\mathtt{I}_t \in \mathtt{I}$ balanced by head pose
    $(\mathtt{I}_t, \tilde{\mathbf{h}}_t) = \text{chooseImages}(\mathtt{I}, \tilde{\mathbf{h}})$
    // Generate training set of patches using stride s
    $\mathtt{I}_t^{\alpha} = \text{generateChannels}(\mathtt{I}_t)$
    $\mathcal{P}_t = \text{generatePatches}(\mathtt{I}_t^{\alpha}, s)$
    $\Phi_t = \text{generateFeatureCandidates}()$
    // Apply Algorithm 2 for each tree node
    $g_t = \text{fitRegressionTree}(\mathcal{P}_t, \Phi_t)$
  end for
Output: $\{g_t\}_{t=1}^{T}$

This algorithm is built around the "bagging" framework, wherein we use multiple trees trained through different subsets of images $\mathtt{I}_t$ chosen randomly with replacement.

Additionally, compared to [26], where only a rough head pose classification is needed to train an individual facial landmarks detector for each orientation, we demand an accurate estimation of each angle. Hence, we increase the amount of head pose classes by defining intervals of 15° rather than 45°.

2.2.1 Patch-based channel features

From each training image $\mathtt{I}_i$, we randomly choose a subset of square patches around the face, $\mathcal{P}_i = \{(\mathtt{I}_i^\alpha, \tilde{\mathbf{h}}_i)\}$, where $\tilde{\mathbf{h}}_i$ is the discretized head pose and $\mathtt{I}_i^\alpha$ is the appearance of the patch described by channel α [29]. These channels are gray-scale pixels and edges detected using Gabor filters, Sobel and Canny operators, typically used for image analysis, some of which are shown in Fig. 2.2.

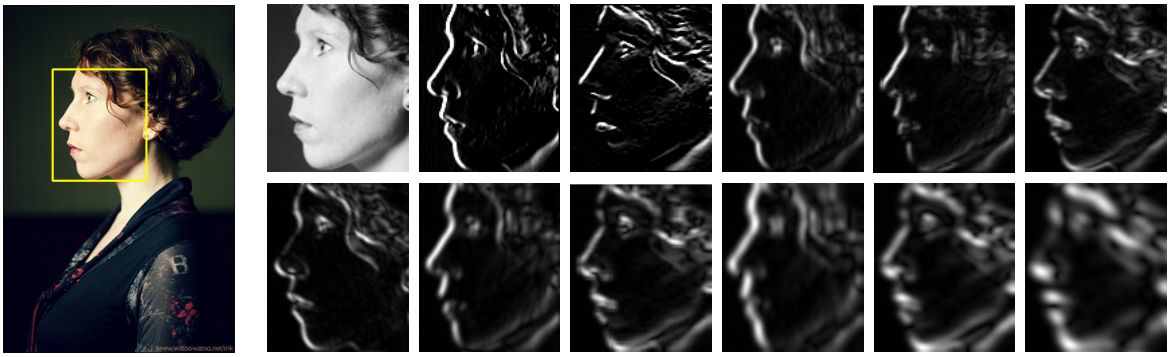


Figure 2.2: Sample $\mathtt{I}_i^\alpha$ channels generated from the input image $\mathtt{I}_i$. The yellow rectangle shows the bounding box detection. The channels provided consist of one gray-scale image, two Sobel borders and nine Gabor filtered images respectively.

Same as Dantone *et al.* [26], our features are the difference between the average values in two rectangles, R_1 and R_2, given a channel α. We describe each of them with the pair of rectangle coordinates within the patch boundaries in image $\mathtt{I}^\alpha$, $\theta = (R_1, R_2, \alpha)$. Thus, given patch p and parameters θ, the feature value is,

$$f(p, \theta) = \frac{1}{|R_1|} \sum_{\mathbf{q} \in R_1} \mathtt{I}^\alpha(\mathbf{q}) - \frac{1}{|R_2|} \sum_{\mathbf{q} \in R_2} \mathtt{I}^\alpha(\mathbf{q}) \tag{2.1}$$

where $\mathbf{q} \in \mathbb{R}^2$ are pixel coordinates.

During training, the splitting nodes from the decision trees in the Random Forest use these local features to find the best channel α and texture subregions R_1 and R_2 allowing to group samples with similar head pose together.

2.2.2 Decision tree learning

In Algorithm 2 we present the training process required to learn how to group patches with similar head pose label within each node in a tree. We require as input a balanced subset of patches $\mathcal{P}$ extracted from the training images and a random pool of candidates Φ. We optimize each weak learner by selecting the candidates $\theta = (R_1, R_2, \alpha)$ and $\phi = (\theta, \tau)$

that maximize the information gain,

$$IG(\Phi) = \mathcal{H}(\mathcal{P}) - \sum_{S \in \{L,R\}} \frac{|\mathcal{P}_S|}{|\mathcal{P}|} \cdot \mathcal{H}(\mathcal{P}_S) \tag{2.2}$$

where τ denotes the threshold on the feature value, $\mathcal{P}_L = \{\mathcal{P}|f(P,\theta) < \tau\}$, $\mathcal{P}_R = \mathcal{P} \setminus \mathcal{P}_L$ represents patches grouped into left and right child nodes respectively, and $\mathcal{H}(\mathcal{P})$ is the class uncertainty measure. In our case, $\mathcal{H}(\mathcal{P}) = 1/2 \cdot log(2\pi e\sigma^2)$ is the Gaussian differential entropy of the patch labels. Consequently, we maximize the information gain by reducing the variance σ^2, which is equivalent to group patches with similar head pose label $\tilde{\mathbf{h}}$.

Algorithm 2 Learning to group patches $\mathcal{P}$ with similar head pose

Input: $\mathcal{P}$, Φ
  $IG^* = 0$
  for $\phi \in \Phi$ **do**
    // Split patches according to feature candidate ϕ
    f = featureComputation($\mathcal{P}$, θ) // following Section 2.2.1
    $\mathcal{P}_L$, $\mathcal{P}_R$ = splitSamples($\mathcal{P}$, f, τ)
    // Evaluate the goodness of the split using ϕ
    IG = evaluateSplit($\mathcal{P}_L$, $\mathcal{P}_R$, ϕ)
    // Parameters that maximize the information gain
    if $IG > IG^*$ **then**
      $\theta^* = \theta$, $\tau^* = \tau$
    end if
  end for
  // Split patches according to the best feature candidate ϕ^*
  f^* = featureComputation($\mathcal{P}$, θ^*) // following Section 2.2.1
  $\mathcal{P}_L$, $\mathcal{P}_R$ = splitSamples($\mathcal{P}$, f^*, τ^*)
Output: $\mathcal{P}_L$, $\mathcal{P}_R$

2.2.3 Head pose classification

Once we have trained the Random Forest, we infer the head pose $\mathbf{h}_i$ for a face image $\mathtt{I}_i$ following Algorithm 3. It aggregates the individual g_t predictions to combine a final estimation based on a majority voting on the individual predictions. We assume the head to be the most prominent object in the bounding box of size $W \times H$ pixels.

As a result, each square patch p of size $N \times N$, extracted from the image using a stride of s, casts one vote for a head pose angle. In comparison with [33], we do not estimate the patch visibility, but we discard votes obtained from high variance nodes. We assume that they are not reliable because they group samples with different head pose label. The final estimation $\mathbf{h}_i$ is the most probable discrete orientation (see Fig. 2.3).

2.3 Head pose regression based on CNNs

Convolutional Neural Networks (CNNs) have been extensively utilized in recent years due to their outstanding performance in various computer vision tasks such as image segmentation [90], object detection [88] or image recognition [98], to name a few. Nowadays,

Algorithm 3 Head pose estimation using a Random Forest

Input: $\mathtt{I}_i$

// Generate dense set of patches using stride $s = 1$

$\mathtt{I}_i^{\alpha} = \text{generateChannels}(\mathtt{I}_i)$

$\mathcal{P} = \text{generatePatches}(\mathtt{I}_i^{\alpha}, s)$

for $p \in \mathcal{P}$ **do**

  for $t = 1$ **to** T **do**

    // The leaf node of each g_t provides a discrete distribution of the head pose

    $p(\mathbf{h}_i|p,t) = \text{evaluateRegressionTree}(p)$

  end for

  // Head pose distribution provided by the Random Forest $\{g_t\}_{t=1}^{T}$

  $p(\mathbf{h}_i|p) = \sum_t p(\mathbf{h}_i|p,t)$

end for

// Final head pose distribution according to all patches $\mathcal{P}$

$p(\mathbf{h}_i|\mathtt{I}_i) = \sum_p p(\mathbf{h}_i|p)$

Output: $\mathbf{h}_i$

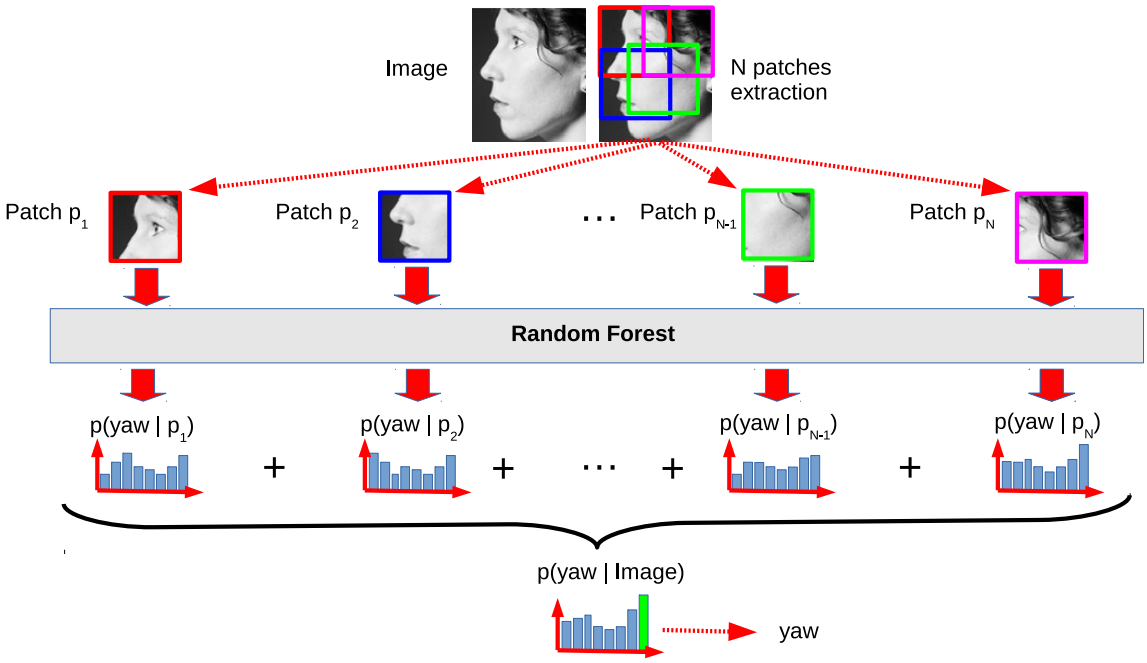


Figure 2.3: Head pose estimation using the most probable discrete orientation in $p(\mathbf{h}_i|\mathtt{I}_i)$.

the depth of these networks has been considered an extremely important component to achieve top results. Their powerful learning ability is related to the extraction of features across multiple stages that automatically learn a hierarchical representation of the object, *i.e.*, early layers identify generic features such as edges and corners, whereas deeper layers consist of higher representations able to recognize specific features. Moreover, CNNs do not demand any human effort designing handcrafted features, compared to other machine learning techniques, because they automatically learn the most suitable feature maps for each problem.

In a preliminary analysis we realized that most state-of-the-art published results are not comparable [3]. We have evaluated a set of CNN-based standard regressors for images acquired in realistic unrestricted conditions. To this end, we follow [39, 84, 92, 51, 68] to

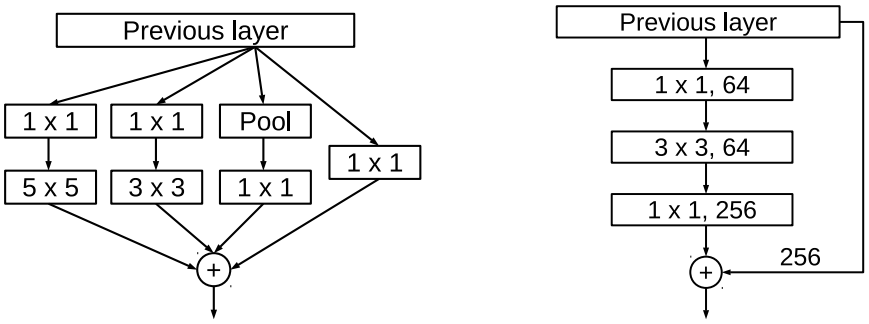
build our baseline regressors around top performing architectures according to the image classification task of the ILSVRC competitions (ImageNet Large-Scale Visual Recognition Challenge). Hence, we go through the details of each CNN to point out their differences in terms of depth, number of convolutional blocks and fully connected layers:

- AlexNet [59] developed by Krizhevsky *et al.* proved that features obtained using deep learning models transcend handcrafted features, breaking the traditional paradigm in the computer vision community. It also incorporates a non-linear activation ReLu function after each convolutional layer instead of Tanh or Sigmoid, and it adds a Dropout layer after each intermediate fully connected layer to deal with overfitting. The whole architecture consists of 8 layers including 5 convolutional layers of 11×11, 5×5 and 3×3 and 3 fully connected layers.

- VGG [98] replaced previous large filters using the concatenation of 3×3 convolutional layers to achieve same effective receptive field. The two main advantages of using smaller filter sizes are a decrease in the number of parameters and the addition of extra ReLu layers (more non-linearity gives extra power to the regressor). In [3], we evaluate both VGG-16 and VGG-19 configurations formed by 5 blocks with 13 and 16 layers respectively and 3 fully connected layers.

- GoogLeNet [102] introduced a creative block of convolutional layers called "inception module" (see Fig. 2.4a) designed to reduce the huge computational requirements of deep networks, both in terms of memory and time. The inception module is based on the combination of 5×5, 3×3 and 1×1 parallel convolutional layers that extract features at different resolutions at the same level. To make GoogLeNet architecture computationally reasonable, Szegedy *et al.* also reduce the number of parameters by adding extra 1×1 convolutional layers before large 5×5 and 3×3 filter sizes, to reduce the number of feature maps.

- ResNet [49] noticed that state-of-the-art models based on deep architectures are hard to train because of the notorious vanishing gradient problem and the lack of effective optimization techniques. He *et al.* introduced the "residual block" to distribute the gradient by using skip connections or shortcuts that learn the identity mapping (see Fig. 2.4b). As a result, they were able to train deep ResNet-50, ResNet-101 and ResNet-152 configurations.



(a) Inception module (b) Residual block

Figure 2.4: Creative configurations which improve performance and efficiency [102, 49].

At this point, we notice that most deep architectures require an immense amount of images for training, and sometimes such large data sets are not readily available or are time consuming to acquire. For the head pose estimation task, we follow [92, 51, 68] and employ a model pre-trained on ImageNet to avoid learning previous baseline architectures from scratch. This training strategy, known as *transfer learning*, allows us to extract the knowledge from one or more source tasks, and apply the resulting model to our target task at hand, being source and destination tasks similar. In Fig. 2.5, we show the same transfer learning procedure followed in [3], where the entire pre-trained model weights are fine-tuned and the softmax loss layer (classification) is replaced by an euclidean loss layer, $\mathcal{L}_P$ (see Eq. 2.3), for modelling the yaw, pitch and roll angles (regression).

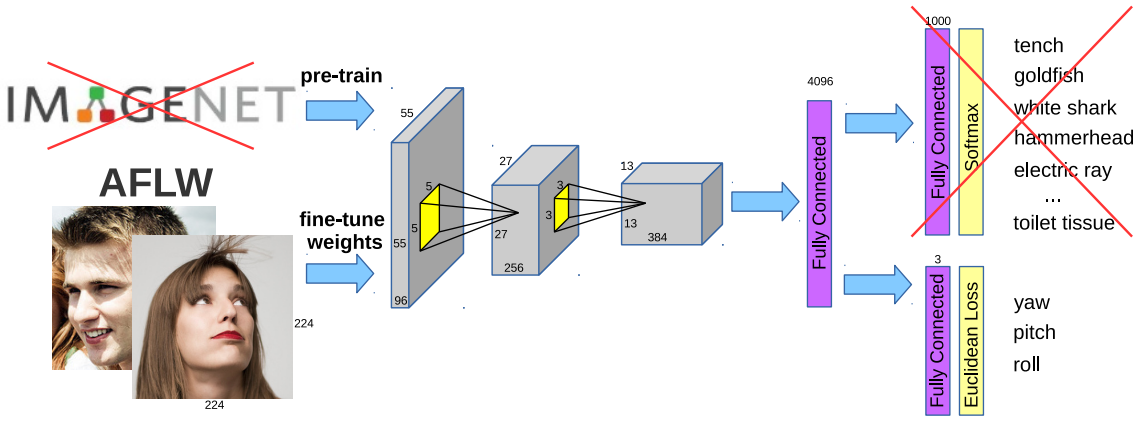


Figure 2.5: Transfer learning scheme using pre-trained ImageNet weights as initialization to train the regressor for head pose estimation. Red-crossed input and output layers have been replaced by AFLW database and euclidean loss respectively.

As we mentioned previously, our main goal in this chapter is to estimate the head pose, which is formed by three Euler rotation angles (see Fig. 2.1). To this end, we minimize directly the estimated pose values (*yaw*, *pitch*, *roll*) using the following euclidean distance L_2 function,

$$\mathcal{L}_P = \sum_{i=1}^{N} ||\tilde{\mathbf{h}}_i - \mathbf{h}_i||_2 \tag{2.3}$$

where N is the number of face images and $\tilde{\mathbf{h}}_i$, $\mathbf{h}_i$ are the ground truth and predicted head pose parameters for the i-th training image.

Transfer learning have won great popularity in deep learning [69, 39, 92, 51, 68] given the large amount of resources required to train CNNs and the small amount of annotated data on which deep networks are trained. Using a pre-trained model leads to a decrease in the training time and a lower generalization error [131] due to its better initialization and more discriminative feature maps (see further discussion in Section 4.1). However, transfer learning also results in a major training complexity because it requires an additional preliminary training. It is typically avoided by using a baseline architecture pre-trained in a standard data set (*e.g.*, ImageNet) but, in case of evaluating a different network design, we require to perform a previous training.

In this thesis we indicate those approaches that fine-tune using an additional data set, because we consider unfair the comparison with previous literature, which did not utilize any additional images and annotations during its training.

2.4 Experiments

In this section, we review the performance of preceding literature based on head pose estimation using face images acquired under in-the-wild conditions. We compare previous research with our proposals, Random Forests [111] and CNNs [3]. However we notice that their published results are sometimes not strictly comparable due to their inconsistencies using different training/testing partitions or fine-tuning from an external database [3]. As discussed previously, this is the reason why we decided to create our own benchmark.

2.4.1 Database

Traditionally, head pose estimation methods have been evaluated with public data sets acquired under laboratory conditions, because of the difficulty in accurately estimating the yaw, pitch and roll angles in more realistic situations [80]. Nevertheless, accurate pose evaluation in unrestricted scenarios is an open issue. We propose a common methodology to perform experiments using the following in-the-wild data sets:

- AFLW [56] provides a collection of 25993 in-the-wild faces, with head pose ranging between $\pm 120°$ for yaw and $\pm 90°$ for pitch and roll angles. These angles have been computed assuming the structure of a mean 3D face and using 21 manually labelled facial landmarks, depending on their visibility, using the POSIT algorithm [27]. We have found several annotation errors and, consequently, removed these faces from our experiments. From the remaining face images we have randomly chosen 19312 and 4828 instances for training and testing respectively. We made this benchmark publicly available.

- AFW [146] has been traditionally used only for testing purposes. This small database has 250 images with 468 faces in quite challenging settings. It provides discrete yaw labels ranging from -90° to 90° with 15° intervals, plus the face bounding box. These labels were manually annotated, hence often they are not very accurate.

- 300W [93] is the most popular face alignment benchmark. We follow the established approach and we use the 689 faces from the public competition set as test. Although it does not provide any pose information, similar to Yang *et al.* [126][2], we use again the POSIT algorithm [27] with 68 manually annotated landmarks to estimate the three angles for each instance. Note also that our labels may also have small errors caused by the assumption that all faces have the same 3D structure.

Additionally, we have chosen Pointing-04 [111] to evaluate discretized head pose results with the traditional algorithms acquired in a controlled scenario. It contains 2790 images of 15 subjects spanning discrete yaw and pitch poses from -90° to 90° with 15° interval. It provides a coarse ground truth obtained by asking subjects to direct their heads toward a set of markers placed around them in a room.

2.4.2 Evaluation metrics

We use common evaluation metrics to quantify the head pose estimation error. First, we employ the Mean Absolute Error (MAE) metric, which represents a normalized version

[2]They compute head pose using a mean 3D face composed of 49 points. Unfortunately, these labels are not publicly available.

of the manhattan distance L_1,

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left(|\tilde{\mathbf{h}}_i - \mathbf{h}_i|\right), \tag{2.4}$$

where N is the number of face images and $\tilde{\mathbf{h}}_i$, $\mathbf{h}_i$ represent the ground truth and predicted Euler rotation angles respectively.

Since AFW provides discrete classification results, we show a confusion matrix, where we discretize each angle in steps of 15° {-90°, -75°, -60°, -45°, -30°, -15°, 0°, +15°, +30°, +45°, +60°, +75°, +90°}. We also compute the classification accuracy (*Accuracy*) as the percentage of images properly labelled with the correct or with the adjacent class labels ($MAE \leq 15°$), which allows us to compare our results with the literature.

2.4.3 Implementation details

To train our algorithms, we shuffle the training set of each database and split it into 90% train and 10% validation subsets [111, 3]. We have trained our algorithms using the training/testing partitions of AFLW defined in Section 2.4.1. All the experiments follow the settings detailed below:

Random Forests (Section 2.2)

We use a Random Forest with $\mathcal{T} = 20$ decision trees trained from a randomly selected subset of faces from AFLW, equally distributed by yaw angle (700 images per discretized angle). We use ground truth annotated bounding boxes resized to 105×125 pixels. For training we randomly extract $|\mathcal{P}| = 20$ patches of 61×61 pixels from each face image. It is worth mentioning that the performance of the algorithm is quite sensitive to this patch size because a smaller region would not capture enough information to predict the head pose, whereas a larger patch would produce results more sensitive to occlusions.

We train each tree node by selecting the best parameters from a collection of $|\Phi| = 50$K samples computed from $|\theta| = 2$K combinations of $[R_1, R_2, \alpha]$ and $|\tau| = 25$ thresholds. The maximum random size of the subregions defining the asymmetric areas R_1 and R_2 is set to be lower than 75% of the patch size. Tree growing stops when the depth reaches 15, or if there are less than 20 patches in a leaf. We also filter out leaves with a maximum variance threshold set to 400 (*i.e.*, training samples were grouped with a completely different head pose label $\tilde{\mathbf{h}}$). This limits the impact in the final prediction of non-informative leaves.

A crucial test-time parameter is the stride, s, controlling the sampling of patches. We process only 1 out of 10 possible patches. Test values have been empirically tuned to find the desired trade-off between accuracy and temporal efficiency of the estimation process, making the algorithm adaptive to the constraints of different applications. Our algorithm outperforms its competitors in terms of computational requirements. It achieves a frame rate of 83 FPS (12 ms per image) on an Intel Core i7 CPU processor at 3.60GHz with 8 cores multi-threaded, 300 times faster than Zhu *et al.* [146] (TSPM).

Convolutional Neural Networks (Section 2.3)

We follow same procedure for AlexNet [59], GoogLeNet [102], VGG [98] and ResNet [49] baseline models. For training, we use 19312 face images and ground truth bounding boxes

provided by AFLW resized to 224×224 pixels. We always select the model parameters with lowest validation error.

The CNNs are fine-tuned from ImageNet using Nesterov Accelerated Gradient Descent (NVG) optimizer with an initial learning rate set to 10^{-5}, which is reduced by a $\gamma = 0.1$ factor each 10K iterations or 13 epochs approximately. We set momentum and weight decay to $\mu = 0.9$ and $\lambda = 5 \cdot 10^{-4}$ respectively.

We train each model in a NVidia GeForce GTX 1080 GPU (8GB) using Python, Caffe and OpenCV libraries. We optimize the memory occupation by setting the batch size to 24 and the maximum number of iterations to 25K. However, deeper networks like ResNet use a smaller batch size and larger number of iterations because of their immense memory requirements. As a result, ResNet-152 baseline architecture achieves a frame rate of nearly 9 FPS (111 ms per image) without using any GPU (see Table 2.1), which is almost 10 times slower than our Random Forest implementation.

2.4.4 CNNs baseline study

In Table 2.1 we present the analysis of the most relevant CNNs proposed in the state-of-the-art for image recognition. For each baseline, we test multiple performance indices, such as the *Accuracy*, *MAE* and inference time required on the CPU. Further, we analyze the relationship between these indices and the architectures of interest.

Method	AFW	AFLW				300W public				Frame rate
	yaw	yaw	pitch	roll	mean	yaw	pitch	roll	mean	mean
	Accuracy	*MAE*	*MAE*	*MAE*	*MAE*	*MAE*	*MAE*	*MAE*	*MAE*	*FPS*
AlexNet [59]	86.32%	6.28	5.02	3.36	4.88	6.86	6.61	5.82	6.43	**52.13**
VGG-16 [98]	85.68%	6.23	4.96	3.35	4.84	6.35	7.02	5.98	6.45	9.31
VGG-19 [98]	94.23%	5.78	4.79	3.20	4.59	5.56	6.35	4.65	5.52	7.51
GoogLeNet [102]	**95.51%**	6.40	5.31	3.74	5.15	5.71	7.99	6.85	6.85	47.28
ResNet-50 [49]	94.44%	6.00	4.90	3.14	4.68	5.71	5.91	3.23	4.95	23.63
ResNet-101 [49]	94.44%	**5.59**	**4.79**	**2.83**	**4.40**	**5.13**	**5.87**	**3.03**	**4.67**	12.79
ResNet-152 [49]	94.01%	5.61	4.79	3.03	4.47	5.52	6.16	3.18	4.95	8.93

Table 2.1: Head pose estimation results and frame rate for each baseline architecture.

In general, these results confirm that the deeper the architecture, the better the results, which is a well-known fact in the deep learning literature [49]. However, we notice that `ResNet-152` obtains slightly worse performance than `ResNet-101`, because a smaller batch size produces lower generalization error and poor batch normalization. Thus, `ResNet-101` provides the best overall performance in AFLW and 300W testing sets. Surprisingly, `GoogLeNet` obtains the highest *Accuracy* in AFW, perhaps because the yaw discretization played against the deepest regressors.

On the one hand, we reach the conclusion that computational complexity and head pose estimation are somehow unrelated because `VGG-16` and `VGG-19`, which are the most demanding in terms of computational requirements, perform worse than `ResNet-101` and `ResNet-152` models. On the other hand, the time required for each CNN is intrinsically related to the computational cost since `VGG-19` is 1.4 FPS slower than `ResNet-152` (see Table 2.1). All these baseline models achieve real-time performance on a GPU, whereas only `AlexNet` and `GoogLeNet` guarantee a suitable frame rate on a CPU.

2.4.5 Comparison with the state-of-the-art

Published results in the literature are sometimes not fully comparable since there was no standard protocol or benchmark to evaluate head pose in-the-wild. In this section, we compare our trained models based on Random Forest and CNNs with other approaches in the literature, by using their published results categorized according to whether they have included additional data sets during training.

Published results training from scratch

First, we compare our Random Forest proposal [111], trained only with AFLW images, with other contemporary published methods that do not include any additional data sets for training purposes. Therefore, we consider the problem of inferring the discretized yaw angle from face images appearing in real-world situations. Yaw and pitch rotations are the most informative for interpersonal communication and cause the largest appearance changes. For this reason, the majority of approaches [47, 40, 101, 39] only estimate one of them or both.

In Table 2.2 we report the yaw angle estimation performance in both AFW and AFLW (testing subset) realistic benchmarks. The published results are still not fully comparable because each approach used different training/testing AFLW partitions, and none of these subsets are publicly available, hence it is impossible to make a fair comparison among any of these methods. In [3], we released our training/testing AFLW annotations to define a common evaluation benchmark, so we expect that future algorithms will be compared on fair grounds.

Method	AFW yaw *Accuracy*	AFLW yaw *MAE*
TSPM [146]	81.0%	46.54
AVM [101]	82.5%	17.48
Peng *et al.* [85]	**86.3%**	-
Valle *et al.* [111]	83.54%	**12.26**

Table 2.2: Error of head pose estimation approaches using AFW and AFLW testing set. These methods have been trained with no additional database.

Similarly, the published results for AFW are not fully comparable because each method was trained in a different AFLW training subset, but may be useful for further analysis. Additionally, Peng *et al.* [85] test on a reduced subset of 459 faces because they only use faces larger than 64×64 pixels.

Our Random Forest outperforms [146, 101, 85] using AFLW, both in terms of *MAE* and classification *Accuracy* with an error less than or equal to 15°. These results prove the powerful representational ability of features based on local patches with a non-linear regression algorithm. We achieve a reduction in *MAE* of 29.8% compared to our main competitor (AVM).

Finally, we visually compare only the yaw angle estimation obtained under laboratory (Pointing-04) and in-the-wild (AFLW) conditions respectively (see Fig. 2.6). It is worth noticing that our Random Forest algorithm based on local features [3], not only achieves competitive results using faces acquired under unconstrained scenarios (see Table 2.2), but

also performs well in laboratory conditions, obtaining similar performance as coetaneous holistic approaches [47, 40].

(a) Pointing-04

(b) AFLW

Figure 2.6: Sample images of yaw estimation results in Pointing-04 and AFLW data sets. Green and blue arrows indicate respectively yaw angle estimation and ground truth.

Published results training using additional data

We compare the results of our top performance CNN baseline proposal [3], fine-tuned from ImageNet weights, with recent work that uses a model pre-trained on an additional database [39, 84, 92, 51, 68], or a simultaneous learning through correlated facial tasks [87, 60, 140, 86]. Here, we consider the problem of estimating the head pose by regressing the yaw, pitch and roll head angles using the whole face image as input.

In Table 2.3 we report yaw classification *Accuracy* in AFW and rotation angles errors on AFLW (testing subset). As mentioned above, we have proposed a common benchmark to establish a fair comparison among different methods. This is an important unmet need since there is no common evaluation protocol. For example, Patacchiola *et al.* [84] applied a five-fold cross validation procedure, Gao *et al.* [39] have randomly chosen 7848 testing face images, Ranjan *et al.* [87, 86] selected randomly 1000 testing face images and the rest for training purposes, and so forth.

When confronted with the best published results in the literature, although not strictly comparable, our baseline model using AlexNet presented in Table 2.1 (4.88 *MAE*) achieves better performance than [84, 86] using the same architecture (see Table 2.3). We get an average Euler angles error reduction of 17% compared to Hyperface [86] (5.88 *MAE*). Similarly, our GoogLeNet baseline model outperforms Kepler [60], in spite of using the same network design. However, it is worse than QuatNet [51] perhaps because its model optimizes a multi-regression loss function which combines the L_2 regression and an ordinal regression loss to address that the appearance variability is lower in extreme poses. For VGG, our baseline result is better than DLDL [39] using the same architecture, because our proposal was pre-trained on ImageNet generic data instead of VGG-Face [83]. This is a remarkable result, since ImageNet is a general data set whereas VGG-Face has only faces. We conjecture that the low-level features learned from ImageNet are more discriminative than those from VGG-Face. Finally, our ResNet baseline result (4.40 *MAE*) also performs better than our main competitors HopeNet [92], HF-ResNet [86] and GLDL [68] using the same architecture. Thus, we achieve an average Euler angles error reduction of 11.1% compared to HF-ResNet [86] (4.95 *MAE*).

Method	AFW	AFLW			
	yaw	yaw	pitch	roll	mean
	Accuracy	*MAE*	*MAE*	*MAE*	*MAE*
Patacchiola *et al.* [84]	-	11.04	7.15	4.40	7.53
DLDL [39]	-	6.60	5.75	-	-
Kepler [60]	96.67%	6.45	5.85	8.75	7.01
Hyperface [86]	97.7%	7.61	6.13	3.92	5.88
AIO [87]	99.1%	-	-	-	-
HopeNet [92]	96.15%	6.26	5.89	3.82	5.32
HF-ResNet [86]	98.5%	6.24	5.33	3.29	4.95
GLDL [68]	**99.7%**	6.00	5.31	3.75	5.02
CCR [140]	-	5.22	5.85	2.51	4.52
QuatNet [51]	-	**3.93**	**4.31**	**2.59**	**3.61**
Amador *et al.* [3]	94.44%	5.59	4.79	2.83	4.40

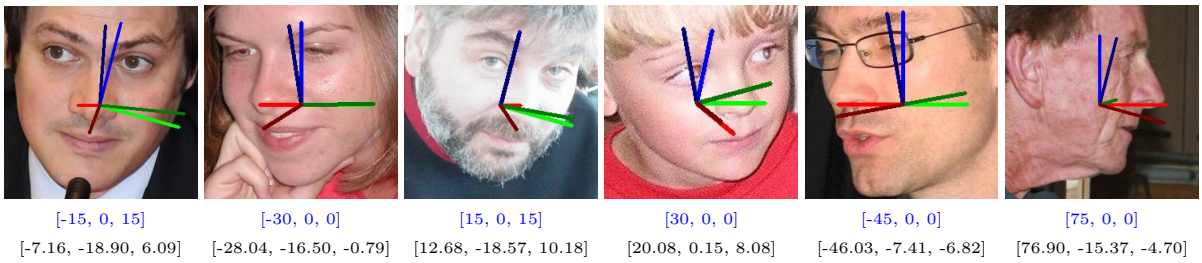
Table 2.3: Error of head pose estimation approaches using AFW and AFLW testing set. These methods have been trained including a model pre-trained in an additional database.

Differently, previous literature reported a better classification *Accuracy* in AFW than those achieved by our CNN baseline regressors. This fact is quite surprising since in the more precise AFLW regression case, the result is the opposite. Probably the presence of inaccurate landmarks annotations harmed POSIT [27] performance due to outliers (see Fig. 3.13), or the manual annotation errors induced by discrete angle labels in AFW. In fact, as can be observed in Fig. 2.7, sometimes our estimation seems to be more accurate than the ground truth (*e.g.*, see first, fourth and sixth faces in Fig. 2.7a).

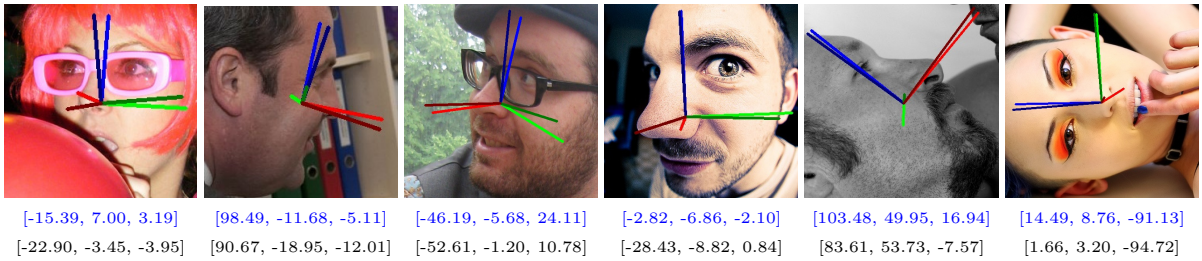
In addition, we evaluate our trained deep networks on the 300W test set following [126], but our ResNet-101 baseline result (4.67 *MAE*) is not as good as the one provided by Yang *et al.* [126] (3.93 *MAE*) because they have trained their CNN using the 300W training set instead of AFLW, and there is a noticeable correlation between face images from the same database, known as "data set bias" [105] (see Section 3.5.5).

In Fig. 2.7 we show some representative face images considered as errors, where the sum of head pose estimation errors (yaw, pitch and roll angles) achieves a *MAE* greater than 15°, using ResNet-101 baseline model. We visually perceive the robustness of CNNs under in-the-wild conditions using the whole face image as input. It seems that yaw and pitch angles are typically the most difficult to successfully estimate, perhaps because both produce the largest appearance changes in the expressive parts of the face. Consequently, we consider that the problem of inferring the head pose is still far from being completely solved under realistic conditions, such as extreme orientation variations (*e.g.*, fifth face in Fig. 2.7b), partial occlusions (*e.g.*, second face in Fig. 2.7c), or perspective distortion artefacts observed in close range photographs (*e.g.*, fourth face in Fig. 2.7b).

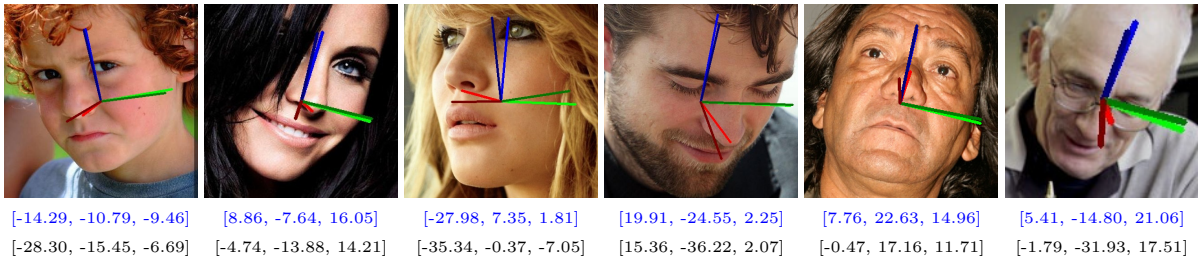
The results from this chapter point towards the idea that most relevant CNN baseline models [3] have outperformed traditional machine learning methods such as our Random Forest proposal [111], following the same common evaluation benchmark. In Fig. 2.8 we compare the classification *Accuracy* between our Random Forest [111] (83.54%) and our GoogLeNet baseline [3] (95.51%) by computing each confusion matrix for the yaw angle using AFW. The element (i, j) of each confusion matrix, where i and j are row and column identifiers respectively, represents the number of instances belonging to i that have been



(a) AFW



(b) AFLW



(c) 300W public

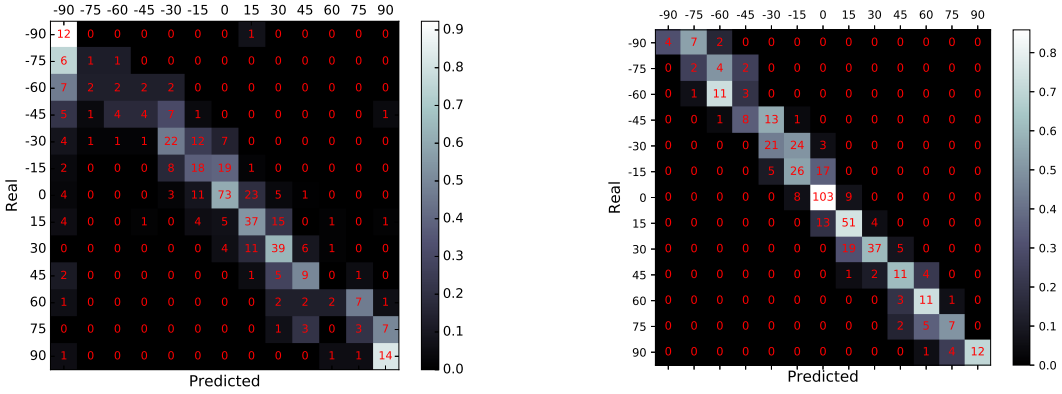
Figure 2.7: Representative errors using ResNet-101 in AFW, AFLW and 300W database. The blue and black coloured text below represent ground truth and predicted angles [yaw, pitch, roll].

classified as j. Hence, the elements in the diagonal are the instances correctly classified, whereas the elements off the diagonal are misclassified. The colour intensity represents the percentage of success for each class (see bar on the right side).

As can be seen, there are 120 face images with head pose annotation set to 0° in AFW. Thus, using our Random Forest [111] we estimate correctly 73 faces, which represents a 60.8% of success, whereas using GoogLeNet [3] the number of correct predictions is higher, 103 faces, which represents a 85.8% of success. In this case, most incorrect predictions are adjacent to the proper ground truth angle, which implies that previous *Accuracy* metric reaches a 89.1% (Random Forest) and 100% (GoogLeNet) respectively (*i.e.*, it considers as success adjacent ±15° predictions). In general, the largest errors are in classes between ±45° and ±90° probably due to the notorious lack of training data for large head rotations.

Conclusions

We have tested two regression methods based on classic machine learning and deep learning techniques to estimate the head pose of face images acquired under in-the-wild scenarios. Presumably, the local handcrafted features presented in those regressors, which map different regions to directly estimate the head pose, should be more accurate under partial occlusions or exaggerated facial expressions. Nevertheless, our results demonstrate



(a) Random Forest [111]

(b) GoogLeNet [3]

Figure 2.8: Confusion matrices of yaw classification results with intervals of 15° comparing both Random Forest [111] and GoogLeNet [3] models using the 468 test faces from AFW. Each cell (i, j) indicates the number of faces belonging to i that had been classified as j.

that the discriminative holistic features extracted from CNNs outperform previous head pose literature by a large margin. However, it is also worth noticing that their comparison in this chapter is not completely fair because our Random Forest proposal has not included any additional database, *e.g.*, ImageNet, as pre-train during its optimization.

As we mentioned previously, the main goal of this chapter has been to generate some preliminary results to determine whether it is possible to estimate accurately the rigid face deformation (*i.e.*, 3D rotation and translation of the face with respect to the camera system). In this way, we would enhance the face alignment problem by means of training a regressor that focuses only on the remaining non-rigid face deformation. From now on, we will review how to perform this non-rigid estimation through the detection of multiple landmarks around the face. In Chapter 4, we propose a multi-task framework, where we set the state-of-the-art results in head pose estimation using a multi-task CNN to optimize the head pose through the location of these landmarks of interest.

Chapter 3

Facial landmark detection

The localization of facial landmarks aims to estimate the projection of a set of fiducial points (*e.g.*, eye/mouth corners, nostrils, earlobes, the ends of the eyebrows, etc.) onto the image plane. These landmarks are essential in *face alignment* to compute the non-rigid facial parts deformation. We also utilize them to infer the head pose through a geometric approach (see Section 2.1). The position of these landmarks may denote a unique location of a facial component, also termed distinct points (*e.g.*, see 24 landmarks in Fig. 3.1) or an interpolated point around the facial contour (*e.g.*, see annotations along the face contour in Fig. 4.6). The precise location of these landmarks captures both the rigid and non-rigid deformation due to head movements and facial expressions respectively.

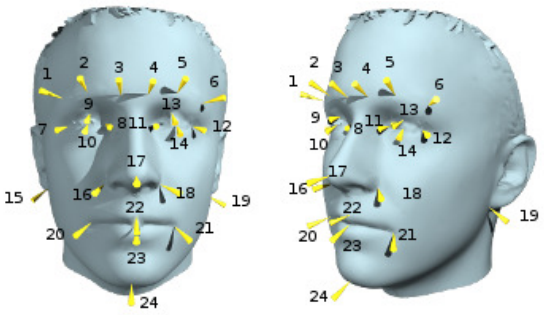


Figure 3.1: Sample subset of distinct landmarks defining the face shape.

The detection of all landmarks in a face define its "shape" within the image. Robustly locating all landmarks in a face is a challenging problem under in-the-wild conditions, due to both the rigid (rotation, translation and scale representation of the face) and non-rigid deformation. State-of-the-art approaches typically fail or lose accuracy in the presence of strong deformations, ambiguous configurations, occlusions produced by other objects, or self-occlusions due to extreme head pose orientations. In Fig. 3.2 we visually observe the imprecise performance of a simple facial landmarks detector [55][1] using some challenging and realistic face images from the WFLW database [117]. In this case, there is a prominent lack of precision. The main question is how to accurately fit these landmarks under large non-rigid face deformations and occlusions, while preserving the face shape.

During the last decade, the landmark detection problem has received much attention from the computer vision community [53, 122], since it is a usual pre-processing step used to improve the performance of other *facial analysis tasks*, *e.g.*, identity recognition [103, 99], facial attributes classification [62, 141], expression recognition [46, 64, 115], action units detection [121, 94], gaze estimation [45, 109], face reenactment [118, 82] and super-resolution [15], to name a few. Traditionally, most methods have aligned face images as a preliminary step to improve their robustness against extreme poses and occlusions [52].

[1] C++ implementation from well-known Dlib library (http://dlib.net/).

Figure 3.2: Representative landmark predictions using the standard Dlib library [55] in the WFLW testing data set. The blue and green landmarks represent ground truth and predictions respectively.

In this thesis, we have principally focused our work on the facial landmark detection problem as a crucial task inherently related to *face alignment*. Nowadays, facial landmarks are attracting widespread interest in computer vision by themselves, because they are intrinsically linked with other applications such as face tracking [96], whose goal is to follow the trace of detected human faces within a video sequence. Traditionally, most tracking methods have focused on the continuous localization of the rigid bounding box provided by a face detector, however these methods do not capture the non-rigid face deformations that are crucial for other facial analysis applications. By tracking the position of the facial landmarks, we capture this non-rigid deformation of interest. This is a challenging task because faces vary pose and appearance over the time, which means that what is being tracked will not look the same in every frame.

The chapter is structured as follows. In Section 3.1 we categorize recent facial landmark detection literature and review the most prominent methods within each category. In Section 3.2 we present a real-time regressor whose goal is to outperform previous Dlib library predictions [55] (see Fig. 3.2) using a coarse-to-fine strategy better suited to handle facial parts deformation. In Section 3.3 we introduce a two-stage cascade of networks, termed CHR2C, based on an encoder-decoder CNN architecture that produces a heatmap for each landmark. The second CNN is trained to refine the predictions of its predecessor, and estimates the landmark coordinates from these heatmaps. In Section 3.4 we propose a novel hybrid method, termed 3DDE, which combines a CNN using previous encoder-decoder architecture and a coarse-to-fine regressor to properly preserve the face shape. Finally, in Section 3.5 we study the performance of CHR2C and 3DDE compared to the state-of-the-art results.

3.1 Related work

Recent literature focuses on developing robust and effective algorithms to handle facial landmark detection using in-the-wild annotated data sets. The majority of methods may be categorized depending on how they model facial texture (*i.e.*, holistic features from the whole face image vs local handcrafted features around the landmarks) and how they preserve the facial shape (*i.e.*, parametric shape model constraint vs alignment without any explicit shape model) respectively [122]. Thus, we organize three principal categories according to whether they employ an explicit head model to parametrize the shape, they implicitly keep the shape, or they have no shape constraint. In each group, we organize recent work in subgroups according to their features.

Approaches explicitly imposing a head model

These approaches commonly deal with how to handle partial face occlusions to ensure that the estimation is a valid face shape, by considering landmark spatial relationships. Among the parametric methods that use an explicit shape model, we organize those using holistic (*e.g.*, AAM, 3DMM) and local parts based features (*e.g.*, ASM, CLM) respectively:

- **Local feature methods** align facial landmarks independently and describe locally the facial appearance as a set of several independent patches, constrained by a global shape representation. Here, Active Shape Models (ASM) [22] combine a generative appearance model for each part, and a global shape model typically learned via PCA. Principal Component Analysis (PCA) is a statistical procedure that allows to learn the mean shape and appearance bases from a collection of normalized annotations.

  We refer to those methods combining independent local detectors and a shape model as Constrained Local Models (CLM) [24]. Belhumeur *et al.* [8] popularized the SVM classifiers using SIFT features [71] to learn the appearance, and an exemplar-based method combined with RANSAC [38] to drawn a valid shape model. Zhu *et al.* [146] (TSPM) compute HOG features and a deterministic model in terms of a tree structure to enforce the face shape conditioned to head pose estimation. Cootes *et al.* [21] use a Random Forest regressor to individually accumulate votes for each landmark location, and incorporate an explicit model to preserve the face shape. Baltrusaitis *et al.* [5] utilize a landmarks detector to produce probability maps that learn spatial relationships between landmarks within a Mean-Shift optimization. These CLM are computationally expensive when the number of landmarks is high, but compared to holistic features, these algorithms are less dependent on the initialization and better suited to handle partial occlusions due to their local nature.

- **Holistic feature methods** learn the correlations among global facial appearance and shape patterns. Traditionally, most successful algorithms were based on 2D and 3D generative approaches, such as, Active Appearance Models (AAM) [20] and 3D Morphable Models (3DMM) [10] respectively.

  Holistic approaches solve shape and texture parameters simultaneously by iterative optimization of the error (*i.e.*, Gauss-Newton optimization), which minimizes a full measure of misalignment between the synthetic face and the input image. The main problem being that Gauss-Newton fitting process is time consuming and suffers from the local minimum problem. Alabort-i-Medina *et al.* [1] formulate a bayesian AAM to model the texture coefficients as hidden variables, and marginalize them out to deal with shape coefficients. Muñoz *et al.* [78] introduce an efficient technique for fitting a 3DMM to the target image based on a locally linear mapping from 3D shape to the projective plane. Tzimiropoulos *et al.* [108] optimize together a flexible appearance model along with a global shape model via Gauss-Newton non-linear optimization. Muñoz *et al.* [79] also study the issue of computational efficiency for Gauss-Newton in the context of image alignment.

  AAM are often criticized because of the complexity of optimizing shape and texture parameters simultaneously, and their difficulty in handling occlusions, but they get accurate results when they are well initialized and appropriate fitting strategies are employed. In fact, there has been an increasing amount of recent research on how to explicitly fit a 3D face shape model using deep learning techniques. Jourabloo *et al.* [54] (PIFAS) and Zhu *et al.* [145] (3DDFA) estimate the 2D landmarks with

their visibility labels from the surface normals of a 3DDM, whose model coefficients are refined using a cascade of CNNs. The main obstacle of 3DDM algorithms reside in the construction process, which requires precise 3D scanning of several subjects and the computation of dense correspondences between the scans [11].

Following the local-based methods strategy mentioned above, there are also holistic approaches that employ CNN regressors to generate multiple heatmaps, representing the probability of locating each landmark in the input face image. Zhang *et al.* [138] (ECT) used these heatmaps to progressively refine an explicit shape model and learn spatial relationships among landmarks using a Mean-Shift procedure, similar to [5]. Zhang *et al.* [139] (ECSAN) also proposed to fit a valid shape model, selected using an exemplar-based method, to the probability maps obtained from a CNN. However, the results of the these approaches in current benchmarks are not as good as those preserving implicitly the face shape without fitting an explicit head model, probably due to the inflexibility of a rigid model to deal with the large deformation required in extreme non-rigid face deformations.

Approaches implicitly preserving face shape

These approaches mostly focus on how to extract features that are robust against non-rigid deformations, without explicitly adding any face shape constraint. Among current methods, there is a subset [17, 16, 55, 63] based on learning how to rank or combine face shape hypotheses to implicitly obtain a valid shape preserving result:

- **Cascaded regression methods** start face alignment from an initial estimation of the landmarks and progressively update their location across different stages. They learn a function mapping between image texture and landmark coordinates by minimizing the alignment error in each stage. Hence, they do not require an explicit shape model. Dollar *et al.* [30] were the pioneers to introduce the popular Cascade Shape Regressor (CSR) framework that lets us train a cascade of boosted weak regressors to learn effective models, with few training images.

  At this point, we build on the large amount of research over the last decade that has enhanced the traditional CSR framework. Cao *et al.* [17] (ESR) proposed a Gradient Boosting regression framework [48]. They prove that whenever the initialization is a valid face shape, their predictions lie on the subspace spanned by training shapes, *i.e.*, the underlying face shape constraint can be preserved implicitly without using any parametric model. Kazemi *et al.* [55] (ERT) improved the original framework by proposing a real-time Ensemble of Regression Trees whose performance has been broadly evaluated thanks to the popular Dlib library (see Fig. 3.2). Burgos-Artizzu *et al.* [16] (RCPR) added landmarks visibility information to avoid the propagation of errors in features extracted around non-visible landmarks. Ren *et al.* [89] (LBF) introduced local binary features to boost the performance up to 3000 FPS. Lee *et al.* [63] (cGPRT) added local features at various scales using the FREAK pattern [2], and substituted previous Gradient Boosting scheme with an optimization based on Gaussian Processes which outperforms alignment results by reducing the overfitting. Xiong *et al.* [124, 125] (SDM) apply SIFT features [71] around landmarks and learn a linear regressor by dividing the search space into individual regions with similar gradient directions. Wu *et al.* [120] refined the landmark location and their visibility together within the CSR to handle severe occlusions and self-occlusion due to large

head rotations. Feng *et al.* [37] (DAC-CSR) proposed a final regression stage to designate an appropriate domain-specific regressor for further landmarks refinement. Zhu *et al.*'s work [144] (CFSS) revolved around how to get better initializations, because CSR is sensitive to the starting point of the regression process. In fact, the majority of these methods are computationally efficient because each stage of these cascaded regressors extracts local measurements around landmarks predicted by its predecessor, also known as shape-indexed features [128]. However they only achieve competitive results when properly initialized.

Approaches with no shape-preserving alignment

These approaches focus on how to extract more discriminative features for the facial landmark detection task, avoiding the inclusion of constraints to explicitly enforce a valid face shape into their predictions. We organize these methods in two subgroups related to the use of holistic features (*e.g.*, deep learning approaches) and local parts based features (*e.g.*, voting regression approaches) respectively:

- **Regression-voting methods** individually compute the location of each landmark by casting votes from different regions around the face. Dantone *et al.* [26] proposed conditional Random Forest to learn different landmarks detectors depending on the head pose (see Section 2.2), which reduces the shapes variability. Yang *et al.* [127] introduced a sieve based on a filter that rejects votes from local patches that are not estimating the nose tip center properly. Both methods learn the direct mapping from the texture to the landmark coordinates efficiently without any initialization. Differently to previous methods, which explicitly impose a face shape model, these direct regression methods lack of a global 3D model to enforce face shape consistency. Thus, they are noticeably inaccurate in estimating the landmarks around the facial contour (see Fig. 4.6).

- **Deep learning methods** have gained popularity over the last years because of their state-of-the-art performance in the majority of computer vision problems. The large receptive field of deep CNNs and their discriminative features convey a high degree of robustness to face rotation, scale and both rigid and non-rigid deformation. Indeed, previous CSR methods based on ensembles of trees would lead to alignment failures depending on the face detection precision, while CNNs extract robust features more invariant to these possible detection errors, *e.g.*, in Fig. 1.2 the bounding box related to the person wearing sunglasses does not fit well to the face contour.

  Nowadays, deep learning models evolve to the concatenation of multiple regressors that progressively refine the estimation of previous stages, following the CSR scheme. Most literature propose to substitute previous boosted regression trees with CNNs, to achieve better accuracy in spite of incrementing the computational requirements. Depending on whether they return the position of the landmarks or their probability maps, these ensembles of CNNs can be organized into two groups:

  – *Coordinate regressors* produce a vector of 2D landmark coordinates as output. Sun *et al.* [100] were the pioneers to apply a three-stage CNN where the first level provides highly robust initial shape estimations, and the two remaining CNN regressors refine the initial prediction to achieve accurate results. Yu *et al.* [132] (DDN) designed a two-stage ensemble trainable end-to-end, where the

output of the first regressor is used to rectify the input face image before it is further passed to the second stage. Similarly, Kowalski *et al.* [58] (DAN) and Yang *et al.* [129] (SHN) introduced a cascade of CNNs where the face image is also rectified as input of the intermediate stages along with the resulting heatmaps produced using a VGG [98] and a Stacked Hourglass Network [81] respectively (*i.e.*, heatmaps were solely demanded as a means for transferring information between stages). As a result, [58, 129] regress the final prediction using the whole face image thanks to the use of these heatmaps, which transmit the information about the landmark location.

Contrarily, Trigeorgis *et al.* [107] (MDM) and Xiao *et al.* [123] (RAR) modelled the extraction of features along the stages of the cascade through a recurrent neural network. The former progressively designed such a coarse-to-fine scheme using patches around the current landmark location, while the latter first refines reliable landmarks that help to infer the location of occluded and challenging landmarks at the end. Feng *et al.* [36] (Wing) analyzed the relevance of classic L_2, L_1 loss functions to regress the landmark coordinates. They prefer to pay more attention to samples with smaller error, to achieve high accuracy in easy face images, using a novel loss function termed "wing loss".

In spite of CNNs impressive discriminative features, it still remains unclear how to deal with the initialization problem (*i.e.*, remove the rigid face deformation to make easier the work of the successive regressors) and how to implicitly keep a valid face shape in their predictions. On the one hand, Lv *et al.* [72] (TSR) and Liu *et al.* [66] (RDN) have designed some specific modules that iteratively regress a good initialization. Additionally, RDN is one of the few approaches that provide an error function that let us evaluate the quality of a prediction (*i.e.*, allowing to evaluate which initialization is best). On the other hand, Miao *et al.* [76] (DSRN) and Zhu *et al.* [143] (ODN) captured geometric relations of different facial components using a low-rank learning layer to recover missing features by means of geometric information and intrinsic correlations between landmarks.

Finally, it is worth mentioning the sophisticated CNN model of Bulat *et al.* [13] (Binary-CNN) that involves four stacked Hourglass networks [81] along with a ResNet-152 [49] to infer the landmark coordinates using as input the original face image and the intermediate heatmaps. As far as we know, this is the first method that attaches a coordinate regressor at the end of a heatmap regressor to generate their prediction.

– *Heatmap regressors* encode the probability of locating the landmarks of interest at certain positions in the face image as output. The final shape coordinates are frequently estimated by computing the maximum response of these heatmaps. Typically, these approaches have been proposed following an encoder-decoder architecture similar to U-Net [90] and Hourglass networks [81]. Both networks share the same encoder-decoder topology, but the latter includes convolutional layers within lateral connections.

From now on, we build on recent literature that utilized a cascade of U-Nets to learn better feature maps. Honari *et al.* [50] (RCN) were the pioneers that proposed an encoder-decoder to perform landmark detection. A key difference with U-Net is that RCN only applies one convolution before the pooling layers rather than two, and U-Net always doubles the amount of feature maps when

the image size is halved. Guo *et al.* [42] studied how to cascade multiple densely connected U-Nets including a novel network topology to obtain accurate results in the challenging faces. Analogously, Tang *et al.* [104] (DU-Net) also get good results using densely connected U-Nets that share features between stages, by quantizing model parameters for better computational efficiency. In addition, Deng *et al.* [28] (MHM) also presented a two-stage model based on Hourglass networks, where their first stage compute the rigid deformation, whereas their second regressor just focuses on the non-rigid deformation.

Other cascades of heatmap regressors, *e.g.*, Dong *et al.* [31] (SAN) investigated how to improve the data augmentation procedure to generate face images with different styles through a generative adversarial network. Wu *et al.* [117] (LAB) complemented a sequence of Hourglass networks [81] with a boundary heatmap module that provides shape information to preserve a valid face shape against occlusions. As we mentioned previously, it still remains unclear how to impose facial shape consistency on the estimated set of landmarks without using any explicit model. Kumar *et al.* [61] (PCD-CNN) introduced a two-stage network, where the shape constraint is imposed by a dendritic structure of landmarks. In addition, they incorporate a second stage based on a coordinate regressor, like GoogLeNet [98], to demonstrate that the alignment error with a heatmap regressor is always smaller than the one obtained with a coordinate regressor. In Section 3.5.6, we also prove that SHN, MHM, LAB, DU-Net and PCD-CNN (heatmap regressors) set the state-of-the-art.

As we noted previously, facial landmark detection is closely related to other facial analysis problems. Not only it is a common pre-processing step to further reduce the error of subsequent tasks [52] (*e.g.*, rectifying face images into a canonical position to reduce their appearance variability), but also it is necessary to train different regressors using only training faces with similar face shape (*i.e.*, learning separate models for each head pose as mentioned in Section 2.1). Recent work [141, 121, 35, 146, 119, 87, 60, 140, 86] demonstrates that learning simultaneously another facial correlated problem (*e.g.*, head pose, action units, face detection, facial expression) along with face alignment allows them to improve their performance. In Chapter 4 we discuss these multi-task approaches and propose a final solution based on a multi-task scheme that achieves state-of-the-art results on facial landmark detection.

To this end, our main proposals in this chapter are based on the concatenation of two regressors, which progressively refine the shape predictions minimizing the alignment error. On the one hand, we introduce a cascade of two U-Net networks that predicts the landmark coordinates using only the information from the heatmap associated to that landmark. In this case, we attempt to preserve the face shape by training using synthetic occlusions, and by randomly deleting intermediate heatmaps to force the model to learn the location of landmarks using the information of their neighbours. On the other hand, we are the first to introduce a hybrid scheme combining a single U-Net and a standard ERT method to implicitly enforce a face shape constraint. In this case, we also design a coarse-to-fine strategy to improve the capacity of the popular ERT scheme [55] giving more flexibility to the face parts.

3.2 Coarse-to-fine ERT regressor

Cascaded regression approaches have been playing a dominat role until very recently due to their good efficiency, high precision and their implicit face shape consistency [17, 55, 16, 89, 63, 124, 144]. These CSR methods [30] typically consist of a sequence of boosted regression trees whose objective is to progressively estimate both rigid and non-rigid face deformation required to perform face alignment in-the-wild. In this section, we enhance the standard Ensemble of Regression Trees (ERT) [55], by adding a coarse-to-fine scheme to give more flexibility to the facial parts and enable them to learn combinations of facial expressions not seen during training.

Let $\mathcal{S} = \{s_i\}_{i=1}^{N}$ be the set of N training face samples, where $s_i = (\mathtt{I}_i, \tilde{\mathbf{x}}_i, \tilde{\mathbf{v}}_i, \tilde{\mathbf{w}}_i, \mathbf{x}_i^0, \mathbf{v}_i^0)$. Each sample s_i, annotated with L landmarks, has its own training image, $\mathtt{I}_i$; ground truth shape, $\tilde{\mathbf{x}}_i \in \mathbb{R}^{L\times 2}$; ground truth visibility label, $\tilde{\mathbf{v}}_i \in \{0,1\}^L$; annotated landmark label, $\tilde{\mathbf{w}}_i \in \{0,1\}^L$; initial shape, $\mathbf{x}_i^0$; and initial visibilities, $\mathbf{v}_i^0$. We divide the ERT regression process into a maximum of T stages. We learn an ensemble of K consecutive decision trees, g, for each t-th stage, $\mathcal{C}_t(f_i) = \mathbf{x}_i^{t-1} + \sum_{k=1}^{K} g_k(f_i)$, where these $f_i = \phi(\mathtt{I}_i, \mathbf{x}_i^{t-1}, \tilde{\mathbf{w}}_i)$ are shape-indexed local features [128] that depend on the location of the landmarks $\mathbf{x}_i^{t-1}$ in image $\mathtt{I}_i$, and whether they are annotated or not, $\tilde{\mathbf{w}}_i$.

To train the ERT, we use the N training shapes in $\mathcal{S}$ to generate an augmented training set of samples, $\mathcal{S}_A$; and a validation set, $\mathcal{S}_V$; with cardinality $N_A = |\mathcal{S}_A|$ and $N_V = |\mathcal{S}_V|$ respectively, by changing their initial shape $\mathbf{x}_i^0$. The total number of available samples is $N_T = N_A + N_V$. Moreover, instead of using a fixed number of stages like [17, 55, 89, 63, 16], we stop training when the validation error stops improving. In this way the regressor has a variable number of stages.

Additionally, we incorporate the visibility label $\mathbf{v}$, along with the shape $\mathbf{x}$, to address occlusions, similar to [16], and the ground truth annotation labels $\mathbf{w}$, to handle partially labelled training data, like [55]. We progressively refine each current shape by estimating a shape and visibility increments $\mathcal{C}_t^{\mathbf{v}}(\phi(\mathtt{I}_i, \mathbf{x}_i^{t-1}, \tilde{\mathbf{w}}_i))$. We train $\mathcal{C}_t^{\mathbf{v}}$ to minimize the landmark position errors, but on each tree leaf we output both the mean shape and the average of all training shapes visibilities that belong to that leaf node. In Algorithm 4, we define the *NME* as the alignment error, while $\mathcal{A}_{t-1} = \{(\mathbf{x}_i^{t-1}, \mathbf{v}_i^{t-1})\}_{i=1}^{N_A}$ and $\mathcal{V}_{t-1} = \{(\mathbf{x}_i^{t-1}, \mathbf{v}_i^{t-1})\}_{i=1}^{N_V}$ are the set of all current shapes and corresponding visibility vectors for all training and validation data respectively, and P is the amount of facial parts, *i.e.*, one facial part with the whole shape or several parts to move each region independently (see Fig. 3.6).

Traditional stage wise learning is a greedy strategy able to fit training data really well, which can lead to overfitting. In Algorithm 4, we apply the coarse-to-fine refinement scheme (see Section 3.2.3) the first time the training error is smaller than the one we get in validation. Then, we reach the maximum number of stages, T, when the validation error continues decreasing during the whole training process. In case of overfitting, we introduce an early stopping criteria that reduces the amount of regression trees required, T^*. In Fig. 3.3 we show the alignment results of our proposal across the cascade stages, using the mean shape $\bar{\mathbf{x}}^0$ as initialization, which suffices to model this easy/semifrontal face. In Section 3.4 we include new robust features and a better rigid initialization able to cope with more challenging in-the-wild conditions, *e.g.*, faces with extreme poses, expressions, occlusions, and so forth.

Algorithm 4 Training a coarse-to-fine Ensemble of Regression Trees

Input: $\mathcal{S}$, T

// Generate an augmented training set of samples

$\mathcal{S}_A, \mathcal{S}_V$ = dataAugmentation($\mathcal{S}$)

for t=1 **to** T **do**

// Extract training ($\mathcal{F}_A$) and validation ($\mathcal{F}_V$) features

$\mathcal{F}_A \cup \mathcal{F}_V = \{f_i\}_{i=1}^{N_T} = \{\phi(\mathtt{I}_i, \mathbf{x}_i^{t-1}, \tilde{\mathbf{w}}_i)\}_{i=1}^{N_T}$

// Apply Algorithm 5 using training samples

$\mathcal{C}_t^{\mathbf{v}}$ = learnCoarseToFineRegressor($\mathcal{S}_A$, $\mathcal{F}_A$, $\mathcal{A}_{t-1}$, K, P)

// Update validation samples

$\mathcal{V}_t = \mathcal{V}_{t-1} + \{\mathcal{C}_t^{\mathbf{v}}(f_i)\}_{i=1}^{N_V}$

// Increase P when $NME(\{\mathbf{x}_i^t, \tilde{\mathbf{x}}_i\}_{i=1}^{N_A}) < NME(\{\mathbf{x}_i^t, \tilde{\mathbf{x}}_i\}_{i=1}^{N_V})$

// Compute validation error improvement

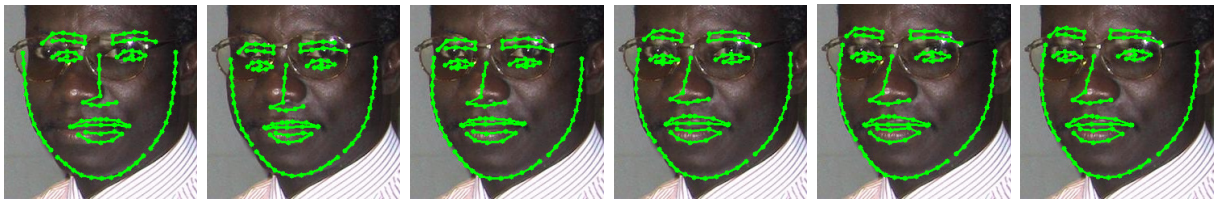
if $NME(\{\mathbf{x}_i^{t-1}, \tilde{\mathbf{x}}_i\}_{i=1}^{N_V}) - NME(\{\mathbf{x}_i^t, \tilde{\mathbf{x}}_i\}_{i=1}^{N_V}) > 0$ **then**

break

end if

end for

Output: $\{\mathcal{C}_t^{\mathbf{v}}\}_{t=1}^{T^*}$ // T^* is the last trained stage



t=0 t=5 t=20 t=50 t=70 t=T^*

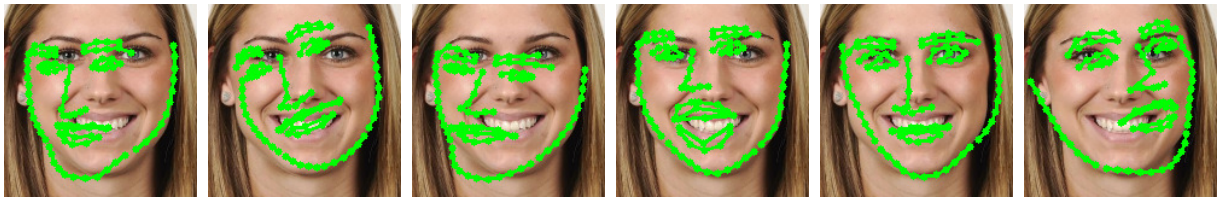
Figure 3.3: Evolution of landmark estimation iteratively updated at different stages of our coarse-to-fine ERT, using the mean shape as initialization. We employ the coarse-to-fine scheme from t=50 towards.

3.2.1 Initial shapes for regression

The selection of the starting point in the ERT is fundamental to reach a good solution. The simplest choice is the mean shape, the average of the ground truth training shapes, $\bar{\mathbf{x}}^0 = \sum_{i=1}^{N} \tilde{\mathbf{x}}_i / N$ (see t=0 in Fig. 3.3). An alternative strategy would be to run the ERT several times from different initializations (*e.g.*, discretize the head pose and compute an individual mean shape model for each orientation), taking an average shape as a result, similar to [16], but it is computationally expensive.

In our coarse-to-fine ERT scheme we use as initialization the average 2D mean shape computed using all faces from the training set. At this moment, we generate an additional source of variability during the data augmentation step (see Algorithm 4), and we provide an augmented training set of samples $\mathcal{S}_A$. The amount of initializations generated will be different for each database, since we train all models using the same number of samples N_A (*i.e.*, large data sets will require a lower amount of training initializations for each face image). We follow [55] using as initializations ground truth shapes $\mathbf{x}_i^0 = \tilde{\mathbf{x}}_j$ being $i \neq j$. It provides an additional source of variability for the training procedure (see Fig. 3.4).

(a) Random samples acquired from the WFLW training subset



(b) Data augmentation generated using above ground truth as initialization

Figure 3.4: Initializations $\mathbf{x}^0$ included during training in addition to the mean shape $\bar{\mathbf{x}}^0$.

To summarize, CSR methods are very sensitive to the starting point of the regression process, and using the mean shape $\bar{\mathbf{x}}^0$, we only achieve competitive results when the face is easy/semifrontal. With these augmented samples we alleviate this problem, but we do not solve it. In Section 3.4.1 we will improve these initializations with a novel procedure based on the robust features produced by a CNN.

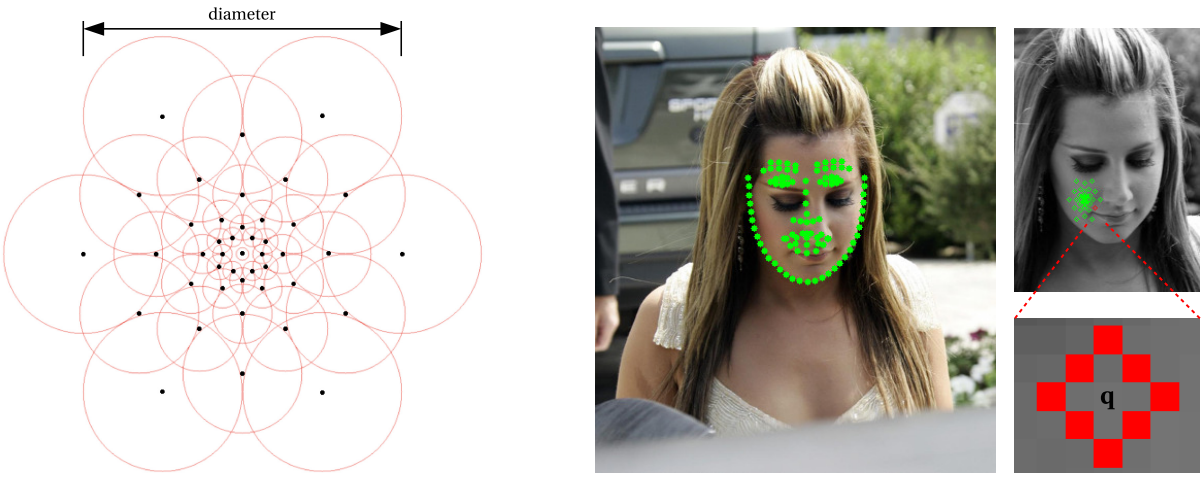
3.2.2 Feature extraction

The shape-indexed feature description is done only once at each stage in the CSR [30], to achieve real-time performance. In our coarse-to-fine ERT strategy, we abandoned the idea of using a standard feature descriptor (*e.g.*, SIFT [71], SURF [6]) to obtain accurate results (see Section 3.5.4), because they demand higher computational cost compared to simpler features such as plain pixel values differences [17, 16, 55, 89]. At this moment, we choose as features, $f_i = \mathtt{I}_i[\mathbf{q}_1] - \mathtt{I}_i[\mathbf{q}_2]$, computed as the difference between two pixel values $\mathbf{q}_1$ and $\mathbf{q}_2$ related to a random l-th landmark from the current shape, $\mathbf{x}_i^{t-1}$.

We follow [55, 63] allowing the training algorithm to select the most informative pair of pixels in each tree node from a FREAK pattern [2]. Additionally, we employ difference of Gaussian features similar to [63], where the smoothing scale of the sampling points in Fig. 3.5a is proportional to their distance to the center point. This means that we must load into memory 8 feature channels for each training image (*i.e.*, 1 gray-scale image and 7 Gaussian filtered images). We propose to crop the images using their labelled bounding boxes to reduce memory requirements in our implementation. These bounding boxes have been enlarged to support the extraction of features from pixels around the face contour.

Another contribution is the way we gradually reduce the FREAK pattern diameter [2] over the CSR stages, to model larger errors at the earlier stages and perform fine-grained landmark location at the end.

In Fig. 3.5b we have randomly chosen a landmark, l, (left corner of the mouth) to show the FREAK pattern projection between the mean shape, $\bar{\mathbf{x}}^0$, and the current shape, $\mathbf{x}_i^{t-1}$. Note that a wrong projection leads to irrelevant shape-indexed feature computation, which in turn leads to even more irrelevant extracted features. In [16, 128], the authors introduced a simple scheme to reference features by linear interpolation between nearest landmarks.



(a) FREAK pattern [2]

(b) Pixel value $\mathbf{q}$ in the gray-scale image

Figure 3.5: Extraction of features around a random landmark l (left corner of the mouth) using a FREAK pattern projected according to the current shape $\mathbf{x}_i^{t-1}$.

In spite of their aforementioned advantages in terms of efficiency and accuracy, these handcrafted local features extracted from the gray-scale channels are not discriminative enough to deal with faces under in-the-wild conditions. Thus, we propose in Section 3.4.2 new feature channels extracted from a CNN.

3.2.3 Training the coarse-to-fine regressor

To train the t-th stage regressor, $\mathcal{C}_t^{\mathbf{v}}$, we fit a sequence of K regression trees g. The goal is to train a combination of weak learners that greedily minimizes the following regression loss function:

$$\mathcal{L}_t(\mathcal{S}_A, \mathcal{F}_A, \mathcal{A}_{t-1}) = \sum_{i=1}^{N_A} ||\tilde{\mathbf{w}}_i \odot (\tilde{\mathbf{x}}_i - \mathbf{x}_i^{t-1} - \sum_{k=1}^{K} g_k(f_i))||^2 \tag{3.1}$$

where $\odot$ is the Hadamard product required to ignore the prediction errors from unlabelled landmarks. There are various ways of minimizing Eq. 3.1, *e.g.*, Gradient Boosting [55], Gaussian Processes [63], etc. According to [63], the Gaussian Processes are less prone to overfitting. However, in our proposal we have followed an efficient greedy strategy based on the Gradient Boosting scheme, due to the computational efficiency obtained during the minimization process described in Eq. 3.3.

A crucial problem when training a global face landmark regressor is the lack of examples showing all possible combinations of face parts deformations. Hence, these regressors quickly overfit and generalize poorly to combinations of part deformations not present in the training set. To address this problem we introduce the coarse-to-fine ERT framework. The goal is to be able to cope with combinations of face part deformations not seen during training. A single monolithic regressor is not able to estimate all these local deformations (see differences between monolithic and coarse-to-fine regressors in Fig. 3.16).

In Algorithm 5 we show the training of individual face parts regressors (each one with a different set of the landmarks) to build a coarse-to-fine regressor. Our implementation consists of only two stages. The coarse stage incorporates one part, $P = 1$, that involves all landmarks, whereas the fine stage has ten parts, $P = 10$, left/right eyebrow, left/right eye,

Algorithm 5 Training P parts regressors

Input: $\mathcal{S}_A, \mathcal{F}_A, \mathcal{A}_{t-1}, \nu, K, P$
  for k=1 **to** K **do**
    for p=1 **to** P **do**
      // Compute prediction error (residual $\mathbf{r}_i^k$):
      // $\odot$ is the Hadamard product
      // (p) selects elements of vectors in that part
      $\{\mathbf{r}_i^k(p) = \tilde{\mathbf{w}}_i(p) \odot (\tilde{\mathbf{x}}_i(p) - \mathbf{x}_i^{k-1}(p))\}_{i=1}^{N_A}$

      $g_k^p = \text{fitRegressionTree}(\{\mathbf{r}_i^k(p)\}_{i=1}^{N_A}, \mathcal{F}_A(p))$

      // Update training samples with the regression tree estimation
      // ν, shrinkage factor to scale each tree contribution
      $\mathcal{A}_k(p) = \mathcal{A}_{k-1}(p) + \nu \cdot \{g_k^p(f_i(p))\}_{i=1}^{N_A}$
    end for
  end for
Output: $\{\mathcal{C}^p\}_{p=1}^{P}$, being $\mathcal{C}^p = \{g_k^p\}_{k=1}^{K}$

nose, top/bottom mouth, left/right earlobe and chin. Moreover, we reduce proportionally the amount of regression trees according to the number of parts (*i.e.*, we use K regression trees in the coarse level and $K/10$ in the fine stage).

In Fig. 3.6 we display all parts of interest connected by coloured lines. Note that each face part is comprised of a different number of landmarks depending on the database and their annotated landmarks (see Section 3.5.1).

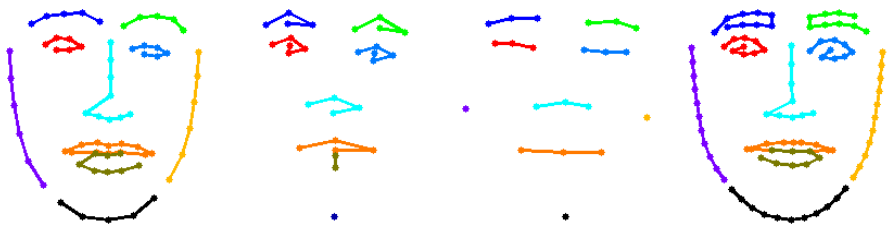


Figure 3.6: The $P = 10$ face parts of our finer stage each represented in a different colour. We show the mean shape using 300W, COFW, AFLW and WFLW data sets respectively.

3.2.4 Fit a regression tree

At this point, we examine different loss functions for regression and characterize them in terms of their robustness to outliers. The objective function, $\mathcal{E}_k$, introduced in Eq. 3.2 minimizes the difference between predicted and ground truth shapes, also referred to as residual, $\mathbf{r}^k$. It is worth noticing that $\mathbf{r}^k$ can be optimized using the whole face shape, composed of L landmarks [17, 55], or using only one landmark [63]. In Table 3.6 we also evaluate both strategies, reaching the conclusion that training each decision tree with a single landmark reduces the risk of overfitting.

The squared error loss (see Eq. 3.1) places the emphasis of the objective function $\mathcal{E}_k$

on samples with large absolute residuals,

$$\mathcal{E}_k = \sum_{i=1}^{N_A} ||\mathbf{r}_i^k||^2 = \sum_{i=1}^{N_A} ||\tilde{\mathbf{w}}_i \odot (\tilde{\mathbf{x}}_i - \mathbf{x}_i^{k-1})||^2. \tag{3.2}$$

Other robust criteria, such as the absolute loss, L_1, perform much better in situations where the error distribution is not normal, being less vulnerable to possible outliers. For squared error loss the gradient is just the ordinary residual, $\mathbf{r}_i^k$, whereas with absolute error loss, the gradient is the sign of the residual, $sign(\mathbf{r}_i^k)$. In our coarse-to-fine ERT, we have selected the squared error loss due to their higher efficiency based on the differentiable loss optimization from Eq. 3.3.

We learn each regression tree by recursively splitting the training set into the left (le) and right (ri) child nodes. To train a regression tree node, we randomly generate a set of candidate split functions, each of them involving four parameters $\theta = (\tau, \mathbf{q}_1, \mathbf{q}_2, l)$, where $\mathbf{q}_1$ and $\mathbf{q}_2$ are pixels coordinates around the l-th landmark in $\mathbf{x}_i^{k-1}$. Finally, we compute the split function thresholding the feature value, $f_i(\mathbf{q}_1, \mathbf{q}_2) > \tau$.

Given $\mathcal{N} \subset \mathcal{S}_A$ the set of training samples at a node, fitting a tree node for the k-th tree, consists of finding the parameter θ that minimizes $E_k(\mathcal{N}, \theta)$

$$\begin{aligned}
\arg\min_{\theta} E_k(\mathcal{N}, \theta) = \arg\min_{\theta} \sum_{b\in\{le,ri\}} \sum_{s\in\mathcal{N}_{\theta,b}} ||\mathbf{r}_s^k - \boldsymbol{\mu}_{\theta,b}||^2 = \\
\arg\min_{\theta} \sum_{b\in\{le,ri\}} \sum_{s\in\mathcal{N}_{\theta,b}} (\mathbf{r}_s^k - \boldsymbol{\mu}_{\theta,b})^T(\mathbf{r}_s^k - \boldsymbol{\mu}_{\theta,b}) = \\
\arg\min_{\theta} \sum_{b\in\{le,ri\}} \sum_{s\in\mathcal{N}_{\theta,b}} ((\mathbf{r}_s^k)^T\mathbf{r}_s^k - 2\boldsymbol{\mu}_{\theta,b}^T\mathbf{r}_s^k + \boldsymbol{\mu}_{\theta,b}^T\boldsymbol{\mu}_{\theta,b}) = \\
\arg\min_{\theta} \sum_{b\in\{le,ri\}} (-2\boldsymbol{\mu}_{\theta,b}^T(\sum_{s\in\mathcal{N}_{\theta,b}} \mathbf{r}_s^k) + \boldsymbol{\mu}_{\theta,b}^T\boldsymbol{\mu}_{\theta,b}) = \\
\arg\max_{\theta} \sum_{b\in\{le,ri\}} |\mathcal{N}_{\theta,b}| \cdot \boldsymbol{\mu}_{\theta,b}^T\boldsymbol{\mu}_{\theta,b}
\end{aligned} \tag{3.3}$$

where $\mathcal{N}_{\theta,le}$ and $\mathcal{N}_{\theta,ri}$ are, respectively, the samples sent to the left and right child nodes due to the decision induced by θ. The mean residual $\boldsymbol{\mu}_{\theta,b}$ for a candidate split function and a subset of training data is given by,

$$\boldsymbol{\mu}_{\theta,b} = \frac{1}{|\mathcal{N}_{\theta,b}|} \sum_{s\in\mathcal{N}_{\theta,b}} \mathbf{r}_s^k. \tag{3.4}$$

Once we know the optimal split, each leaf node stores the mean residual, $\boldsymbol{\mu}_{\theta,b}$, as the output of the regression for any example reaching that leaf node. In addition, we also return the mean visibility of the samples reaching the tree leaf.

In summary, we have introduced a novel design to perform face alignment efficiently following a coarse-to-fine ERT to estimate both rigid and non-rigid deformation progressively, whereas we enforce implicitly a face shape constraint. It is based on a simple mean shape initialization and the difference of two pixel values acquired from the gray-scale channel. We have improve this algorithm in Section 3.4 through the extraction of robust feature maps from an encoder-decoder CNN baseline. In the following section, we analyze the performance of an ensemble of regressors, where each model in based on a heatmap regressor instead of a simple regression tree.

3.3 Cascade of CNN regressors

The mainstream solution based on CNNs [132, 58, 129, 13, 42, 104, 36, 117] focuses on the concatenation of two or more Convolutional Neural Network (CNN) regressors. In this section, we investigate the use of a sequence of CNNs to estimate the facial landmark location [70]. Hence, we first introduce a Cascade of Heatmap Regressors (CHR), which consists of two encoder-decoder CNNs, *e.g.*, U-Net [90], RCN [50], Hourglass [81]. These popular CNNs architectures have obtained excellent performance for problems that require both local and global features at different scales. Both encoder-decoder models are also symmetric and provide skipping connections across levels with the same resolution. In our proposal we present the RCN [50] baseline, where the levels represent the different image resolutions that we handle in our network, *i.e.*, an encoder-decoder that progressively halves the image resolution from 32×32 to 1×1 pixels contains 6 levels.

Additionally, in [110] we extend our proposal by adding an extra regression module that estimates the most likely landmark coordinates from the probability maps generated in the CHR. We call this framework: Cascaded Heatmap Regression into 2D Coordinates (CHR2C) (see Fig. 3.7). Compared to [13], our method does not require any sophisticated coordinate regressor attached at the end of the CHR to compute these coordinates from the heatmaps. Differently to previous heatmap regressors that estimate the maximum response of each heatmap as output, we propose a new layer that learns the position of each landmark from its probability map.

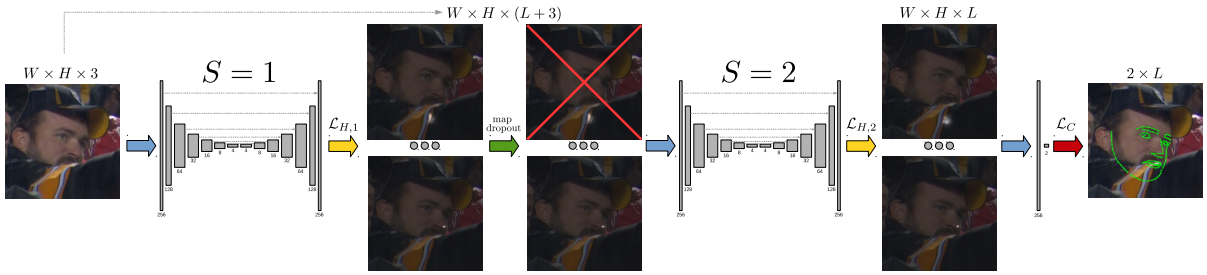


Figure 3.7: CHR2C framework architecture diagram. Each stage S is an encoder-decoder heatmap regressor. Yellow arrows represent the softmax losses that produce the heatmaps for the landmarks. Between each stage we introduce a *map dropout* layer which deletes a fraction f of the heatmaps. Finally, the red arrow represents the regression from heatmaps to 2D coordinates.

The key idea behind our proposal is to employ a sequence of CNNs that incrementally refines the location of the set of landmarks. The input to each encoder-decoder network is the original input face image and the set of heatmaps produced by the previous stage of the cascade. We also train the two stages of our framework progressively. We start with the first heatmap regressor ($S = 1$), which focuses on learning geometric transformations to roughly estimate the location of each visible landmark. Then, fine-tuning the learned weights as initialization, we cascade the second heatmap regressor ($S = 2$) and train it including synthetic occlusions. Between the two encoder-decoder CNNs, we also introduce a *map dropout* layer that deletes a fraction f of the heatmaps (red-crossed map in Fig. 3.7) and an auxiliary loss to preserve these heatmaps during the end-to-end refinement. In this way, the second stage must learn the relative location of the landmarks, since f of them must be predicted from the position of its neighbours. Finally, we include a fully connected layer with shared weights among all heatmaps that accurately regresses the

landmark coordinates (see Table 3.4). This novel module replaces the computation of the maximum response of each heatmap, so-called `argmax`.

3.3.1 Heatmap regression model

We propose an architecture termed Cascade of Heatmap Regressors (CHR) [70], based on two stacked U-Net [90] models each consisting of 7 levels, reducing the spatial extent of the input face image from 256×256 to 4×4 pixels. Whenever this spatial resolution is halved, we double the number of feature maps, from 64 and up to 256. In any standard encoder-decoder, finer and deeper levels pass information to the coarser ones allowing the network to combine information at different levels of abstraction and scales. We show in Tables 3.1 and 3.2 both encoder and decoder modules. In Fig. 3.8 we also display the diagram of the CNN architecture. The output of each stage is a heatmap for each of the L landmarks. Each heatmap represents the probability distribution of the actual location of one landmark within the input face image. We also include BatchNormalization and ReLu after each convolutional layer, but for the sake of simplicity we do not display them in Tables 3.1 and 3.2.

Name	Layer	Output	Connected to
`input`	InputLayer	(256, 256, 3)	
`conv_9_1`	Conv2D (1x1)	(256, 256, 64)	`input`
`conv_9_2`	Conv2D (1x1)	(256, 256, 64)	`conv_9_1`
`conv_8_1`	Conv2D (2x2)	(128, 128, 128)	`conv_9_2`
`conv_8_2`	Conv2D (1x1)	(128, 128, 128)	`conv_8_1`
`conv_7_1`	Conv2D (2x2)	(64, 64, 256)	`conv_8_2`
`conv_7_2`	Conv2D (1x1)	(64, 64, 256)	`conv_7_1`
...	...	...	...
`conv_4_1`	Conv2D (2x2)	(8, 8, 256)	`conv_5_2`
`conv_4_2`	Conv2D (1x1)	(8, 8, 256)	`conv_4_1`
`conv_3_1`	Conv2D (2x2)	(4, 4, 256)	`conv_4_2`
`conv_3_2`	Conv2D (1x1)	(4, 4, 256)	`conv_3_1`

Table 3.1: CNN encoder architecture.

Name	Layer	Output	Connected to
`conv_3_3`	Conv2DTrans (2x2)	(8, 8, 256)	`conv_3_2`
`concat_4_3`	Concatenate	(8, 8, 512)	`conv_4_2, conv_3_3`
`conv_4_4`	Conv2D (1x1)	(8, 8, 256)	`concat_4_3`
`conv_4_5`	Conv2D (1x1)	(8, 8, 256)	`conv_4_4`
`conv_4_6`	Conv2DTrans (2x2)	(16, 16, 256)	`conv_4_5`
...	...	...	...
`concat_7_3`	Concatenate	(64, 64, 512)	`conv_7_2, conv_6_6`
`conv_7_4`	Conv2D (1x1)	(64, 64, 256)	`concat_7_3`
`conv_7_5`	Conv2D (1x1)	(64, 64, 256)	`conv_7_4`
`conv_7_6`	Conv2DTrans (2x2)	(128, 128, 128)	`conv_7_5`
`concat_8_3`	Concatenate	(128, 128, 256)	`conv_8_2, conv_7_6`
`conv_8_4`	Conv2D (1x1)	(128, 128, 128)	`concat_8_3`
`conv_8_5`	Conv2D (1x1)	(128, 128, 128)	`conv_8_4`
`conv_8_6`	Conv2DTrans (2x2)	(256, 256, 64)	`conv_8_5`
`concat_9_3`	Concatenate	(256, 256, 128)	`conv_9_2, conv_8_6`
`conv_9_4`	Conv2D (1x1)	(256, 256, 64)	`concat_9_3`
`conv_9_5`	Conv2D (1x1)	(256, 256, 64)	`conv_9_4`

Table 3.2: CNN decoder architecture.

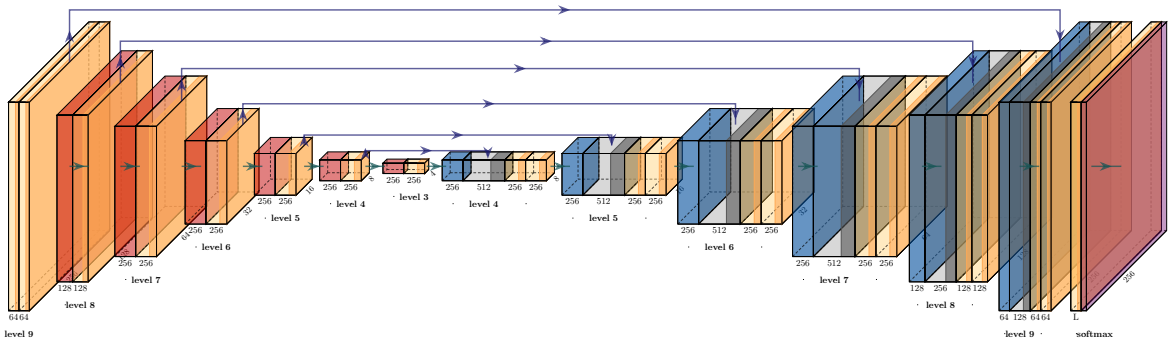


Figure 3.8: Architecture of each CHR2C encoder-decoder module. It consists of 7 levels where we progressively reduce the resolution of the input face image from 256×256 to 4×4 pixels. The red and blue layers represent convolutional and transposed convolutional layers with stride 2 and kernel size 2×2, to increase and reduce the resolution respectively.

Our approach involves a loss function, $\mathcal{L}_H$, that we describe in Eq. 3.5. It evaluates the goodness of fit for heatmaps between cascade components. We propose a loss function that is able to handle missing landmarks. This enables us to augment our data with large rigid transformations, treating landmarks falling outside of the bounding box as missing. It also allows us to train the model with data sets having missing landmarks.

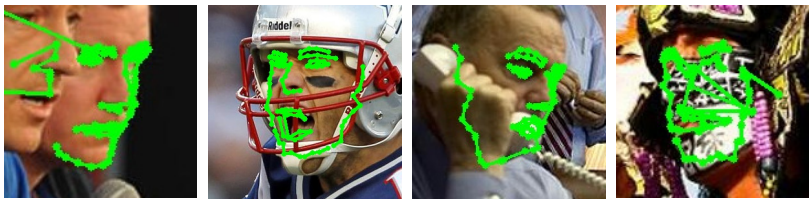
We use one-hot encoding for representing the ground truth of each heatmap. Thus, in the ground truth heatmap, $\tilde{\mathbf{h}}_i^l$, we set to 1 the pixel with the l-th landmark location. We employ a softmax to get a sum to one output in the l-th heatmap, $\sum_i^P \mathbf{h}_i^l = 1$ and adopt the cross-entropy loss for learning the heatmaps,

$$\mathcal{L}_H = \sum_{i=1}^{N} \left(\sum_{l=1}^{L} \left(\frac{\tilde{\mathbf{w}}_i^l}{||\tilde{\mathbf{w}}_i||_1} \sum_{p=1}^{P} (-\tilde{\mathbf{h}}_i^l(p) \cdot \log(\mathbf{h}_i^l(p)) \right) \right), \tag{3.5}$$

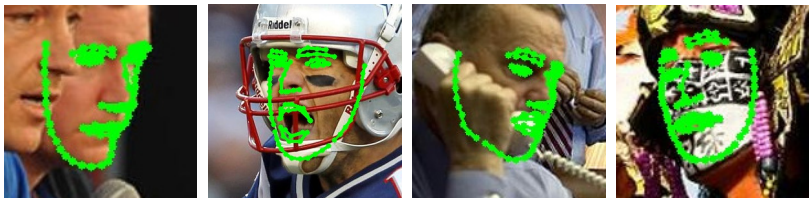
where N is the number of training images, L the number of landmarks and P the number of pixels to evaluate, *i.e.*, 256×256 for heatmaps. To handle unlabelled landmarks we include $\tilde{\mathbf{w}}^l$, the per landmark labelled mask indicator variable ($\tilde{\mathbf{w}}_i^l = 1$ when a landmark is annotated and $\tilde{\mathbf{w}}_i^l = 0$ otherwise). This loss also enables data augmentation with large rotations, translations and scalings, labelling landmarks falling outside of the bounding box as missing ($\tilde{\mathbf{w}}_i^l = 0$).

Similar to [81], we introduce a heatmap loss head, $\mathcal{L}_H$, between each encoder-decoder stage to improve the learning convergence and encourage the intermediate feature maps produced by the decoder to be actual heatmaps.

We set the number of stages to 2 since more stages produce a marginal improvement in accuracy and a marked increase in the computational cost. In Fig. 3.9 we show some representative results after the first and second stages. Note that $S = 1$ predictions were sensitive to occlusions and did not follow the shape of a face. In fact, it is readily noticeable that occluded landmarks are usually located far from their neighbours. Differently, $S = 2$ shows the importance of the synthetic random occlusions (see Fig. 3.14) and a *map dropout* layer between stages to enforce the regressor to learn a valid face shape.



(a) CHR ($S = 1$)



(b) CHR ($S = 2$)

Figure 3.9: First and second rows show results obtained using `argmax` over the heatmaps from first ($S = 1$) and second ($S = 2$) stage of our CHR respectively.

3.3.2 Coordinate regression model

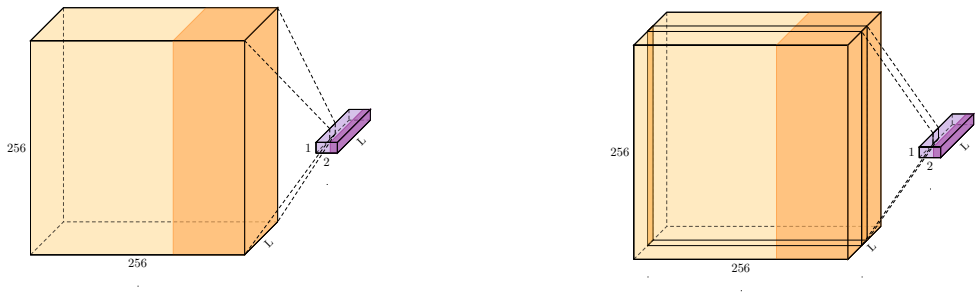
We also develop a simple and effective way of learning how to estimate the coordinates of the corresponding landmarks from the heatmaps. We refine and extend previous CHR [70] by adding a fully connected layer with shared parameters among all heatmaps, to regress the $L \times 2$ landmark coordinates. We show in Tables 3.3 and 3.4 the classic fully connected layer and our own design respectively. In Fig. 3.10 we also plot the diagram of these two modules that convert from heatmaps to coordinates. Fig. 3.10a represents the classic fully connected layer where we flatten all dimensions before connecting them to the output, whereas Fig. 3.10b represents a layer where only one heatmap is connected to each coordinate.

Name	Layer	Output	Connected to
`lnd`	InputLayer	(256, 256, L)	
`reshape_lnd`	Reshape	(L×65536)	`lnd`
`coord`	Dense	(L×2)	`reshape_lnd`

Table 3.3: Classic fully connected layer.

Name	Layer	Output	Connected to
`lnd`	InputLayer	(256, 256, L)	
`reshape_lnd`	Reshape	(65536, L)	`lnd`
`permute_lnd`	Permute	(L, 65536)	`reshape_lnd`
`coord`	Dense	(L, 2)	`permute_lnd`

Table 3.4: Heatmaps to coordinates layer.



(a) Different weights for all heatmaps

(b) Shared weights among heatmaps

Figure 3.10: Representation of the receptive field of the standard fully connected layer vs our proposal layer sharing weights among heatmaps.

As a result, our module requires the same amount of parameters (65536×2) independently of the number of landmarks, however in the classic procedure [58, 36] the amount of parameters would be huge (L×65536×2). It represents a parameters reduction of 96.55% in case of $L = 29$, and 98.52% in case of $L = 68$, making it computationally faster and less prone to overfitting.

Our full CHR2C approach uses an euclidean loss, $\mathcal{L}_C$, to evaluate the accuracy of the estimated landmark coordinates at the output of the last layer,

$$\mathcal{L}_C = \sum_{i=1}^{N} \left(\sum_{l=1}^{L} \left(\frac{\tilde{\mathbf{w}}_i^l}{||\tilde{\mathbf{w}}_i||_1} \cdot ||\tilde{\mathbf{x}}_i^l - \mathbf{x}_i^l|| \right) \right), \tag{3.6}$$

where $\mathbf{x}_i^l$ and $\tilde{\mathbf{x}}_i^l$ represent the l-th landmark predicted and the ground truth coordinates respectively for the i-th training image.

We introduce the following loss function to train the full CHR2C framework,

$$\mathcal{L} = \mathcal{L}_{H,1} + \mathcal{L}_{H,2} + \alpha \cdot \mathcal{L}_C, \tag{3.7}$$

where $\mathcal{L}_{H,s}$ denotes the heatmap loss, $\mathcal{L}_H$, at the output of the s-th encoder-decoder, and α is the weighting parameter balancing the contribution of the euclidean loss compared to softmax losses. Then, we fine-tune CHR2C using the weights of CHR, and train it end-to-end minimizing Eq. 3.7.

In summary, we have proposed a cascade of regressors formed by two encoder-decoder CNNs, whose goal is to alleviate the main problems of our coarse-to-fine ERT, *i.e.*, the initialization and the need of very informative low-level features to perform face alignment robust to the challenging in-the-wild conditions (see Section 3.2). Therefore, we follow the standard approach based on a sequence of CNNs, to achieve robust feature channels able to detect the location of landmarks, whereas we design a novel training procedure based on synthetic occlusions and spatial dropout, to let our second regressor learn how to enforce a valid face shape in our predictions. In the following section, we analyze the performance of another two-stage framework, where we substitute the second CNN with our coarse-to-fine ERT algorithm that implicitly preserves the face shape.

3.4 Hybrid CNN+ERT approach

In this section, we present a hybrid approach that inherits the best properties of both coarse-to-fine ERT (Section 3.2) and stacked CNNs (Section 3.3) techniques. As far as we know, this is the first time that an algorithm combines deep learning and classic boosted decision trees in a single approach for facial landmark detection. This research has been published in [112, 113].

Thus, we introduce 3DDE (3D Deeply-initialized coarse-to-fine Ensemble), a robust and efficient facial landmarks detector that consists of two main stages: CNN-based rigid face pose computation, and ERT-based non-rigid face deformation estimation, shown in Fig. 3.11.

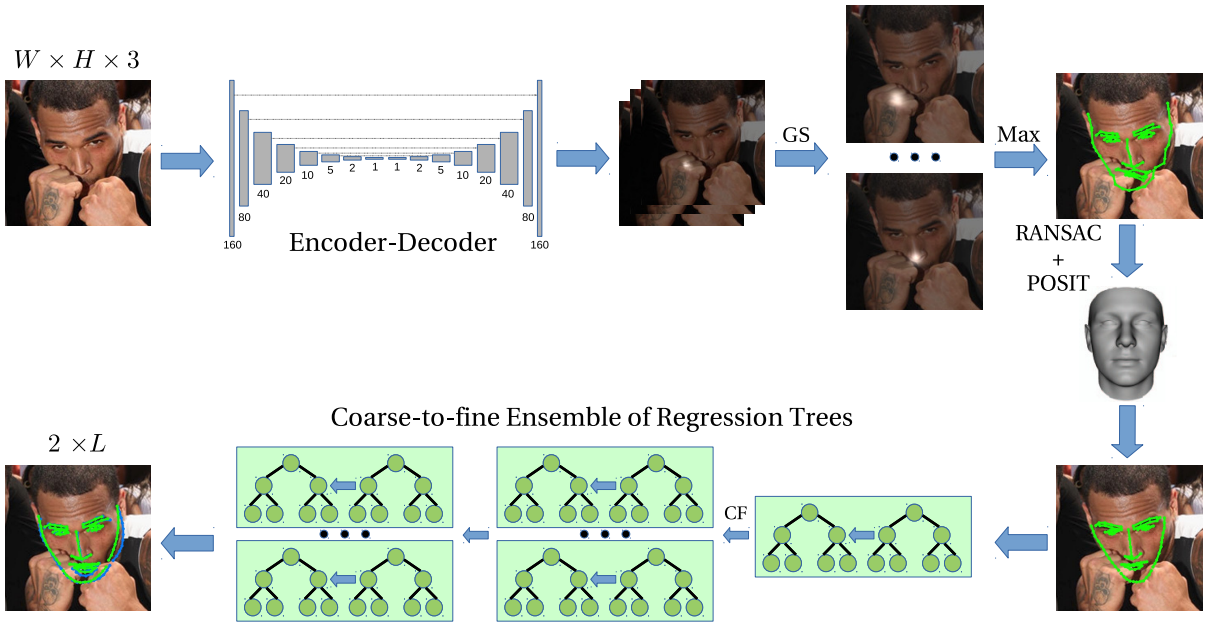


Figure 3.11: 3DDE hybrid framework architecture [113]. GS, Max and RANSAC+POSIT represent the Gaussian filter, the maximum of each probability map and the robust 3D pose estimation respectively.

In [112], we introduce a preliminary version of 3DDE, termed DCFE (Deeply-initialized

Coarse-to-Fine Ensemble), where a coarse-to-fine ERT is initialized by robustly fitting a 3D face model to the heatmaps produced by a CNN. With this initialization we tackle one of the main drawbacks of ERT, namely, the difficulty in initializing the regressor, $\mathbf{x}^0$, in the presence of occlusions and large face rotations. In [113], we refine and extend this scheme, through a RANSAC-like procedure [38] that increases its robustness in the presence of occlusions, at the expense of the increase in computational time.

Our proposal tries to leverage on the best properties of CNNs, 3D and ERT models. Using a CNN-based initialization we inherit the robustness of deep learning models. Like the simple 3D approaches, we fit a rigid 3D face model to initialize the regressor and estimate the initial face orientation to address self-occlusions and ambiguities. Then, we use an ERT to implicitly enforce a prior face shape on the solution, addressing the shortcomings of stacked CNNs when occlusions and ambiguous face configurations are present. Its coarse-to-fine structure tackles the combinatorial explosion of parts deformation, which is also a key limitation of approaches using shape constraints.

3.4.1 Rigid pose computation

As we mentioned in Section 3.2, ERT-based regressors require a good initialization to converge. At this point, we propose the use of previous heatmap regressors [90, 50, 81] to generate plausible shape initialization candidates. We define an encoder-decoder with a single stage ($S = 1$) following Section 3.3, with a loss function, $\mathcal{L}_H$, which also handles missing landmarks (see Eq. 3.5). We train this CNN to obtain a set of heatmaps, $\mathcal{P}(\mathtt{I})$, that model the position of each landmark in the input image. Then, we also smooth each heatmap using a Gaussian filter to reduce the noise. The mode of each smoothed heatmap determines our initial landmark positions. We note in Fig. 3.11 that these predictions are quite sensitive to partial occlusions, and may not be a valid face shape.

To start the ERT with a plausible face, we compute the initial shape by fitting a rigid 3D head model to the estimated 2D landmark locations. To this end, we use the POSIT algorithm [27] within a robust strategy. Unlike [112], here we use a set of the distinct landmarks (see Fig. 3.1) to establish the correspondences between the CNN predictions and the mean 3D face shape model. This avoids problems related to ambiguous landmarks around the jaw that do not always correspond to the same 3D points and produce wrong initializations, mainly in profile faces. Moreover, we have also implemented a RANSAC-like procedure [38], that runs POSIT several times with subsets of correspondences, to get a robust estimation (see Algorithm 6).

Let $\mathbf{X} \in \mathbb{R}^{L\times 3}$ be the 3D coordinates of the L landmarks on the 3D face model, $\mathbf{x} \in \mathbb{R}^{L\times 2}$ their 2D projections onto the image plane and $\mathbf{v} \in \{0,1\}^L$ their visibilities. We produce subsets of correspondences $(\mathbf{x}_s, \mathbf{X}_s)$ from the distinct landmarks shown in Fig. 3.1. We also infer the rigid pose $(\mathtt{R}, \mathbf{t})$ with POSIT and evaluate the goodness of each estimation as the sum of landmarks probabilities,

$$p(\mathbf{x}_z) = \sum_{l=1}^{L} \mathcal{P}^l(\mathtt{I})[\mathbf{x}_z^l], \tag{3.8}$$

where $\mathbf{x}_z^l$ are the 2D coordinates of the l-th landmark and $\mathcal{P}^l(\mathtt{I})$ is the probability map for landmark l produced by the CNN. Finally, we select the rigid transformation $(\mathtt{R}, \mathbf{t})$ with highest $p(\mathbf{x}_z)$. As a result, we project the 3D model onto the image using the most likely estimated rigid transformation. This provides the ERT with a rough estimation of

Algorithm 6 Initialization algorithm (d_0)

Input: $\mathcal{P}(\mathtt{I})$, $\mathbf{X}$
  // Select coordinates of maximum probability
  $\{\mathbf{x}^l = \arg\max(\mathcal{P}^l(\mathtt{I}))\}_{l=1}^{L}$
  $p^* = 0$
  for z=1 **to** Z **do**
    // Select subset from distinct landmarks
    $\mathbf{x}_s, \mathbf{X}_s$ = chooseLandmarksSubset($\mathbf{x}$, $\mathbf{X}$)

    // Compute projection matrix between $\mathbf{x}_s, \mathbf{X}_s$
    $\mathtt{R}, \mathbf{t}$ = POSIT($\mathbf{x}_s$, $\mathbf{X}_s$)

    // Project 3D face model using previous matrix
    $\mathbf{x}_z, \mathbf{v}_z$ = projectPoints($\mathbf{X}$, $\mathtt{R}$, $\mathbf{t}$)

    // Evaluate the goodness of the initialization
    $p(\mathbf{x}_z) = \sum_{l=1}^{L} \mathcal{P}^l(\mathtt{I})[\mathbf{x}_z^l]$
    if $p(\mathbf{x}_z) > p^*$ **then**
      $p^* = p(\mathbf{x}_z)$, $\mathtt{R}^* = \mathtt{R}$, $\mathbf{t}^* = \mathbf{t}$
    end if
  end for
  $\mathbf{x}^0, \mathbf{v}^0$ = projectPoints($\mathbf{X}$, $\mathtt{R}^*$, $\mathbf{t}^*$)
Output: $\mathbf{x}^0, \mathbf{v}^0$

the scale, translation and 3D pose of the target face, and the visibility prediction of the self-occluded parts of the face.

Furthermore, we provide an error function described in Eq. 3.8 that lets us evaluate the quality of a candidate shape for initialization.

Let $\mathbf{x}^0 = d_0(\mathcal{P}(\mathtt{I}), \mathtt{X})$ be the initial shape, the output of the initialization function after processing the input image $\mathtt{I}$. With our initialization we enforce two key requirements for the convergence of the ERT. First, that $\mathbf{x}^0$ lies on the face with an approximately correct 3D face pose. Second, that $\mathbf{x}^0$ is a valid face shape, which enforces that the predictions in the next step of the algorithm will also be valid face shapes [17].

3DDE computes the initialization for each sample using the 3D projections produced by d_0 (see Algorithm 6), instead of a simple 2D mean shape as in Section 3.2.1. As a result, these initial shapes provide a robust pose and a valid shape under in-the-wild conditions. Fig. 3.12 shows the worst initialization achieved by our proposal (*i.e.*, challenging faces with highest alignment error), and as we observe, the results are always quite accurate. Henceforth, our coarse-to-fine ERT estimates only the non-rigid deformation component of the faces.

In our coarse-to-fine ERT scheme, we provide an additional source of variability during the data augmentation procedure, and generate an augmented training set of samples $\mathcal{S}_A$ (see Algorithm 4). To this end, we add random noise to the yaw, pitch and roll angles from the rotation matrix $\mathtt{R}^*$ estimated with d_0, to generate new training initializations.

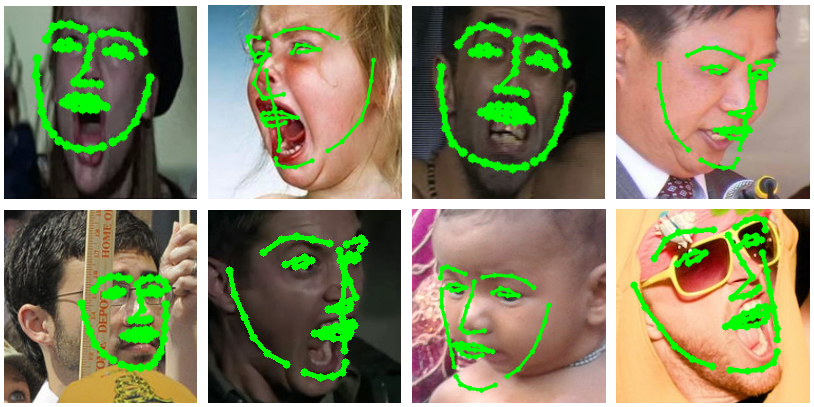


Figure 3.12: The 8 worst initial shapes produced by Algorithm 6 in 300W private.

3.4.2 ERT-based non-rigid shape estimation

In our 3DDE implementation we employ shape-indexed features [128], $\phi(\mathcal{P}(\mathtt{I}), \mathbf{x}^{t-1}, \tilde{\mathbf{w}})$ using the heatmaps as features for the decision trees instead of gray-scale pixel values as in Section 3.2.2. We utilize the probability maps $\mathcal{P}(\mathtt{I})$ to extract features for the cascade. To this end, we select a landmark l and its associated probability map $\mathcal{P}^l(\mathtt{I})$. We compute the feature as the difference between two pixels values in $\mathcal{P}^l(\mathtt{I})$ from a FREAK pattern [2] around l.

In comparison with the coarse-to-fine ERT initialized from the mean shape and using gray-scale channel features, 3DDE requires fewer decision trees because we only have to estimate the non-rigid face deformation, since the 3D rigid component has already been estimated in the previous stage.

Finally, we have also modified Algorithm 4 to verify whether the validation error is not improving, $NME(\{\mathbf{x}_i^{t-1}, \tilde{\mathbf{x}}_i\}_{i=1}^{N_V}) - NME(\{\mathbf{x}_i^{t}, \tilde{\mathbf{x}}_i\}_{i=1}^{N_V}) > 0$. In this case, we modify the feature channels from heatmaps to gray-scale images to alleviate alignment errors due to wrong heatmaps. When validation converges again, we stop training.

3.5 Experiments

In this section, we study the performance of recent literature based on facial landmark detection using face images acquired under in-the-wild conditions. Initially, we compare previous published results with our proposals, CHR2C [110] and 3DDE [113], using public evaluation metrics. Afterwards, we include an ablation study to evaluate the importance of each contribution.

3.5.1 Database

It is worth mentioning some problems related to the prevailing facial landmark ground truth on public in-the-wild data sets [122]. On the one hand, the annotation is inherently biased and inconsistent across data sets (see Section 3.5.5). Thus, it is difficult to combine these landmarks. On the other hand, since their manual annotation is a time consuming process, there is a limited amount of public data properly annotated. We have discarded data sets acquired under laboratory conditions due to the necessity to perform experiments in realistic scenarios. In the experiments, we evaluate CHR2C and 3DDE proposals using the following in-the-wild data sets:

- 300W [93] provides 68 manually annotated facial landmarks. We followed the most established approach and divide the annotations into 3148 training and 689 testing images (public competition), divided into 554 easy/semifrontal faces (*Common*) and 135 images in more realistic conditions (*Challenging*). In addition, evaluation is also performed on the 300W private competition using previous 3837 images for training and 600 newly updated images as testing set, also organized in 300 indoor faces and 300 outdoor faces.

- COFW [16] focuses on landmarks partially occluded. There are 1345 training images in total. The testing subset is made of only 507 face images. The annotations include the landmark positions and the binary occlusion labels for 29 points. On average in this data set 28% of the landmarks are occluded.

- AFLW [56] provides a collection of 25993 in-the-wild faces, with 21 facial landmarks annotated depending on their visibility. We have found multiple annotations errors and, consequently, we manually removed some of these faces from our experiments (see Fig. 3.13). From the remaining faces we have randomly chosen 19312 images for training/validation and 4828 instances for testing. Like [54], we divide AFLW test set into intervals of [0°, 30°], [30°, 60°] and [60°, 90°] according to head absolute yaw angle.

(a) Inaccurate landmark location (b) Annotation errors due to incorrect labels

Figure 3.13: Representative ground truth landmarks from AFLW test set. (a) Inaccurate landmarks manually annotated. (b) Annotation errors due to a wrong landmark identifier, *e.g.*, chin manually annotated as right mouth corner.

- WFLW [117] consists of 7500 extremely challenging training and 2500 testing images divided into six subgroups, pose, expression, illumination, make-up, occlusion and blur, with 98 fully manual annotated landmarks.

3.5.2 Evaluation metrics

We use common evaluation metrics to quantify the alignment estimation error. First, we use the Normalized Root Mean Squared Error (NRMSE) or Normalized Mean Error (*NME*) metric, which computes the average euclidean distance L_2 normalized by d_i.

$$NME = \frac{100}{N} \cdot \sum_{i=1}^{N} \left(\sum_{l=1}^{L} \left(\frac{\tilde{\mathbf{w}}_i^l}{||\tilde{\mathbf{w}}_i||_1} \cdot \frac{||\tilde{\mathbf{x}}_i^l - \mathbf{x}_i^l||}{d_i} \right) \right), \tag{3.9}$$

where N is the number of face images, L is the number of landmarks, $\tilde{\mathbf{w}}$ is a vector with the labelled mask, and $\mathbf{x}$, $\tilde{\mathbf{x}}$ represent the estimated and ground truth landmark location respectively.

Depending on the database we report our results using different values of d_i: the ground truth distance between both eye centers (*pupils*), the ground truth distance between the outer eye corners (*corners*) and the ground truth bounding box size (*height*).

In addition, we also compare our results using Cumulative Error Distribution (CED) curves. We calculate AUC_ε as the area under the CED curve for faces with *NME* smaller than ε, and FR_ε as the failure rate representing the percentage of testing faces with *NME* greater than ε.

Finally, we include both precision/recall percentages to compare landmarks visibility predictions.

3.5.3 Implementation details

To train our algorithms, we shuffle the training set of each database and split it into 90% train and 10% validation subsets [110, 113]. We always select the model parameters with lowest validation error. We have trained an individual model for each database using the training/testing configuration presented in Section 3.5.1. All the experiments follow the settings detailed below:

Two-stage CNN+CNN regressor (Section 3.3)

We follow the same procedure training the CHR and CHR2C networks (our heatmap and coordinate regressors respectively). Both stages consist of the same encoder-decoder. We use Adam stochastic optimization with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1e^{-8}$. We train each encoder-decoder stage until validation convergence. The initial learning rate is 10^{-3}, and it is halved every 10 epochs without a validation improvement. The fraction of heatmaps removed between stages is set to $f = 0.5$. The weighting parameter α that balances heatmaps and coordinates losses, is set to $\alpha = 100$.

To increase and decrease the spatial resolution in our encoder-decoder, we introduce convolutional and transposed convolutional layers with stride 2. We reduce the cropped input face from 256×256 to 4×4 pixels by gradually halving the spatial extent of their feature maps across 7 levels. Whenever the spatial resolution is halved we double progressively the number of feature maps, from 64 up to 256 (see Tables 3.1 and 3.2). We also use BatchNormalization before ReLU activations, after each convolutional layer.

We crop faces using the bounding box annotations enlarged by 30%. During training, we have performed data augmentation by applying to each training sample the following random operations: in plane rotation between $\pm 45°$, scaling by $\pm 15\%$, translation by $\pm 5\%$ of the bounding box size, mirroring face image horizontally and colour change multiplying each HSV channel by a random value between $[0.5, 1.5]$. Additionally, we include synthetic rectangular occlusions to enforce the second encoder-decoder to learn a valid face shape. Data augmentation is a critical step to reduce overfitting on small training data sets. See sample results in Fig. 3.14.

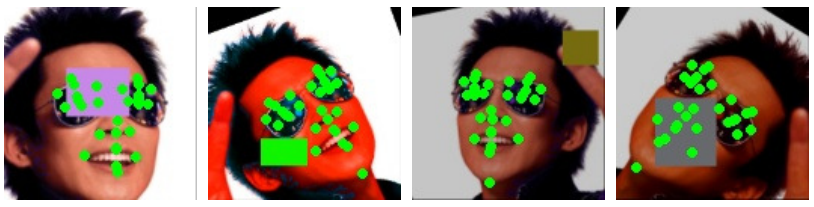


Figure 3.14: Data augmentation including random synthetic occlusions.

Training end-to-end the full CHR2C framework by fine-tuning previous CHR weights as mentioned in Section 3.3, takes 54 hours using a NVidia GeForce GTX 1080Ti GPU (11GB) with a batch size of 6 images on WFLW. At run-time our network requires on average 100 ms to process each detected face, a rate of 10 FPS, using Python, Tensorflow and OpenCV libraries. This processing speed could be halved by reducing the number of stages, at the expense of a slight reduction in accuracy.

Two-stage CNN+ERT regressor (Section 3.4)

Our hybrid 3DDE algorithm consists of a single CNN and a coarse-to-fine ERT, whose configuration parameters are described below. Same as previous CHR2C encoder-decoder, we train our CNN until validation convergence, using Adam stochastic optimization with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e^{-8}$ parameters and an initial learning rate set to 10^{-3}, which is halved every 10 epochs without a validation improvement.

However, in this case, the cropped input face is reduced from 160×160 to 1×1 pixels gradually halving their size across 8 levels, through convolutional and transposed convolutional layers with stride 2 and kernel size 2×2[2]. All layers contain 68 feature maps. We also apply BatchNormalization after each convolution.

Similarly, we crop faces using the ground truth bounding boxes annotations enlarged by 30%, and we generate random in plane rotations between ±45°, scale changes by ±15%, translations by ±5% of bounding box size, mirroring face images horizontally, and adding random rectangular occlusions.

Between both stages, we apply a Gaussian filter with $\sigma = 33$ to the output heatmaps to stabilize the ERT initialization (see Algorithm 6). We train the coarse-to-fine ERT with the Gradient Boosting algorithm [48], which requires a maximum of $T = 20$ stages of $K = 50$ regression trees per stage. Our 3DDE requires fewer decision trees than previous literature [17, 16, 55, 89, 63] because we substitute simple gray-scale with discriminative CNN feature channels, and the rigid face deformation has already been estimated in the initialization $\mathbf{x}^0$ (*i.e.*, the mean shape requires more stages to converge to the solution).

The depth of trees is set to 4. The number of tests to choose the best split parameters, θ, is set to 200. We resize each image to set the face size to 160×160 pixels. We generate $Z = 25$ initializations in the robust RANSAC+POSIT scheme. We augment the training shapes to create a set, $\mathcal{S}_A$, of at least $N_A = 60$K samples to train the coarse-to-fine ERT. To avoid overfitting we use a shrinkage factor $\nu = 0.1$ and subsampling factor $\eta = 0.5$. We also set the FREAK pattern diameter to be gradually reduced an overall 15% along the ERT stages (see Fig. 3.5a). Finally, our regressor triggers the coarse-to-fine strategy once the training error is below the validation error, *e.g.*, the fifth stage in the model trained on WFLW (see Fig. 3.16a).

Training the whole 3DDE from scratch takes 48 hours for WFLW (34 hours fine-tuning the CNN and 14 hours for the ERT) using a NVidia GeForce GTX 1080Ti GPU (11GB) with a batch size of 32 images, and a dual Intel Xeon Silver 4114 CPU at 2.20GHz (2×10 cores/40 threads, 128 GB). At run-time our hybrid method requires on average 65 ms to process each face, where the CNN takes 60 ms and the coarse-to-fine ERT 5 ms. It achieves an overall frame rate of 15.3 FPS, which is a 35% faster than previous CHR2C, using C++, Tensorflow and OpenCV libraries.

[2] 5×5 images are reduced to 2×2 pixels applying a kernel size of 3×3

3.5.4 Ablation study

In this section, we analyze the contribution of certain modules of CHR2C and 3DDE to study their effect in the overall behaviour. As mentioned previously, we consider that current research on facial landmark detection is lacking of a global head model to preserve the face shape under occlusions. Hence, we have presented two different ways to address this problem:

Two-stage CNN+CNN regressor (Section 3.3)

Initially, we evaluate the importance of the second CNN stage (see Fig. 3.9) and our fully connected layer with shared weights attached to the end to improve the robustness of CHR2C against partial occlusions. We show in Table 3.5 the comparison among the three configurations, CHR using a single stage (S=1), two-stage CHR (S=2), and the full CHR2C in 300W, COFW and WFLW.

Method	Common pupils NME	Challenging pupils NME	Full pupils NME
CHR ($S=1$)	4.21	8.65	5.08
CHR ($S=2$)	4.04	7.58	4.73
CHR2C	3.96	7.44	4.64

(a) 300W public

Method	Indoor corners NME	Outdoor corners NME	Full corners NME
CHR ($S=1$)	4.29	4.27	4.28
CHR ($S=2$)	3.90	3.89	3.90
CHR2C	3.78	3.77	3.77

(b) 300W private

Method	pupils NME
CHR ($S=1$)	6.02
CHR ($S=2$)	5.30
CHR2C	5.09

(c) COFW

Method	Full corners			Pose corners			Expression corners			Illumination corners			Make-up corners			Occlusion corners			Blur corners		
	NME	AUC_{10}	FR_{10}	NME	AUC_{10}	FR_{10}	NME	AUC_{10}	FR_{10}	NME	AUC_{10}	FR_{10}	NME	AUC_{10}	FR_{10}	NME	AUC_{10}	FR_{10}	NME	AUC_{10}	FR_{10}
CHR ($S=1$)	5.02	53.48	6.64	9.33	24.89	26.99	5.44	49.95	6.68	6.68	54.30	5.01	4.91	54.04	7.28	6.41	43.94	13.72	5.75	46.96	8.53
CHR ($S=2$)	4.57	56.39	4.20	8.10	29.70	20.24	4.89	53.55	3.50	4.58	56.98	3.29	4.36	57.24	3.39	5.64	48.00	8.28	5.28	50.24	6.20
CHR2C	4.39	57.55	3.55	7.58	31.85	18.09	4.72	55.04	3.82	4.39	57.94	2.57	4.18	58.82	1.94	5.37	49.63	7.06	5.09	51.54	5.30

(d) WFLW

Table 3.5: Ablation study in 300W public, 300W private, COFW and WFLW respectively. CHR (S=1) and (S=2) represent our individual and two-stage heatmap regressors, being the landmark location the maximum response on each heatmap. CHR2C incorporates a last dense layer to regress coordinates from heatmaps.

The experiments carried out highlight the importance of the synthetic random occlusions and the *map dropout* layer between stages to enforce the CNN to learn the location of occluded landmarks from their neighbours. `CHR (S=2)` decreases face alignment *NME* between 6.89% and 8.96% in 300W and WFLW. However, the reduction becomes even more notorious in COFW, which is the standard benchmark to evaluate partial occlusions since, on average, 28% of the facial landmarks are occluded. In this case, it reduces the *NME* in 11.96%. On the other hand, our full `CHR2C` architecture, reaches even superior precision, *i.e.*, an additional improvement rate between 1.90% and 3.96% in all data sets.

Two-stage CNN+ERT regressor (Section 3.4)

At this point, we show the results obtained by different configurations of our framework when evaluated on WFLW. We have selected WFLW in our study because it allows the analysis of results stratified by different types of difficulties (*i.e.*, facial expressions, large poses, illumination changes, and so on). In this case, since there are many profile faces, we

use the *height* as normalization for the *NME*. Thus, the numerical values are not directly comparable to those in Table 3.5.

Our first analysis recaps the importance of our coarse-to-fine ERT stage in place of a second CNN, like CHR2C. The main goal of the ERT is to implicitly enforce a valid face shape, to improve the robustness of 3DDE against partial occlusions. We show in Table 3.6 the evolution of our last stage (coarse-to-fine ERT) beginning from the original Dlib library [55]. In this case, we always employ the mean shape $\mathbf{x}^0$ as initialization, and report both training and testing *NME* to manage the risk of overfitting.

Method	Training set height	Full height		
	NME	NME	AUC_4	FR_4
$T = 10$ + $K = 500$ + SE (10 random pixels per landmark) + GB (all)	3.21	3.81	31.06	26.80
$T = 10$ + $K = 500$ + SE (43 random pixels per landmark) + GB (all)	3.03	3.67	33.19	24.03
$T = 100$ + $K = 50$ + SE (43 random pixels per landmark) + GB (all)	3.01	3.61	34.10	23.16
$T = 100$ + $K = 50$ + SE (SURF descriptor) + GB (all)	2.03	3.16	38.32	18.04
$T = 100$ + $K = 50$ + SE (FREAK pattern, 1 channel) + GB (all)	3.36	3.62	34.82	24.16
$T = 100$ + $K = 50$ + SE (FREAK pattern, 8 channels) + GB (all)	3.18	3.56	35.48	22.88
$T = 100$ + $K = 50$ + SE (FREAK pattern, 8 channels) + GB (single)	3.36	3.49	36.27	22.51

Table 3.6: Ablation study on WFLW. SE denotes simple features extracted from a SURF descriptor [6] or a difference of pixel intensities from the gray-scale/smoothed channels. GB represents Gradient Boosting optimization, whereas "all" and "single" denote whether Eq. 3.3 is minimized according to the whole shape or a single landmark respectively.

From these results it is clear that the ERT does not improve the performance proportionally to the amount of feature extraction iterations T. We keep the number of decision trees constant (sequence of 5K regression trees), and evaluate the frequency of feature extraction using $T = 10$ (3.67 *NME*) and $T = 100$ (3.61 *NME*). A higher frequency yields a marginal error reduction, at the expense of increasing the computational time required.

Another noticeable finding is that features relying on SURF [6] generate a promising alignment error reduction (3.16 *NME*) in comparison with differences of pixel values from a single gray-scale channel (3.62 *NME*) and Gaussian smoothed channels (3.56 *NME*) at the expense of worse computational efficiency. It is relevant because it proves that using more robust features we reduce the *NME* in 12.7%. However, this performance is still far from satisfactory, and leads us to investigate how to include more discriminative features extracted from a CNN into our ERT. In fact, this has been our main motivation during the construction of the hybrid proposal.

The 3DDE framework is based on three key ideas: 3D initialization, an ERT regressor operating on probabilistic CNN heatmaps and a coarse-to-fine strategy. In Table 3.7, we analyze the contribution of each to the overall performance of our algorithm. It is worth mentioning that we use `CNN` in two situations. It is required to initialize the procedure in Section 3.4.1 (`3D`), and also to extract deep features from the CNN (`DE`). Here, `MS+SE+CF` represents the top performance configuration that has been already reported in Table 3.6 (3.49 *NME*), `T=100+K=50+SE(FREAK,8)+GB(single)`. Finally, `CF` represents the coarse-to-fine scheme.

The 3D initialization is a key step because it takes care of the rigid component of face pose, so that the ERT only models non-rigid deformations. Moreover, the projection of the 3D face model is a correct 2D shape, a requirement for the ERT to converge to a valid face shape [17]. In Table 3.7, we report a *NME* reduction greater than 27% using plain features (`CNN+3D+SE`), and a reduction of almost 9% using robust CNN features (`CNN+3D+DE`) in

Method	Full height			Pose height			Expression height			Illumination height			Make-up height			Occlusion height			Blur height		
	NME	AUC_4	FR_4	NME	AUC_4	FR_4	NME	AUC_4	FR_4	NME	AUC_4	FR_4	NME	AUC_4	FR_4	NME	AUC_4	FR_4	NME	AUC_4	FR_4
MS+SE+CF	3.47	37.01	21.40	6.42	17.05	47.85	3.70	32.09	22.92	3.46	37.74	18.19	3.94	33.59	28.64	4.86	24.20	38.31	4.11	29.23	28.33
CNN+3D+SE	2.52	41.10	11.56	3.53	24.08	28.83	2.90	33.22	15.92	2.53	41.85	10.45	2.59	39.08	15.53	3.06	31.10	22.14	2.91	33.98	15.78
CNN+MS+DE	2.23	49.77	7.04	3.33	35.13	17.79	2.56	45.15	8.91	2.17	49.29	5.87	2.33	46.85	9.70	2.69	40.33	12.90	2.53	42.71	9.57
CNN+3D+DE	2.03	51.14	5.47	2.68	39.55	11.96	2.21	46.66	7.96	2.11	50.09	5.01	2.13	48.57	7.28	2.56	40.83	12.36	2.40	43.84	8.27
CNN+3D+DE+CF	2.01	51.67	5.20	2.63	39.90	10.73	2.15	48.19	5.73	2.06	50.79	4.87	2.12	49.05	7.28	2.54	40.94	12.22	2.39	43.93	8.02

Table 3.7: Ablation study on WFLW. MS and 3D represent the initialization $\mathbf{x}^0$ using 2D mean shape and projections of a 3D mean face respectively. SE and DE represent the type of features used in the cascade being simple gray-scale features and deeper features from heatmaps respectively. The CNN+3D+DE+CF row represents the full 3DDE approach.

the *Full* set. Of course, the 3D initialization is fundamental to achieve good performance in presence of large face rotations. Thus, it provides the largest improvement in the *Pose* subset. Similarly, the use of CNN probability maps as feature channels improves the *NME* in the *Full* set in about 20%. The large receptive fields of CNNs are specially helpful in challenging situations, specifically those in the *Pose* and *Occlusion* subsets.

Although the initialization has solved the rigid pose estimation, we can see in Fig. 3.12 that the result of the RANSAC+POSIT alignment is far from satisfactory. This is caused by the difference between the 3D shape of the generic model and that of the real face in the image, and also by the imprecision in the estimations of the initial set of landmarks. However, it does a good job approximately estimating the location and pose of the target face. This enables the initialization of the ERT, $\mathbf{x}^0$, from a configuration roughly on top of the target face. Finally, we align the 2500 test samples of WFLW and plot the *NME* distribution in Fig. 3.15, produced both with the `CNN+3D` regressor (RANSAC+POSIT) and the full `CNN+3D+DE+CF` framework (3DDE). The values of percentiles 10 and 90 of the *NME* distribution are 3.71 and 6.87 for the `CNN+3D` regressor and 1.03 and 3.32 for the `CNN+3D+DE+CF`. So, on average, the full regressor reduces in about 60% the *NME* achieved by the rigid initialization.

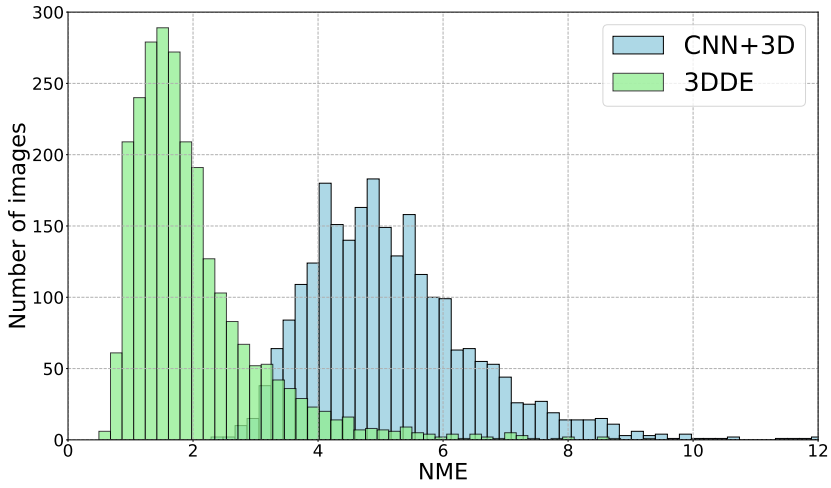
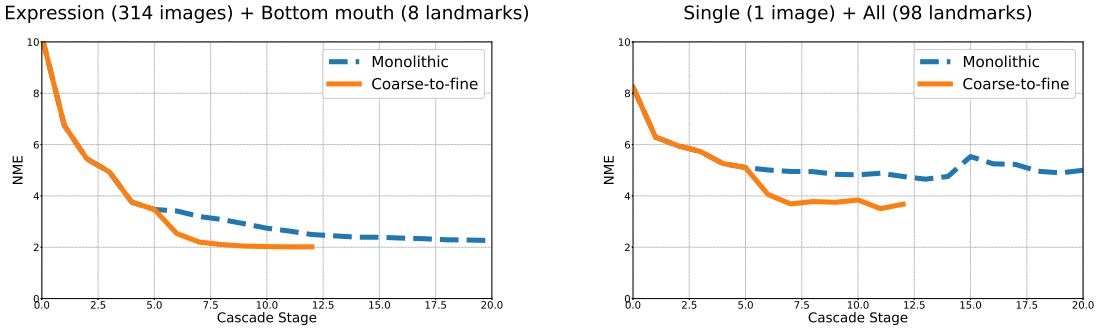


Figure 3.15: Sample *NME* distribution produced by the CNN+3D and 3DDE regressors. We employ the *height* as normalization.

The coarse-to-fine strategy from our ERT provides significative local improvements in

challenging cases, with rare facial part combinations (see Fig. 3.16a). For this reason, the largest gain of `CNN+3D+DE+CF` vs `CNN+3D+DE` occurs in the *Expression* subset, 2.7%. Even though this strategy provides improvements in all the subsets, the actual *NME* differences are washed out when averaged over the number of landmarks in the face and the number of images in the subset. They may be appreciated by looking into specific data subsets or samples (see Fig. 3.16a), such as the left eyebrow/eye location improvement in Fig. 3.16b and 3.16c.



(a) Evolution of the *NME* through the different stages in the cascade

(b) Monolithic (c) Coarse-to-fine

Figure 3.16: Example of a monolithic ERT regressor vs our coarse-to-fine approach. (a) *NME* evolution through the stages in the cascade (left plot, 8 mouth landmarks for all test images in the *Expression* subset; right plot, all 98 landmarks in one image). (b) Predicted shape and zoom-in with a monolithic regressor. (c) Predicted shape and zoom-in with our coarse-to-fine approach.

3.5.5 Cross-dataset evaluation

Manual landmark annotation is a time consuming process that usually presents some inconsistencies across data sets. For this reason, it is difficult to combine multiple data sets for evaluation. At this point, we also utilize the subset of distinct landmarks shown in Fig. 3.1, to perform cross-dataset evaluation. We consider them distinct because they are accurately located by a human annotator.

We benefit from the fact that 3DDE may be trained in presence of missing and occluded landmarks in the training set. This has enabled us to perform cross-dataset experiments to evaluate the quality of different benchmarks and the generalization of the 3DDE regressor trained on them. We train and evaluate different models respectively using the training and test sets of each database. We have also performed one additional experiment training 3DDE with the training sets of all data sets, and evaluating it successively with the test sets of each of them. We denote this experiment with label `All`. In Table 3.8, we show the results of our evaluation.

On the one hand, the smallest database, COFW, has the worst cross-dataset results. On the other hand, the database with greatest diversity, WFLW, achieves the best results.

Train \ Test	300W	COFW	AFLW	WFLW	All
300W	**2.00**	3.11	4.90	3.44	4.15
COFW	3.68	**2.09**	4.56	4.03	4.19
AFLW	4.19	2.51	**2.15**	3.29	2.65
WFLW	2.57	2.53	3.28	**1.70**	2.71
All	2.34	2.23	2.41	1.96	**2.26**

Table 3.8: Cross-dataset experiment using only distinct landmarks to compute *NME*. We employ the *height* as normalization.

Moreover, the model `All`, trained with the training sets of all data sets, is able to improve, in all cross-dataset experiments the models trained in a single database. However, the most prominent outcome of this experiment is that we always achieve the best result when training with the train subset of the same database. And this holds even when compared against the model trained with all data sets, confirming the existence of the so-called "data set bias" in current benchmarks [105] that may limit the generalization capabilities of regressors trained on present data sets. To the best of our knowledge, this is the first time such a problem has been raised in the facial landmark detection field.

In the final experiment, we employ model `All` to evaluate the *NME* of each landmark using the test sets of all data sets (see Fig. 3.17). The landmarks with highest *NME* are those related to the earlobes (15, 19), the bottom of the mouth (23) and the chin (24). The results of the experiment found clear support of a problem using imprecise manual landmark annotation, *e.g.*, earlobes location inferred due to self-occlusion in extreme poses or hair occlusion respectively.

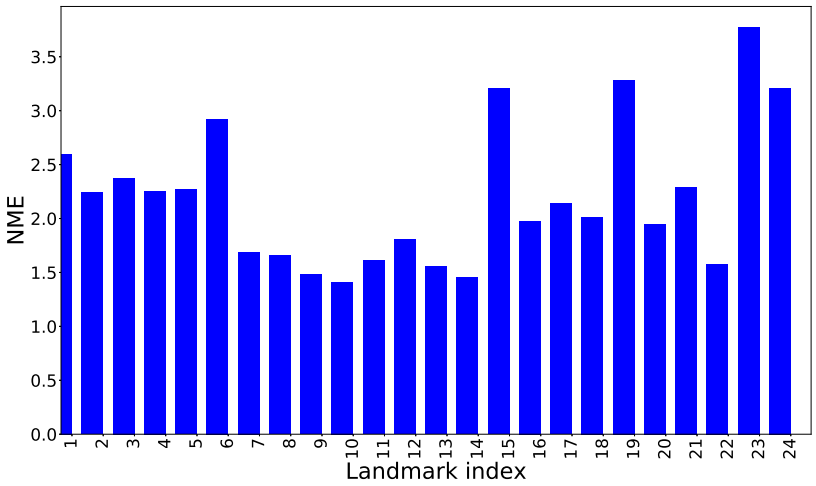


Figure 3.17: Location of distinct face landmarks and the *NME* related to each landmark.

3.5.6 Comparison with the state-of-the-art

In this section, we compare our models using CHR2C [110] and 3DDE [113] with other contemporary approaches in the literature by using their published results on facial

landmark detection. In Table 3.9 we initially test our methods against the popular 300W public benchmark. Since its release in 2014, there has been a tremendous amount of research done, including CSR frameworks using a sequence of boosted regressors, based on Random Forests [16, 17, 55, 63, 89, 120] or CNNs [58, 129, 36, 104, 117, 61] respectively.

Method	Common		Challenging		Full			
	pupils	corners	pupils	corners	pupils	corners		
	NME	NME	NME	NME	NME	NME	AUC_8	FR_8
RCPR [16]	6.18	-	17.26	-	8.35	-	-	-
ESR [17]	5.28	-	17.00	-	7.58	-	43.12	10.45
SDM [124]	5.60	-	15.40	-	7.52	-	42.94	10.89
ECSAN [139]	5.42	-	11.80	-	6.67	-	-	-
ERT [55]	-	-	-	-	6.40	-	-	-
LBF [89]	4.95	-	11.98	-	6.32	-	-	-
PIFAS [54]	5.43	-	9.88	-	6.30	-	-	-
SAN [31]	-	3.34	-	6.60	-	3.98	-	-
CFSS [144]	4.73	-	9.98	-	5.76	-	49.87	5.08
cGPRT [63]	-	-	-	-	5.71	-	-	-
DDN [132]	-	-	-	-	5.65	-	-	-
TCDCN [141]	4.80	-	8.60	-	5.54	-	-	-
MDM [107]	-	-	-	-	5.88	-	52.12	4.21
HF-ResNet [86]	-	-	8.18	-	-	-	-	-
3DDFA [145]	5.09	-	8.07	-	5.63	-	-	-
RCN [50]	4.67	-	8.44	-	5.41	-	-	-
ECT [138]	4.66	-	7.96	-	5.31	-	-	-
DSRN [76]	4.12	-	9.68	-	5.21	-	-	-
DAN [58]	4.42	3.19	7.57	5.24	5.03	3.59	55.33	1.16
TSR [72]	4.36	-	7.56	-	4.99	-	-	-
RAR [123]	4.12	-	8.35	-	4.94	-	-	-
SHN [129]	4.12	-	**7.00**	**4.90**	4.68	-	-	-
DU-Net [104]	-	2.82	-	5.07	-	3.26	-	-
PCD-CNN [61]	3.67	-	7.62	-	4.44	-	-	-
Wing [36]	**3.27**	-	7.18	-	**4.04**	-	-	-
CHR2C [110]	3.96	2.85	7.44	5.15	4.64	3.30	58.92	**1.16**
3DDE [113]	3.73	**2.69**	7.10	4.92	4.39	**3.13**	**61.24**	1.30

Table 3.9: Error of face alignment methods using the 300W public test set.

As we mentioned in Section 3.5.1, the 300W public test set consists of 554 semifrontal faces (*Common*) and 135 images in more realistic conditions (*Challenging*). Consequently, an algorithm like [36], which introduces a loss conceived to pay more attention to the minimization of samples with small alignment errors, is the one with the best reported result (4.04 *NME*). Their loss focuses on the *Common* subset that represent 80% of the *Full* set. In the *Challenging* subset of the 300W public competition, SHN [129] gets state-of-the-art results (7.00 *NME*) because they cascaded four Hourglass networks [81], which produce better features in comparison with our two-stage proposals, CHR2C and 3DDE.

However, CHR2C outperforms most published results in 300W public using a cascade of CNN regressors such as DDN [132], DAN [58], MDM [107], RAR [123] or TSR [72],

although its *NME* is 5.91% worse than SHN [129]. However, their model, based on the concatenation of four CNNs, has more computational requirements. We do not take into account published LAB [117] results in 300W due to their inconsistencies between the *pupils* and *corners* normalization. We will compare with LAB in WFLW for which we have confirmed the published results using their public code. It is also evident that 3DDE exhibits better capability in handling typical and challenging cases than previous ERT regressors such as RCPR [16], ESR [17], ERT [55], LBF [89], CFSS [144] or cGPRT [63]. In fact, compared to cGPRT [63], which reports a 5.71 *NME* in the *Challenging* subset, 3DDE achieves 23.11% improvement using the CNN robust features.

It is also worth mentioning that 3DDE obtains a reduction in *NME* of 5.38% compared to CHR2C in all subsets, which denotes the excellent accuracy achieved by the coarse-to-fine ERT scheme enforcing valid face shapes through the deep robust features extracted from the CNN. We also assess the improvement achieved by the 3D initialization and the coarse-to-fine ERT comparing the accuracy of 3DDE in the *Full* set (4.39 *NME*), with one U-Net model like RCN [50] (5.41 *NME*). It roughly represents a 18.85% improvement in the *NME*.

In Table 3.10 we evaluate the performance of CHR2C and 3DDE in the well-known *Indoor* and *Outdoor* subsets of the 300W private competition. Among the reported results on the private 300W benchmark, 3DDE is the one with the lowest reported error (3.73 *NME*). However, in this case, CHR2C is only marginally worse (3.77 *NME*), *i.e.*, it represents a 1.06% improvement. So, when compared with previous reduction of 5.38% on 300W public data set, it seems that CHR2C is going to perform better than 3DDE in data sets with a high proportion of challenging face images. In fact, the FR obtained by 3DDE (1.30 FR_8 and 2.33 FR_8) is worse than that reported by CHR2C (1.16 FR_8 and 0.83 FR_8) in 300W public and private benchmarks respectively.

The CED curves that we present in Fig. 4.12b show that the face shape enforced by 3DDE effectively achieves better performance, except for the most difficult faces, with *NME* above 6.5. In fact, 3DDE is the one with the highest published AUC on both 300W public (61.24 AUC_8) and private (53.94 AUC_8).

Method	Indoor corners	Outdoor corners	Full corners		
	NME	NME	NME	AUC_8	FR_8
ESR [17]	-	-	-	32.35	17.00
CFSS [144]	-	-	-	39.81	12.30
cGPRT [63]	-	-	-	41.32	12.83
MDM [107]	-	-	5.05	45.32	6.80
ECT [138]	-	-	-	45.98	3.17
DAN [58]	-	-	4.30	47.00	2.67
SHN [129]	4.10	4.00	4.05	-	-
CHR2C [110]	3.78	3.77	3.77	52.85	**0.83**
3DDE [113]	**3.74**	**3.71**	**3.73**	**53.94**	2.33

Table 3.10: Error of face alignment methods using the 300W private test set.

Table 3.11 shows the performance of CHR2C and 3DDE using the COFW data set, which mainly focuses on occluded facial landmarks. Both CHR2C and 3DDE deliver significantly better results than the literature due to our motivation to inherit discriminative

features based on CNNs, whereas preserving the face shape to cope with severe occlusions. In this case, CHR2C is marginally better than 3DDE, establishing a new state-of-the-art. In terms of landmarks visibility estimation, 3DDE, including the visibilities into the CSR scheme, performs better than [16, 119, 120] (85.92/51.04 precision/recall). However, ECT reports a similar, although not fully comparable performance (80/63.4 precision/recall) by inferring each visibility depending on the confidence associate to its associated heatmap, *i.e.*, a heatmap with high variability is prone to be non-visible.

Method	pupils NME	occlusion precision/recall
RCPR [16]	8.50	80/40
TCDCN [141]	8.05	-
Wu *et al.* [119]	6.40	80/44.43
RAR [123]	6.03	-
DAC-CSR [37]	6.03	-
Wu *et al.* [120]	5.93	80/49.11
ECT [138]	5.98	80/63.4
SHN [129]	5.6	-
PCD-CNN [61]	5.77	-
3DDE [113]	5.11	85.92/51.04
CHR2C [110]	**5.09**	-

Table 3.11: Error of face alignment methods using COFW.

Since CHR2C and 3DDE are able to train with unannotated landmarks, we train and evaluate using AFLW (see Table 3.12). This is a challenging database, not only because of its size and the large variability of face poses, but also because of the large number of faces with occluded landmarks, which are unannotated. In fact, few approaches can train with missing landmarks. This is the reason for the small number of methods that test in AFLW.

Although the results are not strictly comparable, because each paper uses its own train and test subsets, we achieve again state-of-the-art results using CHR2C (2.07 NME) and 3DDE (2.10 NME). There are some other works as TCDCN [141], RAR [123] and RCN [50] which use the eye *pupils* distance to normalize the NME, which it is not reliable at all with profile faces.

Finally, we study CHR2C and 3DDE using the newly released WFLW database [117]. It also enables us to evaluate different sources of variability (*i.e.*, expressions, illumination, make-up, occlusions and blur). In Table 3.13 we provide the results of various competing methods, normalized by the outer eye *corners* distance. 3DDE and CHR2C outperform our main competitors in all the WFLW subsets by a large margin. In comparison with LAB [117], both 3DDE and CHR2C algorithms generate an impressive NME reduction of 11.2% and 16.7% respectively. Additionally, we provide the best AUC and FR in all subsets, confirming that our key motivation to preserve the face shape in each result is a good strategy under all capture conditions (easy/semifrontal and difficult/profile) including all subsets that contain multiple types of difficulties.

In this case, 3DDE has been further improved by CHR2C in a 6.2%. We hypothesize that the reason for this is that current models that cascade multiple CNNs such as CHR2C, perform better in challenging situations. In fact, CHR2C always achieves the best FR_{10}

Method	21 landmarks			
	[0°, 30°] height *NME*	[30°, 60°] height *NME*	[60°, 90°] height *NME*	Full height *NME*
RCPR [16]	5.43	6.58	11.53	7.85
CCR [140]	-	-	-	5.72
PIFAS [54]	-	-	-	4.45
Hyperface [86]	3.93	4.14	4.71	4.26
Kepler [60]	-	-	-	2.98
AIO [87]	2.84	2.94	3.09	2.96
HF-ResNet [86]	2.71	2.88	3.19	2.93
Binary-CNN [13]	2.77	2.60	2.64	2.85
PCD-CNN [61]	2.33	2.60	2.64	2.49
3DDE [113]	2.10	2.00	2.04	2.06
CHR2C [110]	**2.07**	**1.86**	**1.81**	**1.98**

Table 3.12: Error of face alignment methods using AFLW.

for all subsets in WFLW, being this benchmark the one that provides the most challenging face images.

Method	Full corners			Pose corners			Expression corners			Illumination corners			Make-up corners			Occlusion corners			Blur corners		
	NME	AUC_{10}	FR_{10}	*NME*	AUC_{10}	FR_{10}	*NME*	AUC_{10}	FR_{10}	*NME*	AUC_{10}	FR_{10}	*NME*	AUC_{10}	FR_{10}	*NME*	AUC_{10}	FR_{10}	*NME*	AUC_{10}	FR_{10}
ESR [17]	11.13	27.74	35.24	25.88	1.77	90.18	11.47	19.81	42.04	10.49	29.53	30.80	11.05	24.85	38.84	13.75	19.46	47.28	12.20	22.04	41.40
SDM [124]	10.29	30.02	29.40	24.10	2.26	84.36	11.45	22.93	33.44	9.32	32.37	26.22	9.38	31.25	27.67	13.03	20.60	41.85	11.28	23.98	35.32
CFSS [144]	9.07	36.59	20.56	21.36	6.32	66.26	10.09	31.57	23.25	8.30	38.54	17.34	8.74	36.91	21.84	11.76	26.88	32.88	9.96	30.37	23.67
LAB [117]	5.27	53.23	7.56	10.24	23.45	28.83	5.51	49.51	6.37	5.23	54.33	6.73	5.15	53.94	7.77	6.79	44.90	13.72	6.32	46.30	10.74
3DDE [113]	4.68	55.44	5.04	8.62	26.40	22.39	5.21	51.75	5.41	4.65	56.02	3.86	4.60	55.36	6.79	5.77	46.92	9.37	5.41	49.57	6.72
CHR2C [110]	**4.39**	**57.55**	**3.55**	**7.58**	**31.85**	**18.09**	**4.72**	**55.04**	**3.82**	**4.39**	**57.94**	**2.57**	**4.18**	**58.82**	**1.94**	**5.37**	**49.63**	**7.06**	**5.09**	**51.54**	**5.30**

Table 3.13: Error of face alignment methods using WFLW.

In Fig. 3.18 we visually compare some representative results of our main competitor LAB [117] and our hybrid approach 3DDE [113] in the WFLW test set. As we notice, the main problem in LAB predictions lies in the two key contributions of our model: how to be robust against partial occlusions, and how to remove the rigid face deformation to be robust against extreme head pose variations.

Conclusions

The estimation of the non-rigid deformation of the face is still far from being completely solved under most in-the-wild conditions, as shown in Fig. 3.19. Nevertheless, it is worth noting the notably evolution of our 3DDE procedure (see Fig. 3.19e) against the original estimations obtained using Dlib library [55] (see Fig. 3.2).

In this chapter, we have adopted a cascade scheme in which a sequence of regressors iteratively refines their estimations to reach a final solution. On the one hand, we propose CHR2C [110], which is based on a pair of networks that exploits the benefits of a high capacity cascade of CNN regressors. We additionally add a simple but important final stage to estimate landmark coordinates from heatmaps, training it in a way that takes

6.407 20.798 24.565 16.859

(a) LAB [117]

3.829 13.105 6.719 8.168

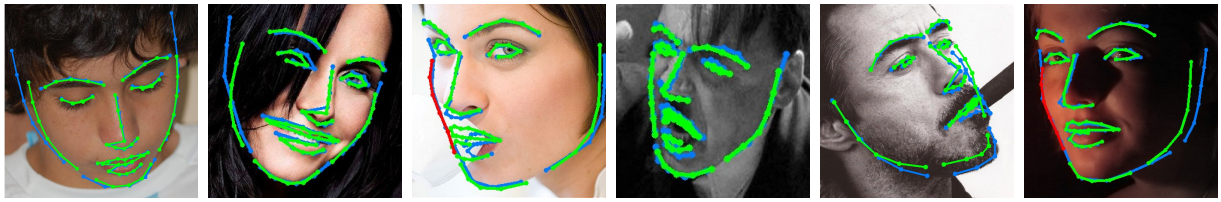
(b) 3DDE [113]

Figure 3.18: Comparison between LAB [117] and 3DDE [113] results using 300W private test set. We also report below their corresponding *NME*, normalized by the eye *corners* distance. Blue and green colours represent ground truth and predictions respectively.

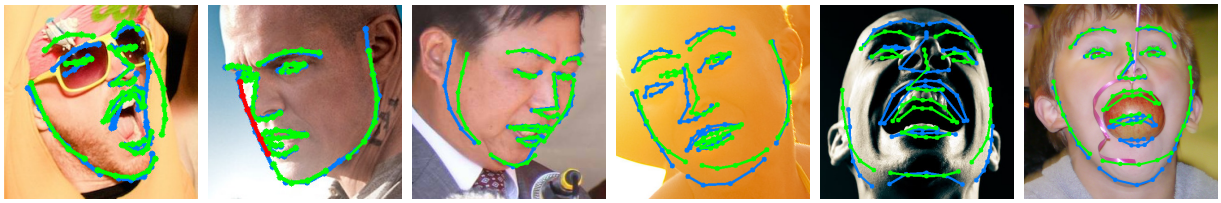
occlusions into account. On the other hand, we propose 3DDE [113], a hybrid method that leverages on good properties of CNNs and ERTs. The CNN provides robust landmark estimations, but weak face shape enforcement, whereas the ERT is able to preserve the face shape and achieve better accuracy in landmark detection, but it only converges with a good initialization.

As a result, 3DDE obtains an excellent accuracy when the heatmaps provide a good approximate estimation for the location of all landmarks. Thus, it outperforms CHR2C in 300W public, 300W private and COFW. However, in challenging situations, with extreme poses and exaggerated expressions, improving the regressor capacity is more important as we have shown in the experiments. CHR2C achieves better results than 3DDE in AFLW and WFLW.

A critical question here is whether the models trained with present data sets generalize to the situations present in real-life operation. The cross-dataset experiments performed above (see Section 3.5.4) reveal the existence of a significant data set bias in present benchmarks that limits the generalization of models trained with them. So, further work in this direction is required to improve the performance of present face alignment algorithms.



(a) 300W public



(b) 300W private

(c) COFW

(d) AFLW

(e) WFLW

Figure 3.19: Representative sample face images considered errors according to FR_{ε} using 3DDE in 300W, COFW, AFLW and WFLW testing subsets. Green, red and blue colours show visible, occluded predictions and ground truth respectively.

Chapter 4

Simultaneous head pose and facial landmark estimation

Multi-task learning (MTL) emerged as an exceptional training scheme harnessing the dependencies among some related tasks [18]. The hypothesis is that common information should be shared among these tasks and training them simultaneously can result in better generalization performance (*i.e.*, it improves the performance on new, unseen data) than the one we would achieve if we learn each task independently.

The standard scheme in the computer vision field is to learn a single task at a time. However, there are some tasks of interest, *e.g.*, *face alignment*, where we can recognize two or more clearly differentiable problems that we would optimize at the same time, *e.g.*, rigid and non-rigid face deformation. There is an increasing amount of research [146, 26, 126, 61] that proves how to enhance the non-rigid estimation by previously disentangling the head pose estimation in another regressor. This scheme results in two different models trained sequentially, where the landmark detection is assisted by a preceding rigid pose regression stage, which is not as optimal as other methods trainable end-to-end, in terms of training and inference speed.

In this chapter, we jointly address the rigid and non-rigid face deformation problems by estimating head pose, facial landmark location and their visibility using a MTL scheme. The main objective is to substitute the sequence of CNNs defined in Section 3.3 with a Multi-task Neural Network (MNN) that, in addition to the heatmaps, also computes the rigid pose initialization and the facial landmarks visibilities. To this end, we propose a framework that provides a robust initialization and visibility estimation, while keeping the coarse-to-fine ERT at the end to preserve the face shape (see Section 3.2). The visibility prediction also let us discard predictions produced by occluded landmarks in the ERT, to outperform 3DDE [113] presented in Section 3.4.

MTL is inspired by human learning [18]. We apply the knowledge acquired by learning related tasks concurrently to help understand a new task, *e.g.*, a person who is learning the Spanish language or how to drive a motorcycle would benefit from simultaneously learning Portuguese or keeping the balance in a bicycle respectively. In computer vision, MTL has been widely used to learn similar tasks (*e.g.*, human pose estimation with action recognition [41], object detection with attribute prediction [77] or semantic segmentation with surface normal prediction [77, 57]). These approaches perform MTL by sharing common features in early hidden layers assuming that their tasks are closely related. The most difficult problem hence is how to identify the correlation among tasks because sharing data between unrelated tasks actually hurts performance, *i.e.*, a phenomenon regarded as *negative transfer* [116]. Zamir *et al.* [136] propose a computational approach to model the relationship between common tasks in computer vision (2D, 3D or semantic information via knowledge distillation) to identify redundancies across these tasks. Their product is a computational taxonomic map for task transfer learning.

The concept of *transfer learning* presented in Section 2.3 is closely connected to MTL. Transfer learning sequentially exploits the knowledge acquired from a pre-trained model focused on one or more similar tasks to improve the performance of another task at hand, whereas MTL concurrently enhances the performance of all these related tasks by sharing a common representation. Thus, transfer learning prioritizes the new target task, whereas MTL assigns similar importance to all tasks. In our proposal, we follow both strategies. First, we train each model from scratch using an encoder-decoder with the most difficult task (*i.e.*, facial landmark detection). Then, we fine-tune from this pre-trained network to perform MTL including the rigid pose estimation and landmarks visibilities into the optimization. Our main goal is to maintain the accuracy of heatmaps, but outperforming the results of head pose and visibility tasks through the facial landmark location. This will help to improve the performance of the ERT.

As a result, we display in Fig. 4.1 multiple frames from two different sample videos from 300VW [96] where head pose, landmark location and their visibility are estimated at the same time. It is also worth mentioning that we process each frame using the algorithm introduced in this chapter, where it is noticeable not only its robustness estimating the head pose and landmarks visibility, but also the remarkable accuracy in the computation of landmark locations and their stability between consecutive frames without any face tracking implemented.

Figure 4.1: Simultaneous head pose, landmark detection and their visibility predictions processing two representative videos from 300VW [96]. Green and red points show visible and non-visible shape predictions respectively.

The chapter is structured as follows. In Section 4.1 we first review previous literature describing works sharing knowledge among different *facial analysis tasks* through transfer learning and MTL strategies. In Section 4.2 we introduce a hybrid two-stage cascade model, termed MNN+OERT, following the framework presented in Section 3.4. In our approach, a MNN estimates both rigid face deformation and landmarks visibilities that we use to provide a better initialization and improve the robustness of the coarse-to-fine ERT under occlusions. Finally, in Section 4.3 we analyze the performance of our proposal compared to the state-of-the-art results.

4.1 Related work

We review previous literature that transfers knowledge from a source task to improve the learning capacity of one or more facial analysis tasks using both transfer learning (*i.e.*, sequential training using first a generic domain to outperform the target task optimization) and MTL (*i.e.*, training simultaneously all the tasks sharing layers). These two standard strategies have won great popularity in scenarios presenting scarce amount of annotated data [131]. In fact, these strategies are not exclusively an area of study for deep learning. Zhu *et al.* [146] (TSPM) simultaneously conduct face detection, facial landmark location and head pose estimation within a tree structure. Shen *et al.* [97] formulated face detection and face alignment together as an image retrieval problem. Wu *et al.* [121] jointly perform facial landmark detection and facial action unit recognition within a CSR framework. Wu *et al.* [119] proposed a unified CSR for head pose estimation, facial landmark location and their visibility estimation by mapping incrementally image features to a valid face shape configuration. This mapping is expressed in terms of shape-indexed SIFT descriptors [71], similar to [124]. Feng *et al.* [35] also learned jointly both face detection and face alignment but using a CSR framework that requires a head pose normalization stage.

Among the deep learning approaches, we can categorize two main subgroups according to whether the weights are fine-tuned from a pre-trained model, or simultaneously learned sharing common information with similar tasks:

- **Transfer learning based methods** fine-tune the weights from a pre-trained model used as starting point for the training process. These approaches modify the weights of all learnable layers to solve a single task. Liu *et al.* [69] perform facial attribute classification and discovered that by fine-tuning a model pre-trained in a more difficult task such as the facial identification problem, many network units are implicitly discovering the presences of some attributes (*e.g.*, gender, age, face shape or wearing glasses). Additionally, they discovered that by training multiple attribute classifiers, early units on the network act as a face detector, whose precision increases according to the number of attributes learnt.

  It is worth noticing that ImageNet has been a critical auxiliary task for the computer vision field to progress. Zhou *et al.* [142] presented a pose-invariant model based on VGG-16 [98] and pre-trained on ImageNet to demonstrate that networks initialized using transferred weights resulted in better generalization error than those trained from random weight initialization [131]. We have also presented in Chapter 2 other methods that perform head pose estimation fine-tuning from ImageNet [92, 51, 68]. In fact, in most facial analysis problems such as [39], it is common to fine-tune from VGG-Face [83], which has been trained on an artificial data set of 2.6 million faces to classify each face image and recognize which person it is.

- **Multi-task learning based methods** train a model using two or more losses to learn different related tasks together. Compared to transfer learning, MTL shares the weights of multiple learnable layers among tasks. In the facial analysis field, one of these tasks sometimes focuses on the reduction of the facial appearance variability due to extreme head pose or exaggerated facial expressions, *i.e.*, rigid pose estimation utilized as a pre-processing step [52]). Here, we organize these methods according to whether they explicitly incorporate a loss to infer the rigid face deformation during their optimization.

On the one hand, the majority of methods that avoid the head pose estimation task can do this because they have alternatively added the facial landmark detection task to manage both rigid and non-rigid deformation. Zhang *et al.* [141] (TCDCN) proposed a multi-task solution to deal with face alignment and recognize heterogeneous but correlated facial attributes (*i.e.*, gender, expression and appearance attributes) at the same time. They learn the correlations between tasks and use a probabilistic classifier to stop learning attributes that are harmful for the main facial landmark detection problem. Bulat *et al.* [15] enhance low quality face images and accurately locate landmarks on such poor resolution images. Bhagavatula *et al.* [9] (3DSTN) and Feng *et al.* [34] (PRN) analyzed how to combine face reconstruction and face alignment tasks, and evaluate two CNNs (AlexNet [59] and VGG-16 [98] fine-tuned from ImageNet weights) and an encoder-decoder to this end. Differently, Hand *et al.* [44] and Han *et al.* [43] share layers in a multi-task network among related facial attributes (*e.g.*, race and hair colour, gender and wearing make-up, etc.) to improve their classification using a MTL scheme without any face alignment task.

On the other hand, other methods explicitly include the head pose estimation task to simultaneously train the rigid deformation along with its facial analysis objective. Zhou *et al.* [142] performed action units classification by adding also the head pose estimation task, which share bottom layers of a CNN. Ranjan *et al.* [87] (AIO) study how to solve face detection, landmarks, head pose, smile classification, gender, age and identity estimation, all of them integrated into a single CNN. Kumar *et al.* [60] (Kepler) proposed an architecture based on GoogLeNet [102] to jointly train facial landmark detection, their visibility estimation and head pose. They achieved good results by initially training the model using only the most challenging faces and then fine-tuning with the whole training data set. Zhang *et al.* [140] (CCR) designed a network consisting of two modules, one responsible for head pose estimation and the other for facial landmark localization. As a result, a final unification layer allows them to communicate both modules. Finally, Ranjan *et al.* [86] proposed two models based on the popular AlexNet [59] (HyperFace) and ResNet-101 [49] (HF-ResNet) to get face detection, landmarks, their visibility, head pose estimation and gender recognition fusing the intermediate layers into a single fully connected layer followed by a loss function per task.

It is important to realize that both categories are not mutually exclusive, which means that some approaches would transfer learning from a pre-trained model to provide a better initialization, and then, perform MTL to simultaneously learn several related tasks.

The main contribution of this chapter is based on this strategy. We propose a novel procedure that combines both transfer learning and MTL to simultaneously infer various tasks: head pose estimation, facial landmark location and their visibilities. We assume that the most challenging task (*i.e.*, facial landmark detection) would be helpful to further improve the performance of the other two problems. To this end, initially, we train from scratch our model for facial landmark detection, and then, we fine-tune previous weights to perform MTL including the other two facial tasks. As a result, we maintain the precision obtained by training the landmarks detector individually, but we outperform head pose and visibility estimations. This helps the subsequent ERT to achieve better performance. It is also worth noticing that we fine-tune the weights generated using always the same database, because our goal is to make fair comparisons with previous literature that does not train using any additional data set.

4.2 Multi-task head pose and landmark estimation under occlusion

In this section, we introduce a final contribution to the face alignment field, where we propose a CNN and a coarse-to-fine ERT (similar hybrid strategy to the one in Section 3.4) to estimate the rigid and non-rigid face deformation. However, in this case, we adopt both transfer learning and MTL strategies to leverage on the strong dependencies among our tasks (*i.e.*, rigid pose estimation, facial landmark detection and their visibility estimation) to train a network much better than the one we otherwise achieve independently [136]. Our full framework, termed MNN+OERT, consists of two different regression stages. A Multi-task Neural Network (MNN) that computes the rigid face deformation, heatmaps and visibilities, and an occlusion-aware ERT (OERT) that estimates the remaining non-rigid face deformation using the predicted heatmaps of the visible landmarks.

The MTL approach applied to facial attributes classification has proved its benefits [86, 141, 119]. At this point, we prove that head pose and visibility tasks greatly benefit when considered in parity with the location of landmarks in the MNN. Our solution is different from these works in various aspects. First, like [60], we include a landmarks visibility task, that is strongly correlated to the head orientation and the landmark location, which also supports the OERT performance. The second difference is the architecture of our multi-task setting and the way we train it. We use an encoder-decoder architecture with the losses located at different positions in the model according to the task type (see Fig. 4.2).

The head pose estimation, and its projection using a 3D mean face shape, are both attached to the bottleneck layer (holistic tasks), whereas the losses of heatmaps, $\mathbf{h}$, and visibility tasks, $\mathbf{v}$, that are spatially related, are attached to the decoder end of the MNN. Moreover, the way in which we train these tasks is also very relevant. We fine-tune the weights of a model pre-trained in a single task (landmark detection), instead of performing MTL from scratch.

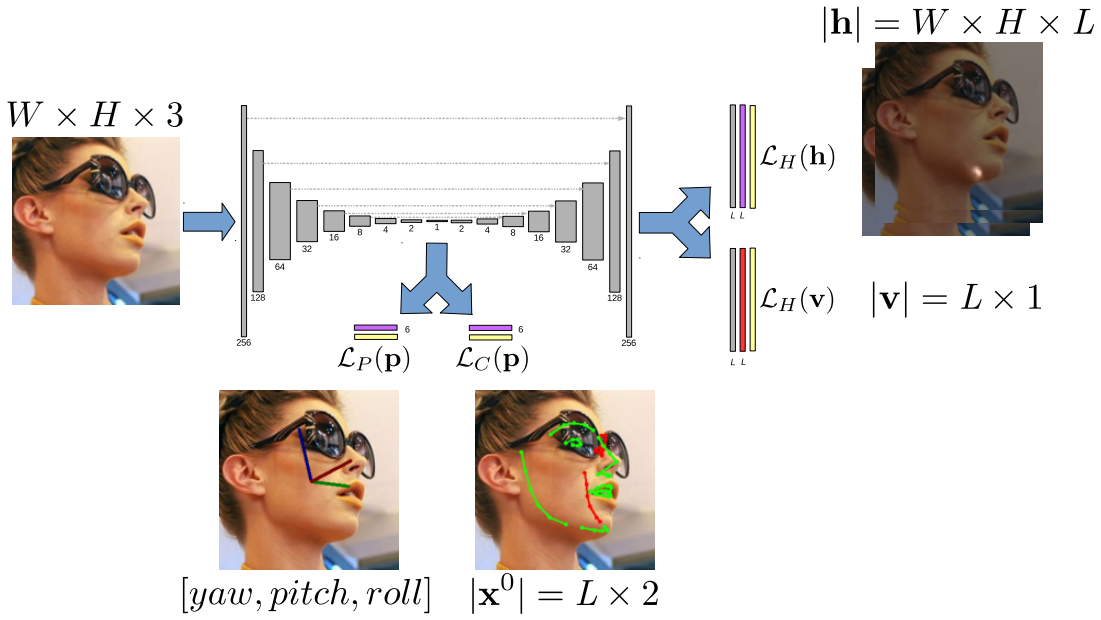


Figure 4.2: Multi-task encoder-decoder for simultaneous estimation of the rigid pose, facial landmark location and their visibility prediction. We compute the rigid deformation at the bottleneck layer, whereas we extract the non-rigid face deformation at the end. Purple, red and yellow layers represent fully connected, pooling and loss layers respectively.

The coarse-to-fine ERT presented in Section 3.2 has also proved its efficiency and great accuracy when it is properly initialized. However, we have noticed that in the challenging 3D data sets (*e.g.*, Menpo-3D [133], 300W-LP [145], AFLW2000-3D [145], LS3D-W [14]) the predictions estimated by trees associated to occluded or self-occluded landmarks are not reliable. For instance, 2D data sets locate landmarks on the face contour along the visible face edge, whereas 3D data sets make them consistent with a 3D shape model (see Fig. 4.6). So, we substitute $g(f)$, in Algorithm 5, with $g(f, \mathbf{v})$ to discard the predictions of regression trees trained using a non-visible landmark. Our OERT approach combines the landmark heatmaps, $\mathbf{h}$, the face initialization, $\mathbf{x}^0$, and their visibilities, $\mathbf{v}$, all of them computed at the MNN, to regress a refined set of facial landmarks that implicitly satisfy a valid shape model (see Fig. 4.3). Note also that, unlike previous regressors that estimate the visibility label in the cascade [16, 119], our approach estimates $\mathbf{v}$ in the MNN.

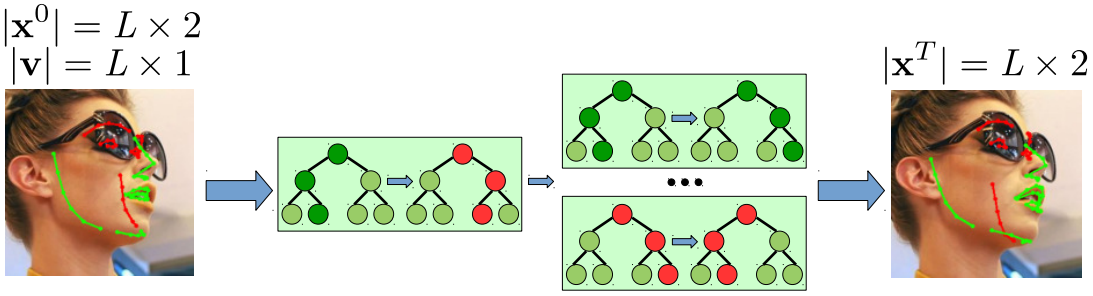


Figure 4.3: Occlusion-aware ERT receives the projected landmarks of a 3D face model and the values of their visibilities, $\mathbf{v}$, as initialization. It incrementally updates the landmark location, $\mathbf{x}^{t-1}$, discarding the predictions of those decision trees $g(f, \mathbf{v})$ whose features, f, are extracted around occluded landmarks, shown in red.

4.2.1 Multi-task Neural Network

As we highlighted in Section 3.4.1, a single heatmap regressor module based on current encoder-decoders [90, 50, 129] is good enough to provide a rough estimation of landmark location as heatmaps. This is due to its outstanding results in tasks that require local and global features at different scales. At this point, we evolve our network to simultaneously estimate head pose, landmark location and visibility into the same model (see Fig. 4.2).

Compared to standard single-task learning, MTL has to deal with two key challenges: how to design an architecture that favours the exchange of information between correlated tasks, and how to balance multiple loss functions for these tasks without allowing easier tasks to dominate.

We utilize a model termed Multi-task Neural Network (MNN), formed by an encoder-decoder network similar to the one presented in Fig. 3.8. In this case, it consists of 9 levels, reducing the spatial extent of the input face image from 256×256 to 1×1 pixels. Following Section 3.4.1, we double the number of feature maps from 64 and up to 256, when the spatial resolution is halved. However, MNN introduces short skip connections using bottleneck residual blocks [49] (see Fig. 2.4b), instead of simple convolutional layers. As detailed in Table 4.2, we also include lateral skip connections that preserve the spatial information, but we do not display them in Figs. 4.4 and 4.5 for the sake of simplicity. The residual block lets us reduce the number of parameters and increase the depth preserving the gradient through skip connections. Since it requires that the input and the output

of each block have the same dimensions, we propose additional 1×1 convolutions to reduce the number of feature maps accordingly. We show in Figs. 4.4 and 4.5 the MNN encoder and decoder architectures that we describe in Tables 4.1 and 4.2. We also include BatchNormalization and ReLu after each convolutional layer.

Name	Layer	Output	Connected to
input	InputLayer	(256, 256, 3)	
conv_9_1	Conv2D (1x1)	(256, 256, 64)	input
conv_9_2	Conv2D (1x1)	(256, 256, 64)	conv_9_1
conv_9_3	Conv2D (3x3)	(256, 256, 64)	conv_9_2
conv_9_4	Conv2D (1x1)	(256, 256, 64)	conv_9_3
add_9_4	Add	(256, 256, 64)	conv_9_1, conv_9_4
conv_8_1	Conv2D (2x2)	(128, 128, 128)	conv_9_4
conv_8_2	Conv2D (1x1)	(128, 128, 64)	conv_8_1
conv_8_3	Conv2D (3x3)	(128, 128, 64)	conv_8_2
conv_8_4	Conv2D (1x1)	(128, 128, 128)	conv_8_3
add_8_4	Add	(128, 128, 128)	conv_8_1, conv_8_4
conv_7_1	Conv2D (2x2)	(64, 64, 256)	conv_8_4
conv_7_2	Conv2D (1x1)	(64, 64, 64)	conv_7_1
conv_7_3	Conv2D (3x3)	(64, 64, 64)	conv_7_2
conv_7_4	Conv2D (1x1)	(64, 64, 256)	conv_7_3
add_7_4	Add	(64, 64, 256)	conv_7_1, conv_7_4
...	...	...	...
conv_2_1	Conv2D (2x2)	(2, 2, 256)	conv_3_4
conv_2_2	Conv2D (1x1)	(2, 2, 64)	conv_2_1
conv_2_3	Conv2D (3x3)	(2, 2, 64)	conv_2_2
conv_2_4	Conv2D (1x1)	(2, 2, 256)	conv_2_3
add_2_4	Add	(2, 2, 256)	conv_2_1, conv_2_4
pool_1_1	MaxPooling (2x2)	(1, 1, 256)	conv_2_4
conv_1_2	Conv2D (1x1)	(1, 1, 64)	conv_1_1
conv_1_3	Conv2D (3x3)	(1, 1, 64)	conv_1_2
conv_1_4	Conv2D (1x1)	(1, 1, 256)	conv_1_3
add_1_4	Add	(1, 1, 256)	pool_1_1, conv_1_4

Table 4.1: MNN encoder architecture.

Name	Layer	Output	Connected to
up_1_5	UpSampling (2x2)	(2, 2, 256)	add_1_4
concat_2_5	Concatenate	(2, 2, 512)	add_2_4, up_1_5
conv_2_6	Conv2D (1x1)	(2, 2, 256)	concat_2_5
conv_2_7	Conv2D (1x1)	(2, 2, 64)	conv_2_6
conv_2_8	Conv2D (3x3)	(2, 2, 64)	conv_2_7
conv_2_9	Conv2D (1x1)	(2, 2, 256)	conv_2_8
add_2_9	Add	(2, 2, 256)	conv_2_6, conv_2_9
conv_2_10	Conv2DTrans (2x2)	(4, 4, 256)	add_2_9
...	...	...	...
concat_7_5	Concatenate	(64, 64, 512)	add_7_4, conv_6_10
conv_7_6	Conv2D (1x1)	(64, 64, 256)	concat_7_5
conv_7_7	Conv2D (1x1)	(64, 64, 64)	conv_7_6
conv_7_8	Conv2D (3x3)	(64, 64, 64)	conv_7_7
conv_7_9	Conv2D (1x1)	(64, 64, 256)	conv_7_8
add_7_9	Add	(64, 64, 256)	conv_7_6, conv_7_9
conv_7_10	Conv2DTrans (2x2)	(128, 128, 128)	add_7_9
concat_8_5	Concatenate	(128, 128, 256)	add_8_4, conv_7_10
conv_8_6	Conv2D (1x1)	(128, 128, 128)	concat_8_5
conv_8_7	Conv2D (1x1)	(128, 128, 64)	conv_8_6
conv_8_8	Conv2D (3x3)	(128, 128, 64)	conv_8_7
conv_8_9	Conv2D (1x1)	(128, 128, 128)	conv_8_8
add_8_9	Add	(128, 128, 128)	conv_8_6, conv_8_9
conv_8_10	Conv2DTrans (2x2)	(256, 256, 64)	add_8_9
concat_9_5	Concatenate	(256, 256, 128)	add_9_4, conv_8_10
conv_9_6	Conv2D (1x1)	(256, 256, 64)	concat_9_5
conv_9_7	Conv2D (1x1)	(256, 256, 64)	conv_9_6
conv_9_8	Conv2D (3x3)	(256, 256, 64)	conv_9_7
conv_9_9	Conv2D (1x1)	(256, 256, 64)	conv_9_8
add_9_9	Add	(256, 256, 64)	conv_9_6, conv_9_9

Table 4.2: MNN decoder architecture.

Henceforth, we analyze how to integrate MTL into the encoder-decoder architecture. Our proposal encourages the encoder to learn a holistic face representation, sharing features that favour the exchange of information among all the correlated tasks, and reducing the risk of overfitting to one task. The decoder specifically learns local features tailored to the estimation of non-rigid landmark location and their visibilities. Here, it is important to describe the losses that we have used for each task, and how we weight them to treat all tasks with the same importance.

Holistic attributes

The head pose is a global attribute, hence it is computed from the 1×1 feature map at the end of the encoder (`add_1_4` in Table 4.1). We show in Fig. 4.4 the MNN encoder that computes the rigid pose at the end. The importance of where to locate each loss function is essential in MTL to share common features between tasks, without decreasing the learning capacity of our model, as we will analyze in Section 4.3.3.

Our main objective in the encoder is to estimate the parameters of the rigid transformation of the head, $\mathbf{p}$, consisting of 6 parameters, *i.e.*, the 3D Euler angles (see Fig. 2.1) and the 3D translation respectively (yaw, $pitch$, $roll$, t_x, t_y, t_z). To this end, we employ two different losses, $\mathcal{L}_P$ and $\mathcal{L}_C$, previously introduced in Eqs. 2.3 and 3.6. We include two separate fully connected layers with 6 outputs at the end of the encoder (see Fig. 4.2).

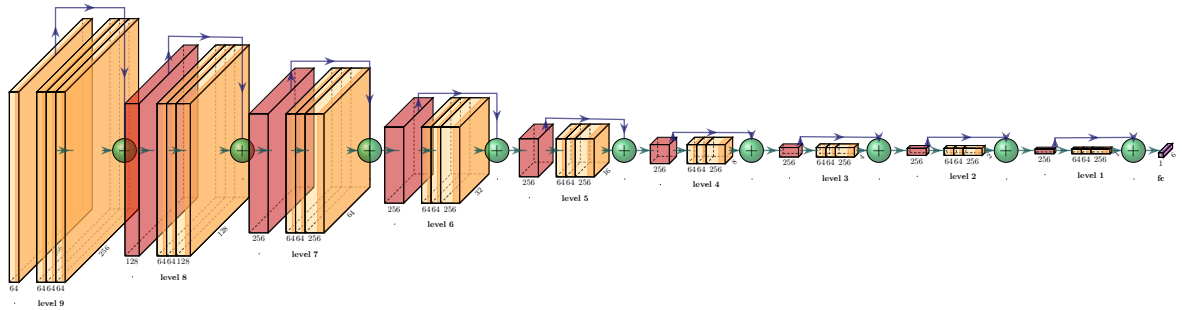


Figure 4.4: MNN encoder diagram detailed in Table 4.1. We introduce bottleneck residual blocks in each level [49]. We estimate the rigid pose parameters at the end of the encoder, using a fully connected layer with 6 outputs (yaw, $pitch$, $roll$, t_x, t_y, t_z).

They optimize this rigid pose parameters, $\mathbf{p}$, using two different loss functions. The former, $\mathcal{L}_P(\mathbf{p})$ directly minimizes the error of $\mathbf{p}$. The latter, $\mathcal{L}_C(\mathbf{p})$ optimizes the alignment error produced by the landmarks rigid projection, $\mathbf{x}_i = \pi(\mathbf{p}_i, \mathbf{X}^l)$. We represent it by,

$$\mathcal{L}_C(\mathbf{p}) = \sum_{i=1}^{N} \left(\sum_{l=1}^{L} \left(\frac{\tilde{\mathbf{w}}_i^l}{||\tilde{\mathbf{w}}_i||_1} \cdot ||\tilde{\mathbf{x}}_i^l - \pi(\mathbf{p}_i, \mathbf{X}^l)|| \right) \right), \tag{4.1}$$

where $\tilde{\mathbf{x}}_i^l \in \mathbb{R}^{L\times 2}$ are the l-th landmark ground truth coordinates for the i-th training image, $\mathbf{X} \in \mathbb{R}^{L\times 3}$ represent the 3D coordinates of the L landmarks on the mean 3D head model, and π is the projection function obtained with the rigid parameters $\mathbf{p}_i$.

$\mathcal{L}_C(\mathbf{p})$ provides an accurate initialization so that the OERT stage successfully fits the remaining non-rigid deformation. Compared to 3DDE [113] (see Section 3.4), now each initialization, $\mathbf{x}^0$, has been further improved by directly estimating the mean 3D shape projection in the network, instead of using the heatmaps, $\mathbf{h}$, to generate it. However, as we use the landmarks annotations as ground truth $\tilde{\mathbf{x}}_i$, there is an intrinsic error produced by comparing a rigid projection with the ground truth non-rigid landmarks. It is worth mentioning that the head pose angles (yaw, $pitch$, $roll$) associated to previous projection matrix, π, are slightly worse than the ones obtained by directly minimizing their error using $\mathcal{L}_P(\mathbf{p})$. We consider that this is due to the comparison between the rigid landmarks projection using the mean 3D head model and the non-rigid landmarks annotations. For this reason, we also use $\mathcal{L}_P(\mathbf{p})$ to evaluate MNN, in the head pose benchmark presented in Chapter 2.

Local attributes

Facial landmark detection and their visibility estimation require global and abstract features and a fine spatial resolution. Therefore, we employ the features maps at the end of the MNN decoder to learn these local attributes (`add_9_9` in Table 4.2). We show in Fig. 4.5 the architecture of our MNN decoder, which computes the tasks associated to the position of each landmark. To this end, following Section 3.3, we employ the cross-entropy loss function $\mathcal{L}_H$ introduced in Eq. 3.5 for both landmarks tasks. From now on, we denote as $\mathcal{L}_H(\mathbf{h})$ and $\mathcal{L}_H(\mathbf{v})$ the heatmap and visibility losses respectively.

On the one hand, we perform the landmark detection by including a convolutional layer to produce $[256 \times 256 \times L]$ feature maps and a softmax activation layer to generate heatmaps, $\sum_i^P \mathbf{h}_i^l = 1$. On the other hand, we reduce the resolution of the feature maps

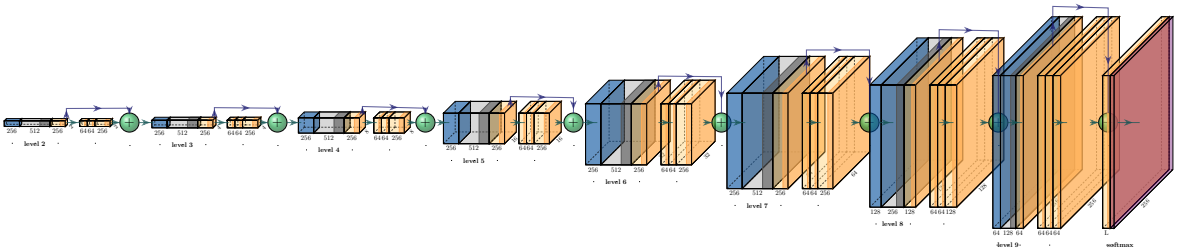


Figure 4.5: MNN decoder diagram detailed in Table 4.2. We introduce bottleneck residual blocks in each level [49]. We estimate the landmarks tasks at the end of the decoder.

from $[256 \times 256 \times L]$ to $[1 \times 1 \times L]$ using a pooling layer with kernel size 256×256. Thus, we generate the vector of L visibilities associated to the L landmarks, $\mathbf{v}$, (see Fig. 4.2).

We adopt the cross-entropy loss, $\mathcal{L}_H$ presented in Eq. 3.5 for both tasks, where the number of pixels to evaluate, P, is different for each task, *i.e.*, $P = 256 \times 256$ for heatmaps, $\mathcal{L}_H(\mathbf{h})$, whereas $P = 2$ for visibility, $\mathcal{L}_H(\mathbf{v})$. In the visibility case, we represent the cross-entropy loss by,

$$\mathcal{L}_H(\mathbf{v}) = \sum_{i=1}^{N} \left(\sum_{p=1}^{P} (-\tilde{\mathbf{v}}_i(p) \cdot \log(\mathbf{v}_i(p)) \right), \tag{4.2}$$

where N is the number of training images and P is the number of classes.

Multi-task loss and training scheme

We compute the multi-task loss by adding all previous losses and empirically tuning weights α_{proj}, α_{pose}, α_{lnd} and α_{vis} to balance the importance of all tasks. The total loss function $\mathcal{L}$ computes a global error obtained from the rigid transformation parameters, $\mathbf{p}$, the projected landmarks using the 3D mean shape, $\mathbf{x}$, the heatmaps, $\mathbf{h}$, and the visibilities, $\mathbf{v}$, by combining them using a weighted sum of the losses,

$$\mathcal{L}(\mathbf{p}, \mathbf{h}, \mathbf{v}) = \alpha_{proj} \cdot \mathcal{L}_C(\mathbf{p}) + \alpha_{pose} \cdot \mathcal{L}_P(\mathbf{p}) + \alpha_{lnd} \cdot \mathcal{L}_H(\mathbf{h}) + \alpha_{vis} \cdot \mathcal{L}_H(\mathbf{v}). \tag{4.3}$$

These weight parameters α are expensive to tune. It is noticeable that when different loss magnitudes exist (*e.g.*, L_2 together with cross-entropy), higher loss magnitude tasks will factor disproportionately high weight in the loss. For this reason, we train individually each task to determine each loss magnitude when the learning process converges, and ponderate them accordingly. Moreover, it is important to balance these losses without allowing easier tasks to dominate.

We achieve the best results (see Fig. 4.9) when training our model first optimizing the landmark task, $\mathcal{L}_H(\mathbf{h})$, and then, with the rigid projection, head pose and visibility tasks using the loss presented in Eq. 4.3. Training first with the most difficult task regularizes the optimization of the others. This is not surprising since the landmarks visibility can only be established when we know its location.

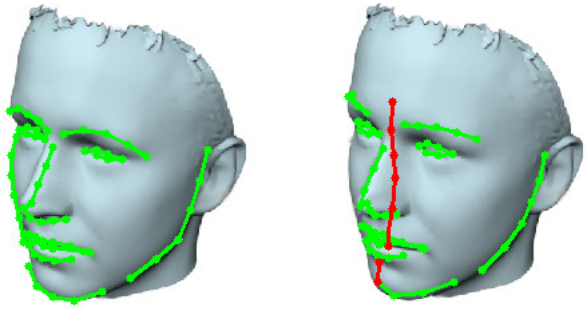
4.2.2 Occlusion-aware ERT

Following the hybrid design presented in Section 3.4 [112, 113], our full MNN+OERT proposal consists of two stages, being the second a coarse-to-fine ERT (see Section 3.2).

ERT regressors have proven to be very efficient and provide accurate results that preserve a valid face shape, but they require a good initialization to converge to the solution.

On the one hand, we employ as initialization, $\mathbf{x}^0$, the projection of a 3D mean shape with the estimated rigid transformation parameters optimized in the MNN using the loss $\mathcal{L}_C(\mathbf{p})$ (see Fig. 4.2). It provides our OERT stage with an excellent initial estimation of the orientation and position of the target face, which solves the rigid pose estimation of the head.

On the other hand, this initialization, $\mathbf{x}^0$, also allows us to infer whether the predicted initial landmarks are self-occluded or not, which is essential in recent 3D face alignment data sets (*e.g.*, Menpo-3D [133], 300W-LP [145], AFLW2000-3D [145], LS3D-W [14]). In these 3D benchmarks, landmarks along the face contour do not successfully fit the visible face edge, as it happens with previous 2D data sets (see Fig. 4.6).



(a) 2D face alignment (b) 3D face alignment

Figure 4.6: Comparison of landmark annotations along the face contour between 2D and 3D views. Red coloured landmarks show self-occluded parts.

In Fig. 4.7 we show the 3D initialization achieved in some challenging faces with large head poses, partial occlusions and extreme non-rigid face deformation due to exaggerated expressions. As can be noticed, the initialization provided by MNN is always quite accurate. In addition, we achieve the self-occlusion visibility prediction, computed according to this initialization, $\mathbf{x}^0$. It is visually noticeable that our MNN initializations are robust against multiple nuisance factors, whereas the initial shapes produced by 3DDE [113] (see Fig. 3.12) are more dependent on the heatmaps performance.

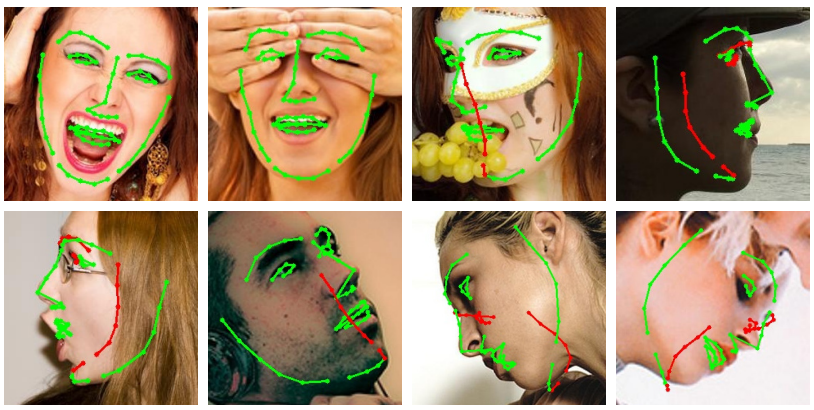


Figure 4.7: Representative initial shapes produced by MNN in AFLW-2000-3D data set. Visible/self-occluded landmarks in green/red colour.

Once the rigid 3D pose is accounted for, the OERT estimates the remaining non-rigid face deformation from the estimated heatmaps of the visible landmarks. Thus, we discard the regression produced by those decision trees that extract features around non-visible landmarks (*i.e.*, the position of a non-visible landmark can be estimated using other visible neighbours). Although there is only one public data set labelled with landmark occlusions, COFW [16], we generate synthetic occlusions following the same procedure introduced in Fig. 3.14 for all data sets, labelling as occluded those landmarks covered by the synthetic occlusion. Moreover, it is also easy to infer self-occlusions in recent 3D data sets.

To better handle occlusions, we incorporate these visibility labels, $\mathbf{v}$, into the coarse-to-fine ERT strategy described in Algorithms 4 and 5, where we progressively refine the current shape prediction, $\mathbf{x}^{t-1}$, with the prediction of K decision trees, g. We represent each stage of our OERT regressor as $\mathcal{C}(f, \mathbf{v}) = \sum_{k=1}^{K} g_k(f, \mathbf{v})$. As a result, we ignore the displacement obtained from decision trees $g_k(f, \mathbf{v})$ whose features f have been extracted around occluded or self-occluded landmarks, *i.e.*, the average residual estimated by trees whose associated landmark is occluded, is not added to the final estimation (see Fig. 4.3). Compared to previous coarse-to-fine ERT (see Section 3.2), the OERT regressor does not estimate visibilities, but it rather uses the predictions provided by the MNN.

4.3 Experiments

In this section, we revisit the performance of previous approaches focused on head pose estimation and facial landmark location problems. We compare the results obtained with the solutions presented on Chapters 2 and 3 [111, 3, 70, 112, 110, 113] and the new MNN+OERT, using the standard evaluation metrics and benchmarks. We also evaluate the importance of each contribution in MNN+OERT and determine its performance using recent data sets that provide information about the location of 3D landmarks projected onto the face image plane, also termed 3D face alignment.

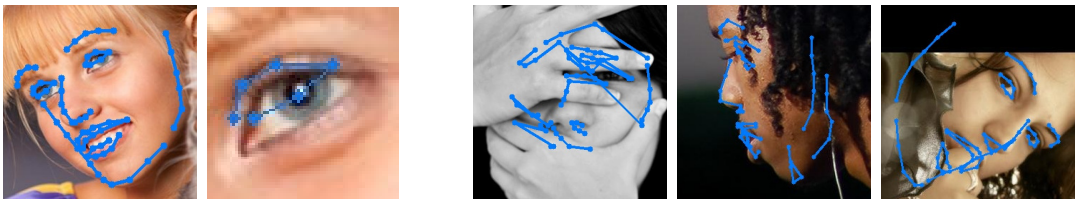
4.3.1 Database

We evaluate our three main problems, head pose estimation, facial landmark location and their visibility prediction, using both 2D and 3D face alignment data sets. Previous standard data sets, introduced in Sections 2.4.1 and 3.5.1 (*i.e.*, 300W, COFW, WFLW, AFLW), provide head pose and 2D landmark annotations that correspond to semantically meaningful parts of the face. Nevertheless, there is an increasing amount of work that performs experiments using in-the-wild data sets annotated with 3D landmark, modelling the projections of the 3D head model (see differences in Fig. 4.6). The main problem providing an accurate 3D ground truth seems the requirement of a faithful face reconstruction step, which still remains very challenging under in-the-wild conditions (see Fig. 4.8a). In the experiments, we evaluate MNN+OERT using previous 2D data sets and the following 3D benchmarks:

- AFLW2000-3D [145] provides 2000 re-annotated faces from AFLW [56] using 68 3D landmarks projected onto the image plane, and yaw, pitch and roll angles obtained assuming the structure of a mean 3D face shape model using the 3D facial landmarks ground truth. It has been extensively utilized for testing purposes. We also divide

AFLW into three intervals of [0°, 30°], [30°, 60°] and [60°, 90°] according to the head absolute yaw angle. Each interval consists of 1306, 462 and 232 faces respectively.

Bulat *et al.* [14] newly re-annotate this data set because they claim that the original face reconstruction algorithm used as ground truth annotation, is prone to alignment errors, but even so, we note in Fig. 4.8 that a few re-annotations are still incorrect.



(a) Inaccurate landmark location (b) Annotation errors produced using [14]

Figure 4.8: Representative ground truth landmarks from AFLW2000-3D. (a) Inaccurate 3D landmark projections. (b) Annotation errors related to incorrect face reconstruction.

- 300W-LP [145] consists of a collection of 61225 synthetic faces obtained by rendering 300W [93] across extreme head rotations, ranging from -90° to 90°. It provides 68 3D landmarks automatically annotated in a consistent manner with AFLW2000-3D [145]. It has been widely used as training set to evaluate AFLW2000-3D [145]. Bulat *et al.* [14] also re-annotated this data set.

- LS3D-W [14] consists of 7200 faces automatically annotated with 68 3D landmarks. It is proposed as an alternative to 300W-LP [145] for training purposes. This data set has been balanced using the same number of images for each absolute yaw range [0°, 30°], [30°, 60°] and [60°, 90°].

- Menpo [134] and Menpo-3D [133] represent two private competitions based on 2D and 3D face alignment tasks respectively. Menpo consists of 8979 training and 7276 testing images divided into different subsets: *semifrontal* and *profile*, which provide 68 and 39 2D landmarks configurations. It presents some landmarks inconsistencies between both subsets. Otherwise, Menpo-3D provides a collection of 67887 training faces (13391 images and 54496 frames from 300VW [96]) and 11100 testing faces (111 videos with 1000 frames each one) labelled with 84 3D landmarks, but without face bounding box annotations. In this case, the annotations have not been released yet, but we show qualitative face alignment results in Fig. 4.13f.

4.3.2 Implementation details

To train MNN+OERT, we shuffle the training set of each database and split it into 90% train and 10% validation subsets. Here, we always select the model parameters with lowest validation error. We have trained an individual model for each database using the training/testing configuration presented in Section 4.3.1. All the experiments follow the settings detailed below.

MNN+OERT consists of two key stages. The first stage focuses on training the MNN. We train it from scratch until validation convergence, by using only the landmark location task. Then, we refine these initial weights by including the remaining tasks, following a MTL strategy. It is also important to note that we do not include any additional training

database during the transfer learning step. We basically use the same training parameters presented in Section 3.5.3, being the cropped input face reduced from 256×256 to 1×1 pixels by gradually halving their size across 9 levels. At this point, the learning rate is halved every 15 epochs without any improvement, *i.e.*, we give more patience to provide robustness against fluctuations produced by using a lower batch size due to its deeper design. In the MNN we also introduce bottleneck residual modules [49] (see Tables 4.1 and 4.2) to preserve the gradient through skip connections.

The second stage focuses on the OERT configuration. We basically employ the same training parameters presented in Section 3.5.3, but in this case, we increase the maximum number of ERT stages to $T = 50$ to compensate the missing displacement of those decision trees whose features have been extracted around a non-visible landmark. We also set the FREAK pattern diameter to be progressively reduced an overall 20% along the OERT stages (see Fig. 3.5a). Here, we avoid the Gaussian filter σ and RANSAC+POSIT procedure presented in Algorithm 6.

Training the full MNN+OERT framework takes 95 hours for WFLW (79 hours fine-tuning the MNN and 16 hours for the OERT) using a NVidia GeForce GTX 1080Ti GPU (11GB) with a batch size of 7 images, and a dual Intel Xeon Silver 4114 CPU at 2.20GHz (2×10 cores/40 threads, 128 GB). At run-time our hybrid method requires on average 78 ms to process each face, where the MNN takes 66 ms and the OERT 12 ms. It achieves an overall frame rate of 12.8 FPS, which is a 22% faster than CHR2C [110], using C++, Tensorflow and OpenCV libraries.

4.3.3 Ablation study

In this section, we analyze the importance of each contribution in the MNN+OERT framework to determine its effect in the overall behaviour. As we mentioned previously, we consider that head pose estimation, landmark detection and their visibility prediction are not completely solved under realistic scenarios. Therefore, we propose a MTL design that outperforms the individual performance of several tasks and generates a face shape initialization better than our previous rigid pose estimation proposals, and a new OERT regressor robust against occlusions.

In Fig. 4.9, we show how much does it help, first training the MNN to perform landmark detection, compared to training simultaneously all tasks from scratch. The orange and blue validation learning curves represent the overall loss detailed in Eq. 4.3 obtained by training our model from scratch ($\mathcal{L} = 8.69$) against, respectively, first optimizing the model using the landmarks task ($\mathcal{L} = 7.93$). In this case, we obtain a loss reduction of 8.7% training first with the most difficult task, which regularizes the optimization of the others. In fact, if we analyze the percentage of loss reduction associated to the head pose estimation, $\mathcal{L}_P$, 9.67%, and the projection of the 3D mean shape, $\mathcal{L}_C$, 10.21%, we reach the conclusion that both rigid estimation tasks clearly benefits from feature maps learned with landmarks as initialization.

Next, we present an ablation study for each task to evaluate how each contribution affects performance.

Head pose

In Fig. 4.9, we also evaluate the importance of locating each loss related to the rigid pose estimation task at the MNN bottleneck layer. The green and blue curves plot the difference between fine-tuning MNN from landmarks with both rigid pose losses, $\mathcal{L}_P$ and

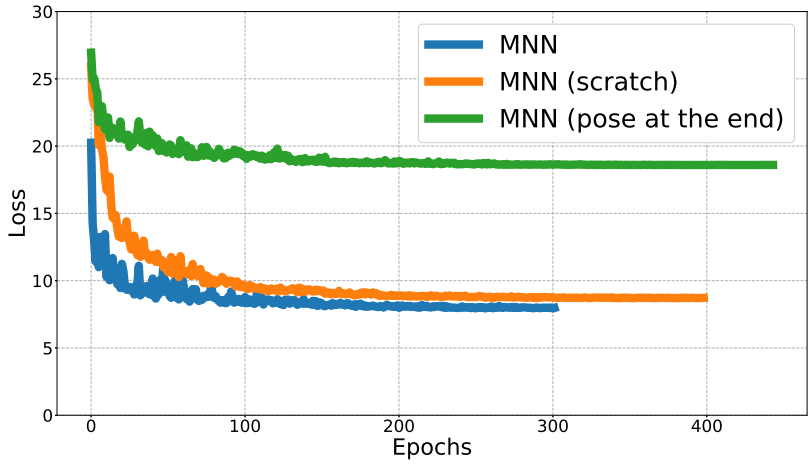


Figure 4.9: Validation learning curves in WFLW test set. Blue, orange and green colours compare the overall loss obtained with MNN by fine-tuning from landmarks, training from scratch, and locating the rigid pose losses at the end of the decoder respectively.

$\mathcal{L}_C$, located at the end of the decoder ($\mathcal{L} = 18.59$) vs in the bottleneck layer at the end of the encoder ($\mathcal{L} = 7.93$), which represents a loss reduction of 57.3%. In addition to the local vs global argument, we speculate that rigid and non-rigid face deformation tasks in the same layer are in conflict.

From now on, we attach the rigid head estimation losses, $\mathcal{L}_P$ and $\mathcal{L}_C$, at the end of the encoder, since we have proved the importance of a good design to favour the exchange of information between facial tasks. In Table 4.3 we also review the performance obtained training the encoder-decoder for head pose estimation alone (*Single task* column) and the results achieved using the MTL strategy (*Multi-task* column). As a result, we obtain an average head pose *MAE* reduction of 0.3° using our MTL approach, which represents a 12.5% *MAE* reduction in the angular error. A key step for this improvement is the prior pre-training in the landmark detection, a difficult task that requires a precise localization of the landmarks in the image.

Database	Single task					Multi-task				
	yaw	pitch	roll	mean	loss	yaw	pitch	roll	mean	loss
	MAE	MAE	MAE	MAE	$\mathcal{L}_P(\mathbf{p})$	MAE	MAE	MAE	MAE	$\mathcal{L}_P(\mathbf{p})$
300W public	2.01	2.40	1.32	1.91	3.75	1.71	1.94	1.12	1.59	3.24
300W private	2.41	2.85	1.40	2.22	3.36	2.11	2.49	1.28	1.96	2.91
COFW	2.61	3.42	2.00	2.67	4.92	2.40	3.16	1.53	2.36	4.36
AFLW	4.27	3.41	2.63	3.43	6.96	4.16	3.07	2.43	3.22	6.55
WFLW	3.96	3.71	2.55	3.40	3.98	3.00	3.05	1.85	2.63	3.42
300W-LP	2.24	1.40	1.58	1.74	2.18	2.15	1.40	1.58	1.71	2.02
LS3D-W	1.67	1.40	1.33	1.46	3.47	1.45	1.19	1.16	1.26	3.03

Table 4.3: Comparison between a single task model trained from scratch and a multi-task model fine-tuned from landmarks for head pose estimation.

Facial landmarks visibility

Similarly, we test the importance of the landmark detection task for their visibility estimation. On the one hand, we train our MNN for a single task. In this case, we achieve a precision/recall of 80.27/20.70 for occlusion detection. This is a poor result in terms of recall, far worse than most other published results (see Table 4.6). On the other hand, when we use a MTL approach, we get a precision/recall of 81.93/69.75, a huge jump in recall with even better precision. The improvement achieved is reasonable, since to decide whether a landmark is occluded or not, we first require an accurate location on the image, which is provided by the pre-training in the landmark estimation task.

Facial landmark location

In this first experiment, we compare the initialization presented in Section 3.4.1, using POSIT [27] on the predicted heatmaps, instead of utilizing the rigid projection of the 3D mean shape, $\mathbf{x}^0$, estimated by MNN (see Fig. 4.2). In Fig. 4.10 we illustrate the *NME* reduction due to the new initialization obtained from MNN, compared to the one obtained in 3DDE [113]. Here, we divide the WFLW test set into ten groups according to the *NME* assigned to each initial sample using POSIT on the predicted hetmaps, like Algorithm 6. We observe that, on average, the MNN prediction reduces in about 2.04% the error of the initial rigid alignment (0.34 *NME* reduction). However, the reduction is largest in samples with wrong initializations, *e.g.*, `Gr. 9` where an improvement of 7.86% represents a 2.84 *NME* reduction. It suggests that our new MNN initialization is especially noticeable in those benchmarks with a high proportion of challenging faces.

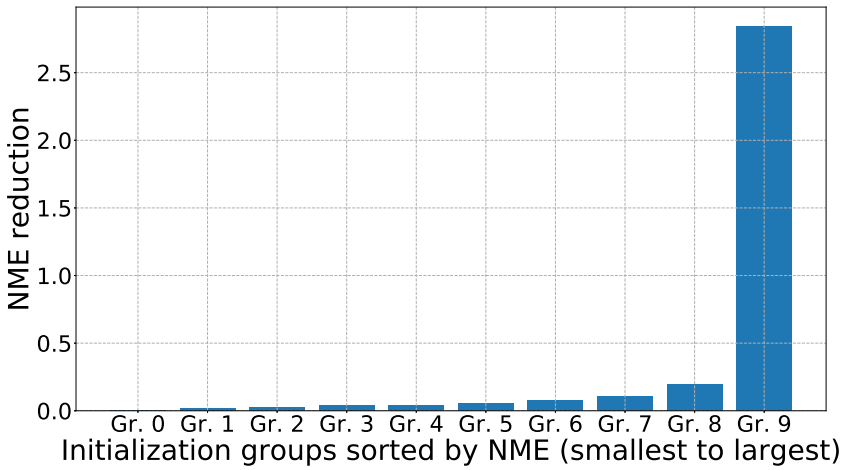


Figure 4.10: Distribution of the average *NME* reduction in the OERT initialization using WFLW. We divide it into ten groups according to the 3DDE [113] initialization error. We use the *height* as normalization for *NME*.

In the second experiment, we compare the predictions generated by our coarse-to-fine ERT (see Section 3.2) with the new OERT, which uses the visibility estimated by MNN, $\mathbf{v}$, and the self-occlusion related to the face orientation of the initialization (see Fig. 4.2). At this point, it is necessary a data set with landmarks visibility labelled, such as COFW, or a benchmark with 3D landmarks, where self-occlusion plays an important role.

Even though this strategy provides improvements in all data sets with visibility data (see Tables 4.9 and 4.12), the actual *NME* differences are washed out when averaged

over the number of landmarks in the face, and the number of face images in the test set. However, these subtle differences are visually appreciated by looking into specific occluded landmarks, such as the eyebrow/eye location improvement in Fig. 4.11a and 4.11b.

(a) ERT (b) OERT (c) Ground truth

Figure 4.11: Example of the coarse-to-fine ERT vs the OERT regressor under occlusions. (a) Predicted shape with a regressor that refines the location using features from occluded landmarks. (b) Predicted shape with our occlusion-aware approach. (c) Ground truth.

To give a clear idea of the contribution of each stage in our approach, we report the *NME* of, 1) `MNN` alone given by the location of the maximum response on each heatmap, 2) our full `MNN+OERT` framework, and 3) `GT+OERT` represents the optimal result obtained with the perfect initialization obtained by using POSIT [27] with the ground truth landmarks. We describe in Table 4.4 the performance obtained using 300W public/private, COFW and WFLW.

Method	Common pupils *NME*	Challenging pupils *NME*	Full pupils *NME*
MNN	4.29	8.23	5.06
MNN+OERT	3.72	7.26	4.41
GT+OERT	3.60	6.89	4.25

(a) 300W public

Method	Indoor corners *NME*	Outdoor corners *NME*	Full corners *NME*
MNN	4.27	4.27	4.27
MNN+OERT	3.70	3.67	3.68
GT+OERT	3.53	3.55	3.54

(b) 300W private

Method	pupils *NME*
MNN	5.65
MNN+OERT	5.04
GT+OERT	4.70

(c) COFW

Method	Full corners			Pose corners			Expression corners			Illumination corners			Make-up corners			Occlusion corners			Blur corners		
	NME	AUC_{10}	FR_{10}	*NME*	AUC_{10}	FR_{10}	*NME*	AUC_{10}	FR_{10}	*NME*	AUC_{10}	FR_{10}	*NME*	AUC_{10}	FR_{10}	*NME*	AUC_{10}	FR_{10}	*NME*	AUC_{10}	FR_{10}
MNN	6.10	47.15	10.88	11.38	17.42	40.79	6.87	42.12	12.10	6.21	48.04	9.31	5.68	49.28	11.16	7.76	37.97	19.29	7.84	38.03	16.81
MNN+OERT	4.61	56.68	4.84	8.32	28.79	22.39	5.24	51.88	6.36	4.63	57.26	4.01	4.30	58.10	3.88	5.79	48.04	9.23	5.40	50.23	6.20
GT+OERT	4.31	58.16	3.52	7.51	31.24	17.48	4.91	53.81	5.09	4.22	58.79	3.15	4.09	59.77	2.91	5.18	50.49	6.52	4.94	52.36	4.65

(d) WFLW

Table 4.4: Ablation study in 300W public, 300W private, COFW and WFLW respectively. MNN represents the detection of landmarks as the maximum response of each heatmap. MNN+OERT preserves face shape using a second OERT stage. GT+OERT presumes a perfect rigid initialization for the OERT using ground truth landmarks.

First, we prove the importance of the OERT stage to enforce the face shape. Our full `MNN+OERT` framework achieves an average reduction in *NME* of 12.8% using 300W public, 13.8% using 300W private, and 10.8% in COFW. As can be noticed, this percentage raises proportionally according to the proportion of difficult images contained in each data set. Thus, 300W private, which is the most challenging, is the one with highest improvement. The *NME* reduction becomes even more notorious in WFLW, where our two-stage method gets an improvement of 24.4% due to the additional sources of variability (*i.e.*, expressions, illumination, make-up, occlusions and blur). Most notably, `MNN+OERT` provides the largest

improvement with respect to `MNN` results in the *Occlusion*, *Pose* and *Blur* subsets, stressing again the importance of enforcing the face shape in these situations.

Finally, the results obtained using `GT+OERT` help us determine the best result we would achieve by computing the perfect rigid pose estimation using the ground truth landmarks instead of using the initialization from MNN. In this case, we obviously reach even superior precision, *i.e.*, an additional improvement rate of 5% on 300W, COFW and WFLW, which suggests that our estimation is still far from being completely solved.

4.3.4 Comparison with the state-of-the-art

In this section, we compare our MNN+OERT model with the state-of-the-art by using their published results on the head pose and facial landmarks tasks. We also organize the results obtained according to each facial task.

Head pose

In Table 4.5 we compare the head pose estimation obtained by regressing the yaw, pitch and roll angles, with the best published results in the literature, training two different MNN models for AFLW and 300W-LP. As detailed in Section 2.4.5, AFLW results are not strictly comparable since there is no standard benchmark to determine the training and testing partitions. Differently, the model trained on 300W-LP has been evaluated using AFLW2000-3D face images following the protocol in [92, 68, 51, 130].

Method	AFLW				300W-LP/AFLW2000-3D			
	yaw	pitch	roll	mean	yaw	pitch	roll	mean
	MAE	*MAE*	*MAE*	*MAE*	*MAE*	*MAE*	*MAE*	*MAE*
Kepler [60]	6.45	5.85	8.75	7.01	-	-	-	-
Hyperface [86]	7.61	6.13	3.92	5.88	-	-	-	-
HopeNet [92]	6.26	5.89	3.82	5.32	6.47	6.55	5.43	6.15
HF-ResNet[86]	6.24	5.33	3.29	4.95	-	-	-	-
GLDL [68]	6.00	5.31	3.75	5.02	3.02	5.06	3.68	3.92
CCR [140]	5.22	5.85	2.51	4.52	-	-	-	-
QuatNet [51]	**3.93**	4.31	2.59	3.61	3.97	5.61	3.92	4.50
Amador *et al.* [3]	5.59	4.79	2.83	4.40	-	-	-	-
FSA-Caps-Fusion [130]	-	-	-	-	4.50	6.08	4.64	5.07
MNN	4.16	**3.07**	**2.43**	**3.22**	**2.15**	**1.40**	**1.58**	**1.71**

Table 4.5: Error of head pose estimation methods using AFLW and 300W-LP.

We outperform the state-of-the-art in AFLW (3.22 *MAE*), which represents an average error reduction of 10.8% over QuatNet [51] (3.61 *MAE*), the best reported result. It is also worth mentioning that MNN head pose results have been obtained fine-tuning first from the landmark detection task and training finally with all tasks simultaneously. However we have just used AFLW annotations, instead of fine-tuning from an additional data set like [3, 92, 51, 68]. But even so, MNN improves by a large margin all of them. Moreover, MNN achieves a 26.8% in *MAE* reduction compared to our CNN based on RestNet-101 [3] (4.40 *MAE*) using exactly the same training/testing partition.

In the experiment performed using 300W-LP and AFLW2000-3D as train and test sets respectively, MNN sets again the state-of-the-art (1.71 *MAE*) without fine-tuning from an additional data set. It represents an impressive reduction in *MAE* of 56.3% compared to our main competitor GLDL [68] (3.92 *MAE*).

Note that the average head pose error in AFLW is the largest (see Table 4.3) probably because of the head pose annotation errors produced by POSIT [27] assuming the structure of the 3D mean face in faces with 2D landmark annotations. In addition, we have found several inaccurate landmarks with labels errors (see Fig. 4.8), which produce wrong head pose ground truth, as we describe in Section 3.5.1. Thus, some of our predictions are better than the ground truth.

Facial landmarks visibility

In Table 4.6 we compare the landmarks visibility estimation that we obtain in MNN, against the best published results in the literature. There is only one data set with labelled occlusions, COFW [16], thus this is the one we use in our tests.

Method	occlusion precision/recall
RCPR [16]	80/40
Wu *et al.* [119]	80/44.43
Wu *et al.* [120]	80/49.11
ECT [138]	80/63.4
3DDE [113]	85.92/51.04
MNN	**81.93/69.75**

Table 4.6: Error of landmarks visibility estimation methods using COFW.

Consequently, we get a precision/recall of 81.93/69.75 inferring the visibility through the discriminative feature maps obtained from MNN, fine-tuning first from the landmark detection task and training finally with all tasks together. Our main competitor, ECT [138], which also estimates the visibility using a CNN achieves a precision/recall of 80/63.4, which indicates that our approach is more than 6 points better in recall, with a higher 81.93% precision. The improvement achieved is reasonable, since to decide whether a landmark is occluded or not, we first require its accurate location on the image. When compared to the landmarks visibility estimated through the ERT (see Section 3.4.2), our higher recall supports the importance of inferring occluded landmarks, by capturing local and global features in the MNN.

Facial landmark location

We continue previous discussion of head pose and landmarks visibility estimation tasks by comparing the MNN+OERT framework with the state-of-the-art, using the published results on 2D and 3D face alignment data sets. Here, we use both, the oldest data sets, *i.e.*, 300W public/private, COFW and AFLW to provide a reference comparison point, and the newer and more challenging ones, *i.e.*, WFLW and 300W-LP/AFLW2000-3D to asses our performance with the top competitors.

In Table 4.7 we first compare MNN+OERT against the 300W public benchmark. As we described in Section 3.5.6, Wing [36] introduces a loss function conceived to pay more

attention to the minimization of low error samples, and consequently achieves the best reported result (4.04 *NME*), because the *Common* subset (*i.e.*, easy/semifrontal faces) represents the 80% of the database. In the *Challenging* subset our results are still far from the ones obtained by SHN [129] based on the concatenation of four Hourglass models [81]. It is caused by the robustness of deep networks to extreme head poses and exaggerated facial expressions. However, we beat SHN by a larger margin in the *Full* set, which proves that our OERT stage is quite accurate estimating the non-rigid face deformation. Our approach results in a good compromise between the *Challenging* and *Common* subsets. ODN [143] focuses on how to preserve the face shape by learning the correlation among multiple landmarks, which sets the state-of-the-art in the *Challenging* subset, however its performance drops in COFW, since the amount of landmarks is lower (*i.e.*, 29 landmarks). Guo *et al.* [42] also achieves better results in the *Challenging* subset by training a cascade of U-Nets [90], but they pre-train using images from an additional data set, thus their results are not strictly comparable with those in Table 4.7, since they have been obtained using only the training images of the 300W benchmark.

Compared to another MTL approach such as HF-ResNet [86], we achieve a reduction of 11.2% in *NME* using the *Challenging* subset. In this case, it is also noticeable that the MNN+OERT framework does not improve the performance of our 3DDE [113] proposal. This is reasonable since the initialization presented in Algorithm 6 is as good as the one produced by our MNN in these easy faces, and 300W does not provide visibility labels.

Method	Common		Challenging		Full			
	pupils	corners	pupils	corners	pupils	corners		
	NME	NME	NME	NME	NME	NME	AUC_8	FR_8
DDN [132]	-	-	-	-	5.65	-	-	-
TCDCN [141]	4.80	-	8.60	-	5.54	-	-	-
MDM [107]	-	-	-	-	5.88	-	52.12	4.21
HF-ResNet [86]	-	-	8.18	-	-	-	-	-
3DDFA [145]	5.09	-	8.07	-	5.63	-	-	-
RCN [50]	4.67	-	8.44	-	5.41	-	-	-
ECT [138]	4.66	-	7.96	-	5.31	-	-	-
DSRN [76]	4.12	-	9.68	-	5.21	-	-	-
DAN [58]	4.42	3.19	7.57	5.24	5.03	3.59	55.33	1.16
TSR [72]	4.36	-	7.56	-	4.99	-	-	-
RAR [123]	4.12	-	8.35	-	4.94	-	-	-
SHN [129]	4.12	-	7.00	**4.90**	4.68	-	-	-
DU-Net [104]	-	2.82	-	5.07	-	3.26	-	-
PCD-CNN [61]	3.67	-	7.62	-	4.44	-	-	-
Wing [36]	**3.27**	-	7.18	-	**4.04**	-	-	-
CHR2C [110]	3.96	2.85	7.44	5.15	4.64	3.30	58.92	**1.16**
3DDE [113]	3.73	**2.69**	7.10	4.92	4.39	**3.13**	**61.24**	1.30
RDN [66]	3.31	-	7.04	-	4.23	-	-	-
ODN [143]	3.56	-	**6.67**	-	4.17	-	-	-
MNN+OERT	3.72	2.69	7.26	5.03	4.41	3.14	61.10	1.30

Table 4.7: Error of face alignment methods using the 300W public test set.

In Table 4.8 we have obtained the best published result in both *Indoor* and *Outdoor*

subsets of the 300W private competition using MNN+OERT because this is a benchmark with more images under challenging settings (extreme poses and illuminations) than those in 300W public.

In 300W private, we outperform the results of previous approaches [66, 117, 113] that achieve a lower *NME* using the 300W public data set. Furthermore, we get a reduction in NME of 1.35% when compared with 3DDE [113], which demonstrates that MNN+OERT benefits in challenging scenarios where the rigid estimation acquired in 3DDE (*i.e.*, initialization produced through the heatmaps) is not as robust as the one predicted directly by MNN.

Method	Indoor corners	Outdoor corners	Full corners				
	NME	NME	NME	AUC_8	FR_8	AUC_{10}	FR_{10}
MDM [107]	-	-	5.05	45.32	6.80	-	-
ECT [138]	-	-	-	45.98	3.17	-	-
DAN [58]	-	-	4.30	47.00	2.67	-	-
SHN [129]	4.10	4.00	4.05	-	-	-	-
CHR2C [110]	3.78	3.77	3.77	52.85	**0.83**	61.82	**0.00**
3DDE [113]	3.74	3.71	3.73	53.94	2.33	62.84	0.50
RDN [66]	-	-	-	-	-	53.70	2.43
LAB [117]	-	-	-	-	-	58.85	0.83
MNN+OERT	**3.70**	**3.67**	**3.68**	**54.52**	1.50	**63.35**	1.00

Table 4.8: Error of face alignment methods using the 300W private test set.

Previously, we have already proved in Table 4.6 that our framework achieves state-of-the-art results for the landmarks visibility estimation task using COFW. At this point, we analyze in Table 4.9 the MNN+OERT performance in COFW (the common benchmark to evaluate occlusions). Here, we outperform by a large margin (5.04 *NME*) the published results of most competing approaches [61, 66, 117, 36, 143, 36]. In fact, we demonstrate that our hybrid method does not require many landmarks annotated to produce consistent face shape predictions, using only features from visible landmarks. Indeed, MNN+OERT does not demand either an immense amount of labelled images to be successful under most non-rigid face deformations. It can be noticed when compared to models that report best results in 300W public (*e.g.*, ODN [143], Wing [36] or RDN [66]). Our three approaches, 3DDE [113], CHR2C [110] and MNN+OERT, set the state-of-the-art in this benchmark, which proves the shape preserving importance.

At this point, we also get a reduction in *NME* of 1.37% when compared with 3DDE [113] (same improvement obtained in 300W private). As the initialization obtained by 3DDE and MNN+OERT generally computes a good rigid pose estimation in all samples (*i.e.*, easy/semifrontal face images), we assume that MNN+OERT benefits in data sets where the visibility of landmarks is properly annotated.

In Table 4.10 we compare MNN+OERT with previous literature using AFLW images. This is a challenging database due to the large number of faces with occluded landmarks, which are unannotated. In fact, few approaches can train with missing landmarks. These semi-supervised methods evaluate their performance using all annotated landmarks (*21 landmarks*). In this case, we set the new state-of-the-art with MNN+OERT (1.97 *NME*) with 21 landmarks. It achieves a reduction of 20.8% in *NME* compared to PCD-CNN [61].

Method	pupils NME
TCDCN [141]	8.05
Wu *et al.* [119]	6.40
RAR [123]	6.03
DAC-CSR [37]	6.03
Wu *et al.* [120]	5.93
ECT [138]	5.98
SHN [129]	5.6
PCD-CNN [61]	5.77
RDN [66]	5.82
LAB [117]	5.58
Wing [36]	5.44
ODN [143]	5.30
3DDE [113]	5.11
CHR2C [110]	5.09
MNN+OERT	**5.04**

Table 4.9: Error of face alignment methods using COFW.

As in COFW, our three approaches, 3DDE [113], CHR2C [110] and MNN+OERT, set the state-of-the-art in this benchmark for all head pose intervals, although the published results are not strictly comparable because each approach defines its own training/testing partition. It is noticeable that MNN+OERT achieves a reduction in *NME* of 4.36% when compared with 3DDE [113] due to the better initialization and occlusion-awareness, since in AFLW most face images have large head rotations, and we infer landmark non-visibility in a semi-supervised fashion.

Method	21 landmarks			
	[0°, 30°] height NME	[30°, 60°] height NME	[60°, 90°] height NME	Full height NME
CCR [140]	-	-	-	5.72
Hyperface [86]	3.93	4.14	4.71	4.26
Kepler [60]	-	-	-	2.98
AIO [87]	2.84	2.94	3.09	2.96
HF-ResNet [86]	2.71	2.88	3.19	2.93
Binary-CNN [13]	2.77	2.60	2.64	2.85
PCD-CNN [61]	2.33	2.60	2.64	2.49
3DDE [113]	2.10	2.00	2.04	2.06
CHR2C [110]	2.07	1.86	**1.81**	1.98
MNN+OERT	**2.05**	**1.86**	1.85	**1.97**

Table 4.10: Error of face alignment methods using AFLW.

However, other competitive methods [54, 37, 72, 76, 143] evaluate their models without two of the most difficult landmarks, each located in one earlobe (*19 landmarks*). Ignoring

these two landmarks, we achieve a 1.95 *NME* in the *Full* set, but this result can be further improved, because we have trained our model in AFLW with 21 landmarks.

In Table 4.11 we evaluate MNN+OERT using WFLW. Here, we outperform our main competitor LAB [117] in all WFLW subsets by a large margin (4.61 *NME*), also confirmed with an improvement in *AUC* of 6.08%, from 53.23 to a 56.68. It is even more significant given that LAB uses edge heatmaps (not landmark based ones) and four encoder-decoder networks in a cascade arrangement.

Again, our three approaches, 3DDE [113], CHR2C [110] and MNN+OERT, establish the state-of-the-art in this benchmark. MNN+OERT achieves a *NME* reduction of 1.49% compared to 3DDE [113], similar to the one obtained in 300W private and COFW. It is worth mentioning the excellent performance achieved by CHR2C [110], which reaches the highest *AUC*, 57.55, in the *Full* set. In this case, a cascade of two CNNs, which infers the location of certain landmarks using the information of their neighbours, results in the best reported performance under all different sources of variability. We hypothesize that the extraction of more discriminative features in CHR2C [110] is decisive against the multiple nuisance factors of WFLW, including extreme poses, expressions, illuminations, make-ups, occlusions, and blurriness. However, it is important to note that MNN+OERT also estimates the head pose and landmarks visibility tasks, and it is more efficient, being a 22% faster than CHR2C [110] (see Section 4.3.2).

Method	Full corners			Pose corners			Expression corners			Illumination corners			Make-up corners			Occlusion corners			Blur corners		
	NME	AUC_{10}	FR_{10}	NME	AUC_{10}	FR_{10}	NME	AUC_{10}	FR_{10}	NME	AUC_{10}	FR_{10}	NME	AUC_{10}	FR_{10}	NME	AUC_{10}	FR_{10}	NME	AUC_{10}	FR_{10}
LAB [117]	5.27	53.23	7.56	10.24	23.45	28.83	5.51	49.51	6.37	5.23	54.33	6.73	5.15	53.94	7.77	6.79	44.90	13.72	6.32	46.30	10.74
3DDE [113]	4.68	55.44	5.04	8.62	26.40	22.39	5.21	51.75	5.41	4.65	56.02	3.86	4.60	55.36	6.79	5.77	46.92	9.37	5.41	49.57	6.72
CHR2C [110]	**4.39**	**57.55**	**3.55**	**7.58**	**31.85**	**18.09**	**4.72**	**55.04**	**3.82**	**4.39**	**57.94**	**2.57**	**4.18**	**58.82**	**1.94**	**5.37**	**49.63**	**7.06**	**5.09**	**51.54**	**5.30**
MNN+OERT	4.61	56.68	4.84	8.32	28.79	22.39	5.24	51.88	6.36	4.63	57.26	4.01	4.30	58.10	3.88	5.79	48.04	9.23	5.40	50.23	6.20

Table 4.11: Error of face alignment methods using WFLW.

Finally, in Table 4.12 we also evaluate MNN+OERT with the recent 3D face alignment data sets. We follow the standard protocol presented in [145] by using 300W-LP as train set and AFLW2000-3D as test set. In this case, we achieve 2.58 *NME* in the *Full* set that sets the new state-of-the-art with MNN+OERT, which represents an impressive reduction in *NME* of 16.2% compared to the best published result in the literature, MHM [28] (3.08 *NME*) based on a two-stage cascade of heatmap regressors. We assume that our two-stage hybrid strategy is specially effective in these 3D face alignment benchmarks, whose OERT stage is initialized using the projected landmarks from a mean 3D head model. Moreover, it validates the importance of the self-occlusion estimation and its use in the OERT.

Comparison using public code

Henceforth, since some published results in the literature are not fully comparable and it is common to make mistakes among the normalization required for each benchmark [31, 117] (*i.e.*, *pupils*, *corners* or *height* normalization measures), we use publicly available code to confirm their performance. In Fig. 4.12 we display the Cumulative Error Distribution (CED) curves for the testing sets of 300W public, 300W private, COFW, WFLW, AFLW and Menpo data sets. In the plot legends we also show the values of the *AUC* for faces with a *NME* smaller than ϵ and the *FR* representing the percentage of images with *NME* greater than ϵ. As a result, we provide the results obtained by running the following facial landmark detection algorithms: RCPR [16], ERT [55], cGPRT [63], RCN [50], DAN [58],

Method	[0°, 30°] height NME	[30°, 60°] height NME	[60°, 90°] height NME	Full height NME
RCPR [16]	4.26	5.96	13.18	7.80
3DSTN [9]	3.15	4.33	5.98	4.49
3DDFA [145]	2.84	3.57	4.96	3.79
PRN [34]	2.75	3.51	4.61	3.62
Binary-CNN [13]	2.47	3.01	4.31	3.26
MHM [28]	**2.36**	2.80	4.08	3.08
MNN+OERT	2.54	**2.24**	**3.34**	**2.58**

Table 4.12: Error of face alignment methods using 300W-LP/AFLW2000-3D.

LAB [117], SAN [31], DCFE [112], 3DDE [113], CHR2C [110] and MNN+OERT. In the Menpo competition, we have obtained the CED curve of MHM [28], where a single model is trained establishing correspondences between the semifrontal and profile configurations.

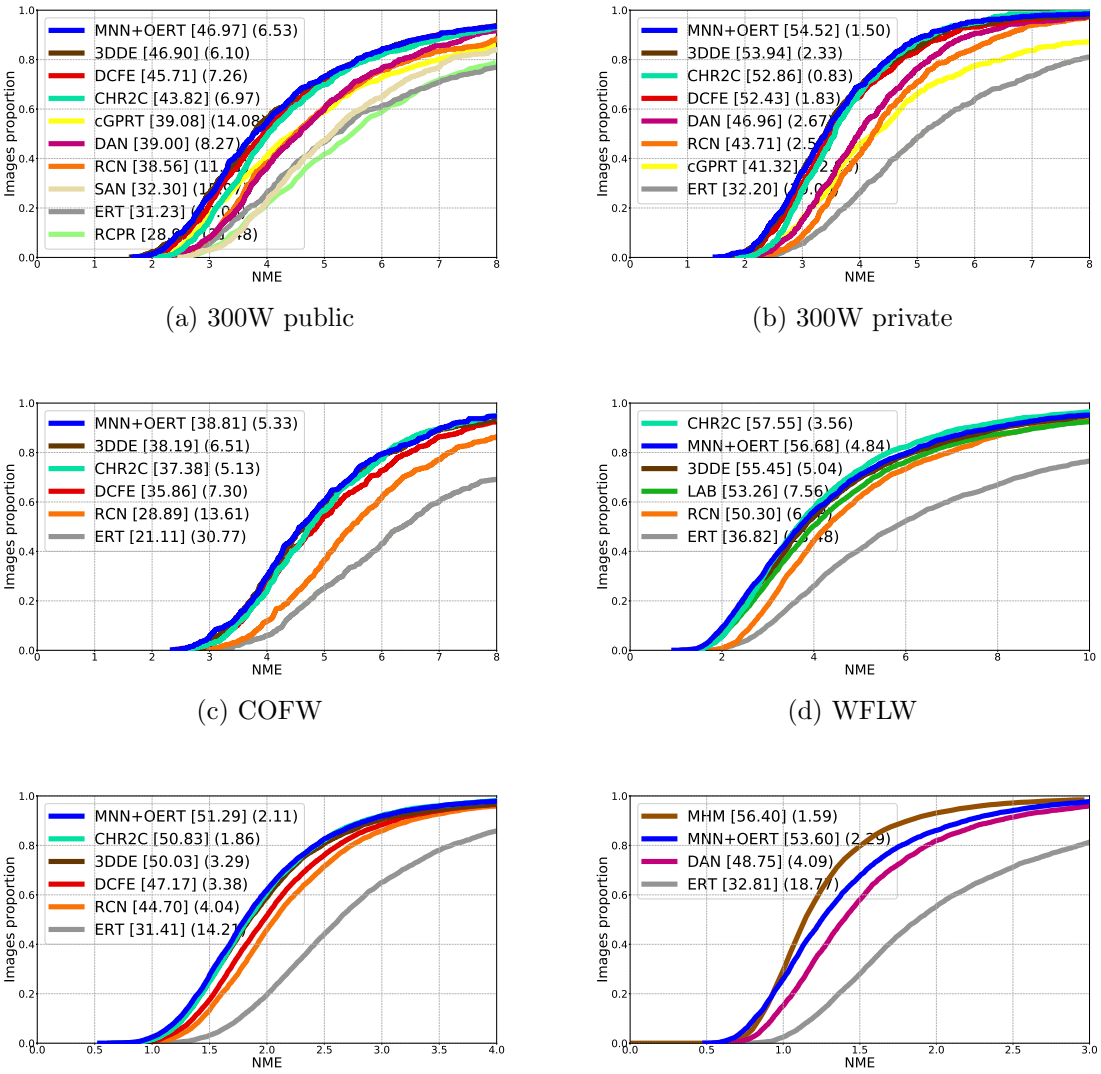
In these experiments we employ the same bounding boxes, training and validation sets. Note that some of the curves are missing, *e.g.*, LAB [117] only provides a trained model for WFLW, SAN [31] a trained model for 300W public and DAN [58] implementation lets us generate curves in 300W public/private and Menpo because it is based on 68 landmarks.

The selected algorithms are representative of the three main families of solutions: 1) ensembles of regression trees (cGPRT, RCPR, ERT); 2) CNN-based approaches (LAB, DAN, RCN, SAN, MHM, CHR2C); 3) our mixed approaches with CNNs and ensembles of regression trees (3DDE, DCFE, MNN+OERT). In these experiments we confirm that our CED curves are consistently above their closest competitors in 300W public/private, COFW, WFLW and AFLW for the majority of *NME* values. However, in Menpo we get the best performance in the "easy" faces with smaller *NME*, whereas MHM reports better results in samples with greater *NME*, because CNNs extract discriminative features, which are decisive against multiple nuisance factors. It seems reasonable since in WFLW we also achieve the best result using another cascade of CNNs, like CHR2C.

Following the discussion from published results, we confirm that MNN+OERT obtains the best CED curves in 300W public/private, COFW and AFLW data sets (see Figs. 4.12a, 4.12b, 4.12c and 4.12e). These findings are a good example of what we can get using the ERT to implicitly keep the face shape. In fact, MNN+OERT gets not only better results than CHR2C in four data sets, but also it computes simultaneously two additional tasks (*i.e.*, head pose and landmarks visibility estimation) and reduces a 22% the computational time required (see Section 4.3.2).

In WFLW and Menpo we prove that two different approaches based on sequences of heatmap regressors, CHR2C and MHM, exhibit superior capability in handling cases with *NME* error greater than $\epsilon = 3$ (*NME* normalized by the distance between eye *corners*) and $\epsilon = 0.75$ (*NME* normalized by the face bounding box *diagonal*) respectively, in comparison with MNN+OERT. This is not surprising since cascades of CNNs have claimed to produce accurate heatmaps against different nuisance factors, including extreme poses, expressions, illuminations, make-ups, occlusions, and blurriness [129, 13, 42, 104, 28, 117].

Further analysis leads to the conclusion that MNN+OERT is better than our preliminary hybrid algorithm, 3DDE, because of the better initialization and non-rigid estimation refinement using only visible landmarks. Even DAN [58] and LAB [117], using a cascade



(a) 300W public

(b) 300W private

(c) COFW

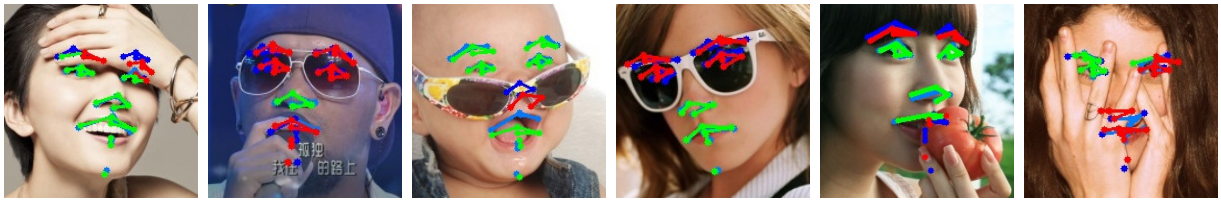
(d) WFLW

(e) AFLW

(f) Menpo semifrontal

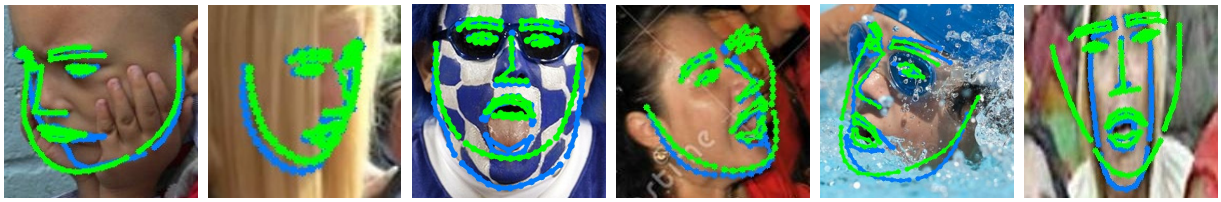
Figure 4.12: CED curves obtained using relevant 2D face alignment benchmarks. In the legend we display both area under each CED curve [AUC] and failure rate (FR) metrics.

of CNN regressors, can not compete with the regularization obtained by our framework. As shown in Fig. 4.13, keeping a correct face shape is crucial to produce accurate results in presence of occlusions and extreme head poses. We have also noticed in Figs. 4.13b, 4.13d and 4.13e that sometimes our prediction is more accurate than ground truth (see AFLW and AFLW2000-3D annotation errors in Figs 3.13 and 4.8). We note that MNN+OERT results are still unsatisfactory when two or more challenging factors occur together, *e.g.*, make-up and facial expression in the third face of Fig. 4.13c, severe occlusion and large head rotation in the fifth face of Fig. 4.13c, or exaggerated rotation in Fig. 4.13b (earlobes in the third face) and Fig. 4.13d (face contour in the fifth and sixth face), to name a few. Menpo-3D test annotations have not been released, but we have processed in Fig. 4.13f their testing images to visually perform an analysis of their results.



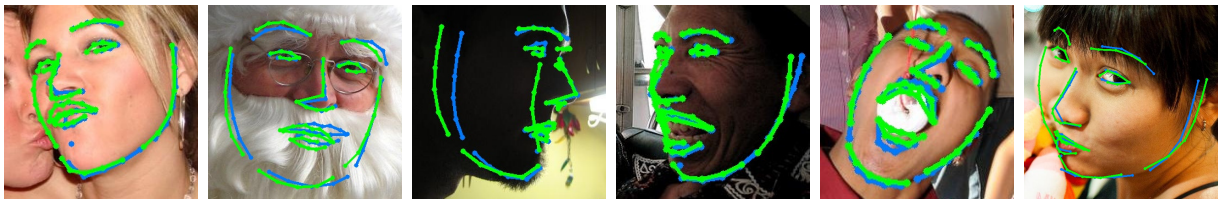
(a) COFW

(b) AFLW



(c) WFLW

(d) AFLW2000-3D



(e) Menpo

(f) Menpo-3D

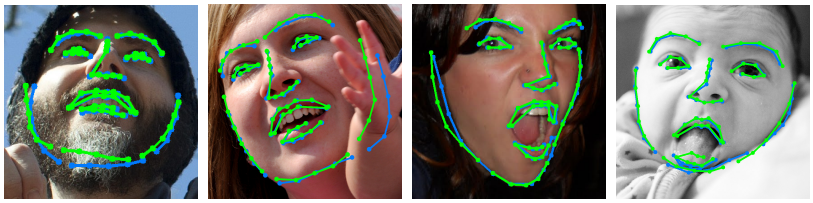
Figure 4.13: Representative sample face images considered errors according to FR_{ε} using MNN+OERT in COFW, AFLW, WFLW, AFLW2000-3D, Menpo and Menpo-3D testing subsets. Green, red and blue colours show visible, occluded predictions and ground truth respectively.

Conclusions

In this chapter, we train an encoder-decoder architecture, termed MNN, following a MTL strategy that sets the new state-of-the-art in the head pose estimation task, using AFLW benchmark introduced in Chapter 2, and 300W-LP/AFLW2000-3D protocol [145]. Our MNN also outperforms previous literature using COFW in the landmarks visibility estimation task. This improvement seems reasonable, since to decide the face orientation and whether a landmark is visible, we also require an accurate location of each face part of interest.

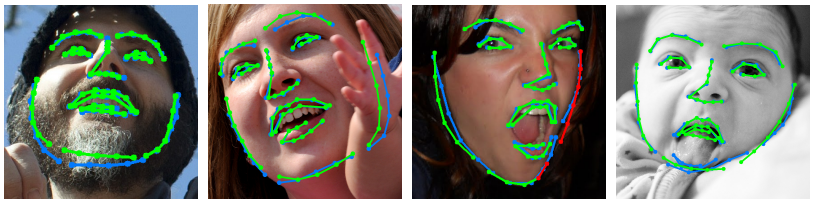
Additionally, we include the coarse-to-fine ERT introduced in Chapter 3, to develop a hybrid framework that consists of two stages, one MNN and one occlusion-aware ERT. Thus, we demonstrate that by combining the MNN with the complementary features of an ensemble of regression trees, we build a regressor that produces accurate facial landmark detection results in presence of occlusions and extreme head poses. MNN+OERT sets the state-of-the-art in 300W private, COFW and AFLW data sets (2D face alignment), and 300W-LP/AFLW2000-3D (3D face alignment) respectively. MNN+OERT also generates competitive results in WFLW and Menpo, obtaining the best accuracy in the "easy" faces, but it can not compete with the robust features extracted by using a sequence of CNNs, required to deal with multiple in-the-wild factors, including extreme poses, expressions, illuminations, occlusions and blurriness. Otherwise, MNN+OERT is more efficient than most cascades of CNNs, and it lets us train accurate models using a low number of images (*e.g.*, 300W, COFW) or landmarks annotated (*e.g.*, COFW, AFLW).

In Fig. 4.14 we visually compare the landmark predictions produced by our three main proposals, CHR2C [110], 3DDE [113], and MNN+OERT, in some images from the 300W private test set. The first two images show the importance of preserving the face shape in presence of heavy beard and partial occlusions (1.1 and 1.2 *NME* reduction respectively). The third face enhances the accuracy of our coarse-to-fine ERT scheme in the location of each part individually (0.8 *NME* reduction), whereas the fourth face shows the importance of the better features computed by CHR2C [110].



6.124 5.337 6.584 4.698

(a) CHR2C [110]



5.893 4.445 5.893 7.524

(b) 3DDE [113]

5.045 4.150 5.765 4.965

(c) MNN+OERT

Figure 4.14: Comparison among CHR2C [110], 3DDE [113], and MNN+OERT predictions using 300W private test set. We also report below their corresponding *NME*, normalized by the eye *corners* distance. Green, red and blue colours show visible, occluded predictions and ground truth respectively.

Chapter 5

Conclusions

Face alignment is a topic of intense research due to its importance as a pre-processing stage to minimize the impact of spurius facial appearance variations in the performance of facial analysis tasks. Further, face analysis is nowadays ubiquitous in applications related to biometry, augmented reality and surveillance, among many others. Given its interest, the goal of this thesis has been the development of an efficient and robust face alignment algorithm that advances the state of-the-art in this area.

The simultaneous estimation of both rigid and non-rigid face deformation entails great difficulty when applied to face images under in-the-wild conditions. Therefore, our first approach was to address the problem of estimating the 3D head orientation. This also alleviates the workload of subsequent non-rigid face deformation estimation. In the thesis we have introduced a benchmark to evaluate 3D head pose problems and provided a set of baseline results composed of a traditional Random Forest regressor using handcrafted local features [111], and several CNN-based regressors based on top performing CNN architectures [3].

Then, we considered the problem of estimating the location of various facial landmarks. To this end, we have adopted a strategy based on cascading a set of regressors. Since the irruption of deep learning, top performing algorithms are based on a sequence of CNNs with similar architecture. In this sense, face alignment is a reflection of what happens in the rest of computer vision research areas, which have been taken by methods based on deep neural networks. The large effective receptive field of deep models enable them to model context better and produce robust landmark estimations. However, in these CNN models, we can not enforce face shape consistency, something that limits their accuracy in the presence of occlusions and ambiguous facial configurations. To address this limitation we have proposed two new models. In the first one, CHR2C [110], we follow the mainstream approach and cascade two encoder-decoder CNNs with a U-Net-like architecture. We have designed and trained it in a way that enables the prediction of occluded and missing landmarks from the visible parts in the face. We have endowed this model with a densely connected layer with shared weights that improves by a significant margin the usual `argmax` approach to estimate landmark coordinates. The second approach emerges from the reflection that it is more effective combining few regressors with uncorrelated failure conditions than cascading a number of CNNs with similar structure and correlated errors. 3DDE is a hybrid model that cascades a CNN and an ERT [113]. It introduces a new coarse-to-fine ERT scheme that is able to deal with the combinatorial explosion of local parts deformation. In this case, the usual monolithic ERT will perform poorly when fitting faces with combinations of facial part deformations not present in the training set. This is a fundamental limitation of implicit shape models addressed by our proposal. It also inherits the best properties of ERTs and deep models. 3DDE is initialized by robustly fitting a 3D head model to the heatmaps produced by a CNN. With this initialization we tackle one of the main drawbacks of ERTs, namely the difficulty in initializing a regressor

in the presence of extreme face rotations. Moreover, the ERT implicitly imposes a prior face shape on the solution, addressing the shortcomings of deep models when occlusions and ambiguous face configurations are present. Lastly, the 3D head model used in the initialization enables 3DDE to exploit face orientation information to improve self-occlusion estimation. To the best of our knowledge this is the first hybrid image alignment model proposed in the literature.

Finally, concluding our quest of a shape preserving landmark estimator, we propose a multi-task model, termed MNN+OERT. Again, this algorithm combines an encoder-decoder CNN together with a coarse-to-fine occlusion-aware ERT regressor. We have trained the CNN using a multi-task scheme that takes advantage of the strong dependencies among the rigid pose, landmark location and visibility estimation tasks. The CNN architecture and its training procedure are novel and key components in our solution.

In our experiments we evaluate our regressors using 300W, COFW, AFLW, WLFW and 300W-LP/AFLW2000-3D, the most recent and challenging 2D and 3D face alignment benchmarks. We compare our proposals with the top performing algorithms in these data sets. Our approach has established new state-of-the-art results for head pose estimation using AFLW and AFLW2000-3D, landmarks visibility estimation using COFW, and facial landmark location using 300W private, COFW, AFLW, WLFW and AFLW2000-3D. In our experiments, we show that the key ingredient for these results is the combination of two regressors conveniently trained and designed. The robust heatmaps produced by the CNN are complemented with the implicit face shape enforced by the OERT. The latter is also positively boosted by the precise visibility and accurate head pose estimated by the multi-task CNN model, which enables the OERT to ignore occluded features and initialize the regression at a good starting point. The OERT shape preserving capability improves the final accuracy when the heatmaps provide a good initialization for the location of the landmarks. However, in challenging situations, with extreme poses, expressions and occlusions, the robustness of the cascade of CNNs is fundamental, as we have shown in the WFLW landmark location experiments.

To sum up, we should use a cascade of CNNs, such as CHR2C, if robustness is the main goal, *i.e.*, providing rough landmark estimation in presence of strong nuisance. However, if accuracy is critical, for example in augmented reality applications, then a face shape preserving solution, such as MNN+OERT, is the desired method.

5.1 Future work

The availability of large annotated data sets has also encouraged research in this area with important performance improvements in recent years. However, it is evident from our work that, there is still a group of extreme situations where the performance is still unsatisfactory. A critical question to improve our results is whether models trained with present data sets will generalize to the situations present in real-life. Our cross-dataset experiments reveal the existence of a significant bias in present data sets that limit the generalization of models trained with them. So, further work in this direction is required to improve the performance of present face alignment algorithms in the most challenging in-the-wild conditions. One possibility would be to increase the amount and quality of training data by synthetically generating more realistic occlusions, facial expressions, and extreme head orientations through a generative adversarial network (*e.g.*, [31]).

We may also improve our model in multiple ways. In the simplest case we use modern architectures, such as MobileNet [95], or densely connected U-Nets [42, 104] to improve

the performance of our backbone CNN. We can also endow our regressor with an end-to-end training capability. We independently train the components of MNN+OERT by first training MNN, and then, OERT. Moreover, we design a handcrafted feature extraction procedure from the MNN heatmaps to feed the OERT. For sure, this two-stage procedure produces non-optimal features while in an end-to-end trainable model, our features would be optimized for the problem at hand. This research is related to one of the main open problems in deep learning, namely, that of including prior information or restrictions in the learning process, such as the valid shape of a face.

A limitation of our approach is the use of a single mean 3D head model to initialize the ERT. We may improve the quality of the initialization using a 3D deformable model that would account for the deformations in faces caused by identity and facial expressions.

In many applications, robustness and precision are subject to limited hardware and battery constraints. Such is the case when applications run on mobile phones, drones or IOT devices. One of the major drawbacks of all top performing approaches is that all of them require the use of at least one CNN for feature extraction. Therefore, its use is inappropriate for real-time computation in hardware or battery limited environments. It is important to develop low-cost face alignment algorithms in terms of the energy and/or hardware required to run them. An immediate way to improve performance and reduce the computational cost of a face analysis application is to integrate in a single multi-task model the problems of face detection, alignment and the subsequent processing.

Bibliography

[1] J. Alabort-i-Medina and S. Zafeiriou. A unified framework for compositional fitting of active appearance models. *International Journal of Computer Vision*, 121(1):26–64, 2017. 29

[2] A. Alahi, R. Ortiz, and P. Vandergheynst. FREAK: fast retina keypoint. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 510–517, 2012. 30, 36, 37, 47

[3] E. Amador, R. Valle, J. M. Buenaposada, and L. Baumela. Benchmarking head pose estimation in-the-wild. In *Proc. Iberoamerican Congress on Pattern Recognition*, pages 45–52, 2017. 7, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 73, 79, 91

[4] M. Ariz, A. Villanueva, and R. Cabeza. Robust and accurate 2D-tracking-based 3D positioning method: Application to head pose estimation. *Computer Vision and Image Understanding*, 180:13–22, 2019. 11

[5] T. Baltrusaitis, P. Robinson, and L. Morency. Constrained local neural fields for robust facial landmark detection in the wild. In *Proc. International Conference on Computer Vision*, pages 354–361, 2013. 29, 30

[6] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, 2008. 36, 52

[7] J. Bekios-Calfa, J. M. Buenaposada, and L. Baumela. Robust gender recognition by exploiting facial attributes dependencies. *Pattern Recognition Letters*, 36:228–234, 2014. 1, 9

[8] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 545–552, 2011. 29

[9] C. Bhagavatula, C. Zhu, K. Luu, and M. Savvides. Faster than real-time facial alignment: A 3D spatial transformer network approach in unconstrained poses. In *Proc. International Conference on Computer Vision*, pages 4000–4009, 2017. 1, 66, 85

[10] V. Blanz and T. Vetter. Face recognition based on fitting a 3D morphable model. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pages 1063–1074, 2003. 29

[11] J. Booth, A. Roussos, S. Zafeiriou, A. Ponniah, and D. Dunaway. A 3D morphable model learnt from 10,000 faces. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 5543–5552, 2016. 30

[12] G. Borghi, M. Venturelli, R. Vezzani, and R. Cucchiara. POSEidon: Face-from-depth for driver pose estimation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 5494–5503, 2017. 9

[13] A. Bulat and G. Tzimiropoulos. Binarized convolutional landmark localizers for human pose estimation and face alignment with limited resources. In *Proc. International Conference on Computer Vision*, pages 3726–3734, 2017. 32, 40, 59, 83, 85

[14] 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 *Proc. International Conference on Computer Vision*, pages 1021–1030, 2017. 68, 72, 74

[15] 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 *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 109–117, 2018. 1, 27, 66

[16] X. P. Burgos-Artizzu, P. Perona, and P. Dollar. Robust face landmark estimation under occlusion. In *Proc. International Conference on Computer Vision*, pages 1513–1520, 2013. 30, 34, 35, 36, 48, 50, 56, 57, 58, 59, 68, 73, 80, 84, 85

[17] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. *International Journal of Computer Vision*, 107(2):177–190, 2014. 30, 34, 36, 38, 46, 50, 52, 56, 57, 59

[18] R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1998. 63

[19] E. Chong, N. Ruiz, Y. Wang, Y. Zhang, A. Rozga, and J. M. Rehg. Connecting gaze, scene, and attention: Generalized attention estimation via joint modeling of gaze and scene saliency. In *Proc. European Conference on Computer Vision*, pages 397–412, 2018. 9

[20] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *Proc. European Conference on Computer Vision*, pages 484–498, 1998. 10, 29

[21] T. F. Cootes, M. C. Ionita, C. Lindner, and P. Sauer. Robust and accurate shape model fitting using random forest regression voting. In *Proc. European Conference on Computer Vision*, pages 278–291, 2012. 29

[22] T. F. Cootes and C. J. Taylor. Active shape models - 'smart snakes'. In *Proc. British Machine Vision Conference*, pages 1–10, 1992. 29

[23] A. Criminisi, J. Shotton, and E. Konukoglu. Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning. Technical Report MSR-TR-2011-114, Microsoft Research, 2011. 12

[24] D. Cristinacce and T. F. Cootes. Automatic feature localisation with constrained local models. *Pattern Recognition*, 41(10):3054–3067, 2008. 10, 29

[25] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 886–893, 2005. 4

[26] M. Dantone, J. Gall, G. Fanelli, and L. V. Gool. Real-time facial feature detection using conditional regression forests. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 2578–2585, 2012. 9, 11, 12, 13, 31, 63

[27] D. DeMenthon and L. S. Davis. Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15(1-2):123–141, 1995. 11, 18, 23, 45, 77, 78, 80

[28] J. Deng, Y. Zhou, S. Cheng, and S. Zafeiriou. Cascade multi-view hourglass model for robust 3D face alignment. In *Proc. International Conference on Automatic Face and Gesture Recognition*, pages 399–403, 2018. 4, 33, 84, 85

[29] P. Dollar, Z. Tu, P. Perona, and S. Belongie. Integral channel features. In *Proc. British Machine Vision Conference*, pages 1–11, 2009. 4, 13

[30] P. Dollar, P. Welinder, and P. Perona. Cascaded pose regression. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1078–1085, 2010. 30, 34, 36

[31] X. Dong, Y. Yan, W. Ouyang, and Y. Yang. Style aggregated network for facial landmark detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 379–388, 2018. 33, 56, 84, 85, 92

[32] E. Eidinger, R. Enbar, and T. Hassner. Age and gender estimation of unfiltered faces. *IEEE Trans. Information Forensics and Security*, 9(12):2170–2179, 2014. 2

[33] G. Fanelli, M. Dantone, J. Gall, A. Fossati, and L. Van Gool. Random forests for real time 3D face analysis. *International Journal of Computer Vision*, 101(3):437–458, 2013. 10, 11, 14

[34] Y. Feng, F. Wu, X. Shao, Y. Wang, and X. Zhou. Joint 3D face reconstruction and dense alignment with position map regression network. In *Proc. European Conference on Computer Vision*, pages 557–574, 2018. 1, 66, 85

[35] Z. Feng, J. Kittler, M. Awais, P. Huber, and X. Wu. Face detection, bounding box aggregation and pose estimation for robust facial landmark localisation in the wild. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2106–2115, 2017. 3, 33, 65

[36] Z. Feng, J. Kittler, M. Awais, P. Huber, and X. Wu. Wing loss for robust facial landmark localisation with convolutional neural networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 2235–2245, 2018. 32, 40, 43, 56, 80, 81, 82, 83

[37] Z. Feng, J. Kittler, W. J. Christmas, P. Huber, and X. Wu. Dynamic attention-controlled cascaded shape regression exploiting training data augmentation and fuzzy-set sample weighting. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 3681–3690, 2017. 31, 58, 83

[38] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981. 10, 11, 29, 45

[39] B. Gao, C. Xing, C. Xie, J. Wu, and X. Geng. Deep label distribution learning with label ambiguity. *IEEE Trans. on Image Processing*, 26(6):2825–2838, 2016. 11, 15, 17, 21, 22, 23, 65

[40] X. Geng and Y. Xia. Head pose estimation based on multivariate label distribution. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1837–1842, 2014. 11, 21, 22

[41] G. Gkioxari, R. B. Girshick, and J. Malik. Contextual action recognition with R*CNN. In *Proc. International Conference on Computer Vision*, pages 1080–1088, 2015. 63

[42] J. Guo, J. Deng, N. Xue, and S. Zafeiriou. Stacked dense u-nets with dual transformers for robust face alignment. In *Proc. British Machine Vision Conference*, page 44, 2018. 4, 33, 40, 81, 85, 92

[43] H. Han, A. K. Jain, F. Wang, S. Shan, and X. Chen. Heterogeneous face attribute estimation: A deep multi-task learning approach. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 40(11):2597–2609, 2018. 1, 66

[44] E. M. Hand and R. Chellappa. Attributes for improved attributes: A multi-task network utilizing implicit and explicit relationships for facial attribute classification. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 4068–4074, 2017. 1, 66

[45] D. W. Hansen and Q. Ji. In the eye of the beholder: A survey of models for eyes and gaze. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(3):478–500, 2010. 9, 27

[46] S. L. Happy and A. Routray. Automatic facial expression recognition using features of salient facial patches. *IEEE Trans. on Affective Computing*, 6(1):1–12, 2015. 1, 27

[47] K. Hara and R. Chellappa. Growing regression forests by classification: Applications to object pose estimation. In *Proc. European Conference on Computer Vision*, pages 552–567, 2014. 11, 21, 22

[48] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer, 2009. 5, 6, 30, 50

[49] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 11, 16, 19, 20, 32, 66, 68, 70, 71, 75

[50] S. Honari, J. Yosinski, P. Vincent, and C. J. Pal. Recombinator networks: Learning coarse-to-fine feature aggregation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 5743–5752, 2016. 32, 40, 45, 56, 57, 58, 68, 81, 84

[51] H. Hsu, T. Wu, S. Wan, W. H. Wong, and C. Lee. Quatnet: Quaternion-based head pose estimation with multi-regression loss. *IEEE Trans. on Multimedia*, 2018. 11, 15, 17, 22, 23, 65, 79

[52] G. B. Huang, V. Jain, and E. G. Learned-Miller. Unsupervised joint alignment of complex images. In *Proc. International Conference on Computer Vision*, pages 1–8, 2007. 1, 3, 27, 33, 65

[53] X. Jin and X. Tan. Face alignment in-the-wild: A survey. *Computer Vision and Image Understanding*, 162:1–22, 2017. 2, 6, 27

[54] A. Jourabloo, M. Ye, X. Liu, and L. Ren. Pose-invariant face alignment with a single CNN. In *Proc. International Conference on Computer Vision*, pages 3219–3228, 2017. 4, 9, 29, 48, 56, 59, 83

[55] V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1867–1874, 2014. 27, 28, 30, 33, 34, 35, 36, 37, 38, 50, 52, 56, 57, 59, 84

[56] M. Koestinger, 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 *IEEE International Workshop on Benchmarking Facial Image Analysis Technologies*, pages 2144–2151, 2011. 11, 18, 48, 73

[57] I. Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 5454–5463, 2017. 63

[58] M. Kowalski, J. Naruniec, and T. Trzcinski. Deep alignment network: A convolutional neural network for robust face alignment. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2034–2043, 2017. 32, 40, 43, 56, 57, 81, 82, 84, 85

[59] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. Annual Conference on Neural Information Processing Systems*, pages 1106–1114, 2012. 11, 16, 19, 20, 66

[60] A. Kumar, A. Alavi, and R. Chellappa. KEPLER: simultaneous estimation of keypoints and 3D pose of unconstrained faces in a unified framework by learning efficient H-CNN regressors. *Image and Vision Computing*, 79:49–62, 2018. 12, 22, 23, 33, 59, 66, 67, 79, 83

[61] A. Kumar and R. Chellappa. Disentangling 3D pose in a dendritic CNN for unconstrained 2D face alignment. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 430–439, 2018. 9, 33, 56, 58, 59, 63, 81, 82, 83

[62] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In *Proc. International Conference on Computer Vision*, pages 365–372, 2009. 1, 27

[63] D. Lee, H. Park, and C. D. Yoo. Face alignment using cascade gaussian process regression trees. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 4204–4212, 2015. 30, 34, 36, 37, 38, 50, 56, 57, 84

[64] H. Li, H. Ding, D. Huang, Y. Wang, X. Zhao, J. Morvan, and L. Chen. An efficient multimodal 2D + 3D feature-based approach to automatic facial expression recognition. *Computer Vision and Image Understanding*, 140:83–92, 2015. 1, 27

[65] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollar. Focal loss for dense object detection. In *Proc. International Conference on Computer Vision*, pages 2999–3007, 2017. 4

[66] H. Liu, J. Lu, M. Guo, S. Wu, and J. Zhou. Learning reasoning-decision networks for robust face alignment. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2019. 32, 81, 82, 83

[67] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg. SSD: single shot multibox detector. In *Proc. European Conference on Computer Vision*, pages 21–37, 2016. 3, 4

[68] Z. Liu, Z. Chen, J. Bai, S. Li, and S. Lian. Facial pose estimation by deep learning from label distributions. In *Proc. International Conference on Computer Vision Workshops*, 2019. 11, 15, 17, 22, 23, 65, 79, 80

[69] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proc. International Conference on Computer Vision*, pages 3730–3738, 2015. 1, 17, 65

[70] P. D. López, R. Valle, and L. Baumela. Facial landmarks detection using a cascade of recombinator networks. In *Proc. Iberoamerican Congress on Pattern Recognition*, pages 575–583, 2018. 7, 40, 41, 43, 73

[71] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 10, 29, 30, 36, 65

[72] J. Lv, X. Shao, J. Xing, C. Cheng, and X. Zhou. A deep regression architecture with two-stage re-initialization for high performance facial landmark detection. In *Proc.*

IEEE Conference on Computer Vision and Pattern Recognition, pages 3691–3700, 2017. 32, 56, 81, 83

[73] M. J. Marín-Jiménez, A. Zisserman, M. Eichner, and V. Ferrari. Detecting people looking at each other in videos. *International Journal of Computer Vision*, 106(3):282–296, 2014. 9

[74] I. Masi, F. Chang, J. Choi, S. Harel, J. Kim, K. Kim, J. T. Leksut, S. Rawls, Y. Wu, T. Hassner, W. AbdAlmageed, G. G. Medioni, L. Morency, P. Natarajan, and R. Nevatia. Learning pose-aware models for pose-invariant face recognition in the wild. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 41(2):379–393, 2019. 1, 9

[75] M. Mathias, R. Benenson, M. Pedersoli, and L. V. Gool. Face detection without bells and whistles. In *Proc. European Conference on Computer Vision*, pages 720–735, 2014. 4

[76] X. Miao, X. Zhen, X. Liu, C. Deng, V. Athitsos, and H. Huang. Direct shape regression networks for end-to-end face alignment. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 5040–5049, 2018. 32, 56, 81, 83

[77] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert. Cross-stitch networks for multi-task learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 3994–4003, 2016. 63

[78] E. Muñoz, J. M. Buenaposada, and L. Baumela. A direct approach for efficiently tracking with 3D morphable models. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1615–1622, 2009. 29

[79] E. Muñoz, P. Márquez-Neila, and L. Baumela. Rationalizing efficient compositional image alignment - the constant jacobian gauss-newton optimization algorithm. *International Journal of Computer Vision*, 112(3):354–372, 2015. 29

[80] E. Murphy-Chutorian and M. M. Trivedi. Head pose estimation in computer vision: A survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(4):607–626, 2009. 9, 10, 18

[81] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *Proc. European Conference on Computer Vision*, pages 483–499, 2016. 32, 33, 40, 42, 45, 56, 81

[82] Y. Nirkin, Y. Keller, and T. Hassner. FSGAN: subject agnostic face swapping and reenactment. In *Proc. International Conference on Computer Vision*, 2019. 1, 27

[83] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *Proc. British Machine Vision Conference*, pages 41.1–41.12, 2015. 11, 22, 65

[84] M. Patacchiola and A. Cangelosi. Head pose estimation in the wild using convolutional neural networks and adaptive gradient methods. *Pattern Recognition*, 71:132–143, 2017. 11, 15, 22, 23

[85] X. Peng, J. Huang, Q. Hu, S. Zhang, and D. N. Metaxas. Three-dimensional head pose estimation in-the-wild. In *Proc. International Conference on Automatic Face and Gesture Recognition*, pages 1–6, 2015. 10, 21

[86] 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. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 41(1):121–135, 2019. 3, 12, 22, 23, 33, 56, 59, 66, 67, 79, 81, 83

[87] R. Ranjan, S. Sankaranarayanan, C. D. Castillo, and R. Chellappa. An all-in-one convolutional neural network for face analysis. In *Proc. International Conference on Automatic Face and Gesture Recognition*, pages 17–24, 2017. 3, 12, 22, 23, 33, 59, 66, 83

[88] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016. 4, 14

[89] S. Ren, X. Cao, Y. Wei, and J. Sun. Face alignment via regressing local binary features. *IEEE Trans. on Image Processing*, 25(3):1233–1245, 2016. 30, 34, 36, 50, 56, 57

[90] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI*, pages 234–241, 2015. 14, 32, 40, 41, 45, 68, 81

[91] R. Rothe, R. Timofte, and L. V. Gool. Deep expectation of real and apparent age from a single image without facial landmarks. *International Journal of Computer Vision*, 126(2-4):144–157, 2018. 11

[92] N. Ruiz, E. Chong, and J. M. Rehg. Fine-grained head pose estimation without keypoints. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2074–2083, 2018. 11, 15, 17, 22, 23, 65, 79

[93] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: database and results. *Image and Vision Computing*, 47:3–18, 2016. 18, 48, 74

[94] E. Sánchez-Lozano, G. Tzimiropoulos, and M. F. Valstar. Joint action unit localisation and intensity estimation through heatmap regression. In *Proc. British Machine Vision Conference*, page 233, 2018. 1, 27

[95] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018. 92

[96] 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 *Proc. International Conference on Computer Vision Workshops*, pages 1003–1011, 2015. 28, 64, 74

[97] X. Shen, Z. Lin, J. Brandt, and Y. Wu. Detecting and aligning faces by image retrieval. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 3460–3467, 2013. 65

[98] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 11, 14, 16, 19, 20, 32, 33, 65, 66

[99] S. Soltanpour, B. Boufama, and Q. M. J. Wu. A survey of local feature methods for 3D face recognition. *Pattern Recognition*, 72:391–406, 2017. 1, 27

[100] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 3476–3483, 2013. 6, 31

[101] K. Sundararajan and D. L. Woodard. Head pose estimation in the wild using approximate view manifolds. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 50–58, 2015. 10, 21

[102] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015. 11, 16, 19, 20, 66

[103] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014. 1, 27

[104] Z. Tang, X. Peng, S. Geng, L. Wu, S. Zhang, and D. N. Metaxas. Quantized densely connected u-nets for efficient landmark localization. In *Proc. European Conference on Computer Vision*, pages 348–364, 2018. 4, 33, 40, 56, 81, 85, 92

[105] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1521–1528, 2011. 23, 55

[106] L. Tran, X. Yin, and X. Liu. Disentangled representation learning GAN for pose-invariant face recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1283–1292, 2017. 1, 9

[107] 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 *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 4177–4187, 2016. 32, 56, 57, 81, 82

[108] G. Tzimiropoulos and M. Pantic. Fast algorithms for fitting active appearance models to unconstrained images. *International Journal of Computer Vision*, 122(1):17–33, 2017. 29

[109] R. Valenti, N. Sebe, and T. Gevers. Combining head pose and eye location information for gaze estimation. *IEEE Trans. on Image Processing*, 21(2):802–815, 2012. 9, 27

[110] R. Valle, J. M. Buenaposada, and L. Baumela. Cascade of encoder-decoder CNNs with learned coordinates. *Pattern Recognition Letters*, (in press), 2019. 7, 40, 47, 49, 55, 56, 57, 58, 59, 73, 75, 81, 82, 83, 84, 85, 88, 89, 91

[111] R. Valle, J. M. Buenaposada, A. Valdés, and L. Baumela. Head-pose estimation in-the-wild using a random forest. In *Proc. Articulated Motion and Deformable Objects*, pages 24–33, 2016. 7, 12, 18, 19, 21, 23, 24, 25, 73, 91

[112] R. Valle, J. M. Buenaposada, A. Valdés, and L. Baumela. A deeply-initialized coarse-to-fine ensemble of regression trees for face alignment. In *Proc. European Conference on Computer Vision*, pages 609–624, 2018. 7, 44, 45, 71, 73, 85

[113] R. Valle, J. M. Buenaposada, A. Valdés, and L. Baumela. Face alignment using a 3D deeply-initialized ensemble of regression trees. *Computer Vision and Image Understanding*, 189, 2019. 7, 8, 44, 45, 47, 49, 55, 56, 57, 58, 59, 60, 63, 70, 71, 72, 73, 77, 80, 81, 82, 83, 84, 85, 88, 89, 91

[114] P. A. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004. 4

[115] W. Wang, X. Alameda-Pineda, D. Xu, P. Fua, E. Ricci, and N. Sebe. Every smile is unique: Landmark-guided diverse smile generation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 7083–7092, 2018. 1, 27

[116] Z. Wang, Z. Dai, B. Póczos, and J. G. Carbonell. Characterizing and avoiding negative transfer. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 11293–11302, 2019. 63

[117] W. Wu, C. Qian, S. Yang, Q. Wang, Y. Cai, and Q. Zhou. Look at boundary: A boundary-aware face alignment algorithm. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 2129–2138, 2018. 27, 33, 40, 48, 56, 57, 58, 59, 60, 82, 83, 84, 85

[118] W. Wu, Y. Zhang, C. Li, C. Qian, and C. C. Loy. ReenactGAN: Learning to reenact faces via boundary transfer. In *Proc. European Conference on Computer Vision*, pages 622–638, 2018. 1, 27

[119] Y. Wu, C. Gou, and Q. Ji. Simultaneous facial landmark detection, pose and deformation estimation under facial occlusion. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 5719–5728, 2017. 12, 33, 58, 65, 67, 68, 80, 83

[120] Y. Wu and Q. Ji. Robust facial landmark detection under significant head poses and occlusion. In *Proc. International Conference on Computer Vision*, pages 234–241, 2015. 30, 56, 58, 80, 83

[121] Y. Wu and Q. Ji. Constrained joint cascade regression framework for simultaneous facial action unit recognition and facial landmark detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 3400–3408, 2016. 1, 27, 33, 65

[122] Y. Wu and Q. Ji. Facial landmark detection: A literature survey. *International Journal of Computer Vision*, 127(2):115–142, 2019. 6, 27, 28, 47

[123] S. Xiao, J. Feng, J. Xing, H. Lai, S. Yan, and A. A. Kassim. Robust facial landmark detection via recurrent attentive-refinement networks. In *Proc. European Conference on Computer Vision*, pages 57–72, 2016. 32, 56, 58, 81, 83

[124] X. Xiong and F. D. la Torre. Supervised descent method and its applications to face alignment. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 532–539, 2013. 30, 34, 56, 59, 65

[125] X. Xiong and F. D. la Torre. Global supervised descent method. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 2664–2673, 2015. 30

[126] H. Yang, W. Mou, Y. Zhang, I. Patras, H. Gunes, and P. Robinson. Face alignment assisted by head pose estimation. In *Proc. British Machine Vision Conference*, pages 130.1–130.13, 2015. 9, 11, 18, 23, 63

[127] H. Yang and I. Patras. Sieving regression forest votes for facial feature detection in the wild. In *Proc. International Conference on Computer Vision*, pages 1936–1943, 2013. 31

[128] H. Yang, R. Zhang, and P. Robinson. Human and sheep facial landmarks localisation by triplet interpolated features. In *Proc. Winter Conference on Applications of Computer Vision*, pages 1–8, 2016. 31, 34, 36, 47

[129] J. Yang, Q. Liu, and K. Zhang. Stacked hourglass network for robust facial landmark localisation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2025–2033, 2017. 32, 40, 56, 57, 58, 68, 81, 82, 83, 85

[130] T. Yang, Y. Chen, Y. Lin, and Y. Chuang. FSA-Net: Learning fine-grained structure aggregation for head pose estimation from a single image. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1087–1096, 2019. 11, 79

[131] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Proc. Annual Conference on Neural Information Processing Systems*, pages 3320–3328, 2014. 17, 65

[132] X. Yu, F. Zhou, and M. Chandraker. Deep deformation network for object landmark localization. In *Proc. European Conference on Computer Vision*, pages 52–70, 2016. 31, 40, 56, 81

[133] S. Zafeiriou, G. G. Chrysos, A. Roussos, E. Ververas, J. Deng, and G. Trigeorgis. The 3D menpo facial landmark tracking challenge. In *Proc. International Conference on Computer Vision Workshops*, pages 2503–2511, 2017. 2, 68, 72, 74

[134] S. Zafeiriou, G. Trigeorgis, G. Chrysos, J. Deng, and J. Shen. The menpo facial landmark localisation challenge: A step towards the solution. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2116–2125, 2017. 74

[135] S. Zafeiriou, C. Zhang, and Z. Zhang. A survey on face detection in the wild: Past, present and future. *Computer Vision and Image Understanding*, 138:1–242, 2015. 3

[136] A. R. Zamir, A. Sax, W. B. Shen, L. J. Guibas, J. Malik, and S. Savarese. Taskonomy: Disentangling task transfer learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 3712–3722, 2018. 63, 67

[137] F. Zhang, T. Zhang, Q. Mao, and C. Xu. Joint pose and expression modeling for facial expression recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 3359–3368, 2018. 1, 9

[138] H. Zhang, Q. Li, Z. Sun, and Y. Liu. Combining data-driven and model-driven methods for robust facial landmark detection. *IEEE Trans. Information Forensics and Security*, 13:2409–2422, 2018. 4, 30, 56, 57, 58, 80, 81, 82, 83

[139] J. Zhang and H. Hu. Exemplar-based cascaded stacked auto-encoder networks for robust face alignment. *Computer Vision and Image Understanding*, 171:95–103, 2018. 4, 30, 56

[140] W. Zhang, H. Zhang, Q. Li, F. Liu, Z. Sun, X. Li, and X. Wanu. Cross-cascading regression for simultaneous head pose estimation and facial landmark detection. In *Biometric Recognition*, pages 148–156, 2018. 12, 22, 23, 33, 59, 66, 79, 83

[141] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Learning deep representation for face alignment with auxiliary attributes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 38(5):918–930, 2016. 1, 6, 27, 33, 56, 58, 66, 67, 81, 83

[142] Y. Zhou, J. Pi, and B. E. Shi. Pose-independent facial action unit intensity regression based on multi-task deep transfer learning. In *Proc. International Conference on Automatic Face and Gesture Recognition*, pages 872–877, 2017. 1, 9, 12, 65, 66

[143] M. Zhu, D. Shi, M. Zheng, and M. Sadiq. Robust facial landmark detection via occlusion-adaptive deep networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 3486–3496, 2019. 32, 81, 82, 83

[144] S. Zhu, C. Li, C. C. Loy, and X. Tang. Face alignment by coarse-to-fine shape searching. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 4998–5006, 2015. 31, 34, 56, 57, 59

[145] X. Zhu, X. Liu, Z. Lei, and S. Z. Li. Face alignment in full pose range: A 3D total solution. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 41(1):78–92, 2017. 29, 56, 68, 72, 73, 74, 81, 84, 85, 88

[146] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 2879–2886, 2012. 2, 3, 10, 12, 18, 19, 21, 29, 33, 63, 65