



# Rectified Wing Loss for Efficient and Robust Facial Landmark Localisation with Convolutional Neural Networks

Zhen-Hua Feng<sup>1</sup> · Josef Kittler<sup>1</sup> · Muhammad Awais<sup>1</sup> · Xiao-Jun Wu<sup>2</sup>

Received: 20 April 2019 / Accepted: 4 December 2019 / Published online: 17 December 2019  
© The Author(s) 2019

## Abstract

Efficient and robust facial landmark localisation is crucial for the deployment of real-time face analysis systems. This paper presents a new loss function, namely Rectified Wing (RWin) loss, for regression-based facial landmark localisation with Convolutional Neural Networks (CNNs). We first systemically analyse different loss functions, including L2, L1 and smooth L1. The analysis suggests that the training of a network should pay more attention to small-medium errors. Motivated by this finding, we design a piece-wise loss that amplifies the impact of the samples with small-medium errors. Besides, we rectify the loss function for very small errors to mitigate the impact of inaccuracy of manual annotation. The use of our RWin loss boosts the performance significantly for regression-based CNNs in facial landmarking, especially for lightweight network architectures. To address the problem of under-representation of samples with large pose variations, we propose a simple but effective boosting strategy, referred to as pose-based data balancing. In particular, we deal with the data imbalance problem by duplicating the minority training samples and perturbing them by injecting random image rotation, bounding box translation and other data augmentation strategies. Last, the proposed approach is extended to create a coarse-to-fine framework for robust and efficient landmark localisation. Moreover, the proposed coarse-to-fine framework is able to deal with the small sample size problem effectively. The experimental results obtained on several well-known benchmarking datasets demonstrate the merits of our RWin loss and prove the superiority of the proposed method over the state-of-the-art approaches.

**Keywords** Facial landmark localisation · Deep convolutional neural networks · Rectified Wing Loss · Pose-based data balancing · Coarse-to-fine networks

## 1 Introduction

Facial landmark localisation, also known as face alignment, aims to automatically localise a set of pre-defined 2D key

---

Communicated by Li Liu, Matti Pietikäinen, Jie Qin, Jie Chen, Wanli Ouyang, Luc Van Gool.

✉ Zhen-Hua Feng  
z.feng@surrey.ac.uk

Josef Kittler  
j.kittler@surrey.ac.uk

Muhammad Awais  
m.a.rana@surrey.ac.uk

Xiao-Jun Wu  
wu\_xiaojun@jiangnan.edu.cn

<sup>1</sup> Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford GU2 7XH, UK

<sup>2</sup> School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China

points for a given facial image. A facial landmark usually has a specific semantic meaning, *e.g.* nose tip or eye centre, which provides rich geometric information for other face analysis tasks such as face recognition (Taigman et al. 2014; Masi et al. 2016a; Liu et al. 2017a; Yang et al. 2017b; Wu and Ji 2019; Deng et al. 2019), emotion estimation (Fabian Benitez-Quiroz et al. 2016; Walecki et al. 2016; Li et al. 2017; Zeng et al. 2009) and 3D face reconstruction (Kittler et al. 2016; Roth et al. 2016; Koppen et al. 2018; Deng et al. 2018; Feng et al. 2018a).

Thanks to the successive developments in this area of research during the past decades, we are able to achieve accurate facial landmark localisation in constrained scenarios even using traditional approaches such as active shape model (Cootes et al. 1995) and active appearance model (Cootes et al. 2001). The existing challenge is to perform efficient and robust landmark localisation of unconstrained faces that are impacted by a variety of appearance variations, *e.g.* in pose, expression, illumination, image blur

and occlusion. To address this challenge, Cascaded Shape Regression (CSR) has been widely used. The key idea of CSR is to form a strong regressor by cascading a set of weak regressors (Doll et al. 2010; Xiong and Torre 2013). CSR-based facial landmark localisation approaches have proved to be very successful, delivering promising performance in terms of both accuracy and efficiency (Cao et al. 2014; Feng et al. 2015b; Ren et al. 2016; Wu et al. 2017a; Feng et al. 2017a; Jourabloo and Liu 2017). However, the capability of CSR is practically saturated due to its shallow structure. After cascading more than four or five weak regressors, the performance of CSR is hard to improve further (Sun et al. 2015; Feng et al. 2015a). More recently, deep neural networks have been put forward as a more powerful alternative in a wide range of computer vision and pattern recognition tasks, including facial landmark localisation (Sun et al. 2013; Zhang et al. 2016b, a; Lv et al. 2017; Yang et al. 2017a; Wu et al. 2017b; Ranjan et al. 2017).

To perform robust facial landmark localisation with deep neural networks, different network architectures have been explored, including Convolutional Neural Networks (CNN) (Sun et al. 2013; Feng et al. 2019), auto-encoder (Zhang et al. 2014; Weng et al. 2016), deep belief networks (Luo et al. 2012) and recurrent neural networks (Trigeorgis et al. 2016; Xiao et al. 2016). In general, deep-learning-based facial landmark localisation approaches can be divided into two main categories: regression-based (Trigeorgis et al. 2016; Lv et al. 2017; Feng et al. 2019) and heatmap-based (Yang et al. 2017a; Deng et al. 2019b; Bulat and Tzimiropoulos 2017a, b; Wu et al. 2018). For regression-based methods, a network directly outputs a vector consisting of the 2D coordinates of all the landmarks. In contrast, a heatmap-based method outputs multiple heatmaps, each corresponding to a single facial landmark. The intensity value of a pixel in a heatmap indicates the probability that its location is the predicted position of the corresponding landmark. Despite the success of heatmap-based approaches in landmark localisation, they are computationally expensive. Such a method cannot meet the requirements for the deployment in real-time facial analysis systems. In this paper, we focus on regression-based facial landmark localisation due to the fast inference speed.

One crucial aspect of regression-based facial landmark localisation with CNNs is to define a loss function leading to a better-learned representation from underlying data. However, this aspect of the design seems to be scarcely investigated by the community. To the best of our knowledge, most existing regression-based facial landmark localisation approaches with deep neural networks are based on the L2 loss function (Lv et al. 2017; Dong et al. 2018b; Zeng et al. 2018). However, it is well known that the L2 loss function is sensitive to outliers, which has been noted in connection with the bounding box regression problem in face detection (Girshick

2015). Rashid et al. (2017) also noticed this issue and used the smooth L1 loss function instead of L2. Additionally, outliers are not the only subset of points which deserve special consideration. We argue that the behaviour of the loss function at points exhibiting small-medium errors is just as crucial to finding a good solution to the facial landmarking problem. Based on more detailed analysis, we propose a new loss function, namely Rectified Wing (RWing) loss, for robust facial landmark localisation with CNNs. The main contributions of our work include:

- Presenting a systematic analysis of different loss functions that could be used for regression-based facial landmark localisation with CNNs, which to our best knowledge is the first such study carried out in connection with the landmark localisation problem. We empirically compare L1, L2 and smooth L1 loss functions and find that L1 and smooth L1 perform much better than the widely used L2 loss.
- A novel RWing loss function that is designed to improve the deep neural network training capability for small and medium range errors. In addition, to reduce the impact of manual annotation noise on the training of a network, our RWing loss omits very small errors by rectifying the loss function around zero. As shown in our experiments, our regression-based networks powered by the new loss function achieve more than 2000 fps on GPU, with comparable or even better performance over the state-of-the-art approaches in terms of accuracy.
- A data augmentation strategy, *i.e.* pose-based data balancing, that compensates the low frequency of occurrence of samples with large out-of-plane head rotations in the training set. The experimental results demonstrate that our pose-based data balancing not only improves the performance of a trained network for the samples with large pose variations but also maintains the performance for the samples with small head rotations.
- A coarse-to-fine framework is proposed to maximise the accuracy of our facial landmark localisation system. The proposed system achieves comparable or even better performance in accuracy as compared with advanced network architectures, *e.g.* ResNet, but with much faster inference speed. The experimental results demonstrate that the advantage of our coarse-to-fine framework is more prominent for the well-known small sample size problem, *i.e.* a training dataset has only a small number of samples, as reported in Sect. 8.3.1. More importantly, we present a deep analysis by comparing the use of two small coarse-to-fine networks and a single large-capacity network in terms of both accuracy and speed.

The rest of this paper is organised as follows. Section 2 presents a brief review of the related literature. The

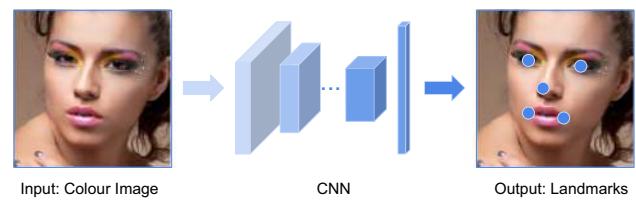
regression-based facial landmarking problem with CNNs is formulated in Sect. 3. The properties of common loss functions (L1, smooth L1 and L2) are discussed in Sect. 4 which also motivate the introduction of the novel RWinG loss function in Sect. 5. The pose-based data balancing strategy is the subject of Sect. 6. The coarse-to-fine localisation framework is presented in Sect. 7. The advocated approach is validated experimentally in Sect. 8 and the paper is drawn to conclusion in Sect. 9.

## 2 Related Work

In the last section, we mentioned some traditional facial landmark localisation algorithms, *e.g.* active shape model (Cootes et al. 1995), active appearance model (Cootes et al. 2001), constrained local model (Cristinacce and Cootes 2006) and cascaded shape regression (Doll et al. 2010; Xiong and Torre 2013). As the current mainstream of the area is to use deep neural networks, this section focuses on deep-learning-based methods. For traditional facial landmark localisation approaches, a reader is referred to comprehensive surveys (Wu and Ji 2019; Wang et al. 2018; Gao et al. 2010).

**Network Architecture** Most existing deep-learning-based facial landmark localisation approaches use regression networks. For such a landmarking task, the most straightforward way is to use a CNN model with a regression output layer (Sun et al. 2013; Rashid et al. 2017; Feng et al. 2019). The input for a regression CNN is usually an image patch enclosing the whole face region and the output is a vector consisting of the 2D coordinates of facial landmarks. Figure 1 depicts an example of CNN-based facial landmark localisation with the whole face region as an input. Instead of the whole face image, shape- or landmark-related local patches have also been widely used in deep-learning-based facial landmark localisation (Trigeorgis et al. 2016; Sun et al. 2013). To use local patches, one can apply CNN-based feature extraction to the neighbourhoods of all the landmarks and concatenate the extracted local features for landmark prediction or update (Trigeorgis et al. 2016). The advantage of the use of an image with a whole face region, in which the only input of the network is a cropped face image, is that it does not require the initialisation of facial landmarks. In contrast, to use local patches, a system usually requires initial estimates of facial landmarks for any given image. This can be achieved by either using the mean facial landmarks (Trigeorgis et al. 2016) or the output of a network coarsely landmarking the whole face image (Sun et al. 2013; Lv et al. 2017; Xiao et al. 2017).

Besides regression-based facial landmark localisation methods, recently heatmap-based variants have been proposed for the task and shown to deliver promising results,



**Fig. 1** Regression-based facial landmark localisation with convolutional neural networks. The input is a colour facial image and the output is a vector consisting of the coordinates of all the landmarks

*e.g.* fully convolutional network (Liang et al. 2015) and the hourglass network (Newell et al. 2016; Yang et al. 2017a; Deng et al. 2019b; Bulat and Tzimiropoulos 2017a, b; Wu et al. 2018). To reduce false alarms of a generated 2D sparse heatmap, Wu et al. (2018b) proposed a distance-aware softmax function that facilitates the training of their dual-path network. Wu et al. (2018) proposed to create a boundary heatmap mask using hourglass network for feature map fusion and showed its beneficial impact on the landmark localisation accuracy.

As noted in the last section, heatmap-based facial landmark localisation approaches are computationally expensive, which becomes an obstacle for the deployment of a network in real-time facial analysis systems. In this paper, we focus on efficient regression-based methods with CNNs. Thanks to the extensive studies of different deep neural network architectures and their use in unconstrained facial landmark localisation, the development of regression-based systems has recently been greatly stimulated. However, the current research lacks a systematic analysis of the effect of different loss functions on the solution. In this paper, we close this gap and design a new loss function for regression-based facial landmark localisation with CNNs.

**Dealing with Pose Variations** Extreme pose variations give rise to many difficulties in unconstrained facial landmark localisation. To mitigate this issue, different strategies have been explored. The *first* opts for multi-view models. There is a long history of the use of multi-view models in landmark localisation, from the earlier studies (Romdhani et al. 1999; Cootes et al. 2002) to recent work on cascaded-shape-regression-based (Xiong and Torre 2015; Zhu et al. 2016a; Feng et al. 2017b) and deep-learning-based approaches (Deng et al. 2019b). For example, we proposed to train multi-view cascaded shape regression models using a fuzzy membership weighting strategy, which, interestingly, outperformed even some deep-learning-based approaches (Feng et al. 2017b). The *second* strategy, which has become very popular in recent years, is to use 3D face models (Zhu et al. 2016b; Jourabloo and Liu 2016; Bhagavatula et al. 2017; Liu et al. 2017b; Jourabloo et al. 2017; Xiao et al. 2017). By recovering the 3D shape and estimating the pose of a given 2D face image, the issue of extreme pose

variations can be alleviated to a great extent. 3D face models have also been widely used to synthesise additional 2D face images with extreme pose variations for the training of a pose-invariant system (Masi et al. 2016b; Feng et al. 2015a; Zhu et al. 2016b). *Last*, multi-task learning has been adopted to address the difficulties posed by image degradation, including pose variation. For example, face attribute estimation, pose estimation or 3D face reconstruction can jointly be trained with facial landmark localisation (Zhang et al. 2016b; Xu and Kakadiaris 2017; Ranjan et al. 2017). The collaboration of different tasks in a multi-task learning framework can boost the performance of individual sub-tasks.

In contrast to these approaches, we treat the pose challenge as a training data imbalance problem and advocate a pose-based data balancing strategy to address this issue.

**Cascaded Networks** Motivated by the well known benefits of coarse-to-fine cascaded shape regression, multiple networks can be stacked to boost the performance further. To this end, shape- or landmark-related features should be used to satisfy the training of multiple networks in cascade. However, a CNN using a global face image as input cannot meet this requirement. To address this issue, one solution is to use local CNN features. This idea is advocated, for example, by Trigeorgis et al. (2016) who use CNN for local feature extraction and a recurrent neural network for landmark localisation in an end-to-end training fashion. As an alternative, one can train a network based on the global image patch for rough facial landmark localisation. Then, for each landmark or a composition of multiple landmarks in a specific region of the face, a new network is trained to perform fine-grained landmark prediction (Sun et al. 2013; Dong and Wu 2015; Weng et al. 2016; Yu et al. 2016; Xu and Kakadiaris 2017; Lv et al. 2017).

In this paper, we advocate a coarse-to-fine localisation framework. The first coarse network is very simple. It performs coarse facial landmark localisation at a very high speed. The aim of the first network is to mitigate the difficulties posed by inaccurate face detection and in-plane head rotations. The second CNN performs fine-grained landmark localisation by applying rigid transformation to an input image with the facial landmarks estimated by the first CNN. More importantly, we analyse the advantages of using two small networks compared to a single large-capacity network, in terms of both accuracy and speed.

### 3 Regression-Based Facial Landmark Localisation

As depicted in Fig. 1, the task of regression-based facial landmark localisation using CNNs is to find a nonlinear mapping function:

$$\Phi : \mathcal{I} \rightarrow \mathbf{s}, \quad (1)$$

that outputs a shape vector  $\mathbf{s} \in \mathbb{R}^{2L}$  for a given input colour image  $\mathcal{I} \in \mathbb{R}^{H \times W \times 3}$ . The input image is usually cropped from a bounding box output by a face detector. The shape vector is in the form of:

$$\mathbf{s} = [x_1, \dots, x_L, y_1, \dots, y_L]^T, \quad (2)$$

where  $L$  is the number of pre-defined 2D facial landmarks and  $(x_l, y_l)$  are the coordinates of the  $l$ th landmark. To obtain this mapping, first, we have to define a multi-layer neural network with randomly initialised parameters. In fact, a deep neural network is a compositional function:

$$\Phi = (\phi_1 \circ \dots \circ \phi_M)(\mathcal{I}), \quad (3)$$

consisting of  $M$  sub-functions, in which each sub-function ( $\phi$ ) stands for a specific layer in the network.

Given a set of labelled training samples  $\Omega = \{\mathcal{I}_i, \mathbf{s}_i\}_{i=1}^N$ , the target of CNN training is to find a  $\Phi$  that minimises:

$$\sum_{i=1}^N \text{loss}(\Phi(\mathcal{I}_i), \mathbf{s}_i), \quad (4)$$

where  $\text{loss}()$  is a pre-defined loss function that measures the difference between a predicted shape vector and its ground truth value. In this case, the CNN is used as a regression model learned in a supervised manner. To optimise the above objective function, a variety of optimisation methods, such as Stochastic Gradient Descent (SGD), Zeiler (2012) and Kingma and Ba (2014), can be used. In this paper, we use SGD with momentum for network training. Note that we also tested other optimisation approaches, but none of them resulted in higher accuracy than SGD.

### 4 Analysis of Different Loss Functions

In this section, we systematically analyse the impact of different loss functions as well as network architectures on regression-based facial landmark localisation. To the best of our knowledge, this is the first work in the area performing such a systematic analysis using different loss functions and CNN architectures.

We compare three different loss functions, including L2, L1 and smooth L1, using four different plain CNN architectures. The configurations of these plain CNN networks are shown in Table 1. In the rest of this paper, we use the term ‘CNN-5/6/7/8’ for these CNN models. Note that we do not use any fancy techniques, such as residual connection or intermediate supervision, in these plain CNN architectures

**Table 1** Configurations of different plain CNN models

| Model | Image resolution          | Convolutional Kernels                             | FC layer        | Model parameters | Model size | FLOPs  |
|-------|---------------------------|---|-----------------|------------------|------------|--------|
| CNN-5 | $32 \times 32 \times 3$   | $3 \times 3 \times 64/128/256/512$                | $512 \times 1$  | 2.6 M            | 10 MB      | 0.06 G |
| CNN-6 | $64 \times 64 \times 3$   | $3 \times 3 \times 64/128/256/512/1024$           | $1024 \times 1$ | 10 M             | 40 MB      | 0.33 G |
| CNN-7 | $128 \times 128 \times 3$ | $3 \times 3 \times 64/128/256/512/1024/2048$      | $2048 \times 1$ | 42 M             | 160 MB     | 1.61 G |
| CNN-8 | $256 \times 256 \times 3$ | $3 \times 3 \times 64/128/256/512/1024/2048/4096$ | $4096 \times 1$ | 168 M            | 640 MB     | 7.71 G |

For each plain CNN architecture, a convolutional block has one convolutional layer with multiple  $3 \times 3$  kernels followed by a ReLU nonlinear activation layer and a max pooling layer. A convolutional layer increases the feature map depth but does not change the feature map resolution by setting padding and stride to 1. A fully connected layer is followed by the last convolutional block, and a ReLU nonlinear activation function is applied for the final facial landmark prediction

**Fig. 2** Our plain CNN-6 network consisting of 5 convolutional and 1 fully connected layers followed by an output layer

so as not to cloud the comparison across different loss functions and network architectures with additional factors. We evaluate the performance of other network architectures such as MobileNets (Howard et al. 2017; Sandler et al. 2018), VGG (Parkhi et al. 2015) and ResNet (He et al. 2017) in Sect. 8.2.1.

The input of a plain CNN architecture is a colour image and the output is a vector of  $2L$  real numbers consisting of the coordinates of  $L$  2D facial landmarks. Each plain CNN has multiple convolutional layers with  $3 \times 3$  kernels, a fully connected layer and an output layer. After each convolutional and fully connected layer, a standard ReLU layer is used for nonlinear activation. A Max pooling layer following each ReLU layer is used to downsize the feature map to half of the size. As an example, Fig. 2 depicts the detailed architecture of our CNN-6 network.

Given a training image  $\mathcal{I}$  and a network  $\Phi$ , we can predict the facial landmarks as a vector  $\mathbf{s}' = \Phi(\mathcal{I})$ . The loss function is defined as:

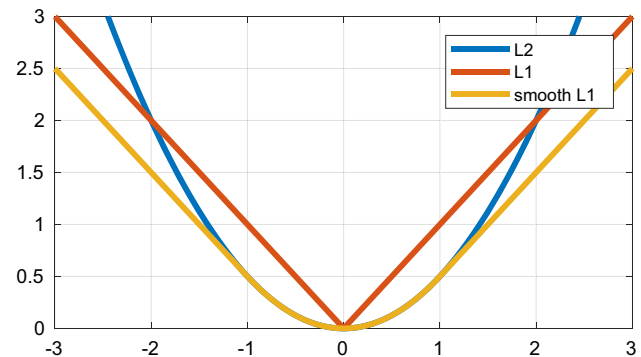
$$\text{loss}(\mathbf{s}, \mathbf{s}') = \frac{1}{2L} \sum_{i=1}^{2L} f(s_i - s'_i), \quad (5)$$

where  $\mathbf{s}$  is the ground-truth shape vector of the facial landmarks and  $s_i$  is its  $i$ th element. For  $f(x)$  in the above equation, the L2 loss is defined as:

$$f_{L2}(x) = \frac{1}{2}x^2, \quad (6)$$

and the L1 loss is defined as:

$$f_{L1}(x) = |x|. \quad (7)$$

**Fig. 3** Plots of the L2, L1 and smooth L1 loss functions

For the smooth L1 loss,  $f(x)$  is piecewise-defined as:

$$f_{smL1}(x) = \begin{cases} \frac{1}{2}x^2 & \text{if } |x| < 1 \\ |x| - \frac{1}{2} & \text{otherwise} \end{cases}, \quad (8)$$

which is quadratic for small values of  $|x|$  and linear for large values (Girshick 2015). More specifically, smooth L1 uses  $f_{L2}(x)$  for  $x \in (-1, 1)$  and shifted  $f_{L1}(x)$  elsewhere. Figure 3 depicts the plots of these three loss functions. It should be noted that the smooth L1 loss is a special case of the Huber loss (Huber 1964). The loss function that has widely been used in facial landmark localisation is L2. However, L2 loss is sensitive to outliers.

To perform empirical analysis, we use the AFLW dataset with the AFLW-Full protocol (Zhu et al. 2016a).<sup>1</sup> This protocol consists of 20,000 training and 4386 test images. Each image has 19 manually annotated facial landmarks. We train the plain CNN networks on AFLW using three different loss functions. In addition, we compare the results obtained by these CNN networks with five state-of-the-art baseline algorithms (Feng et al. 2017b; Lv et al. 2017; Dong et al. 2018b, a; Wu et al. 2018b). The first baseline method is a multi-view cascaded shape regression model, namely Dynamic Attention Controlled Cascaded Shape

<sup>1</sup> More details of AFLW are introduced in Sect. 8.3.2.

**Table 2** A comparison of different loss functions and network architectures on the AFLW-Full protocol in terms of the Normalised Mean Error (NME)

| Baseline Method                      | NME ( $\times 10^{-2}$ ) | Loss Function | NME ( $\times 10^{-2}$ ) |             |             |             |
|--------------------------------------|--------------------------|---------------|--------------------------|-------------|-------------|-------------|
|                                      |                          |               | CNN-5                    | CNN-6       | CNN-7       | CNN-8       |
| DAC-CSR (CVPR17) Feng et al. (2017b) | 2.27                     | L2            | 2.73                     | 2.33        | 2.35        | 2.41        |
| TR-DRN (CVPR17) Lv et al. (2017)     | 2.17                     | L1            | 2.16                     | 1.91        | 1.85        | 1.72        |
| CPM+SBR (CVPR18) Dong et al. (2018b) | 2.14                     | smooth L1     | 2.24                     | 1.93        | 1.85        | 1.73        |
| SAN (CVPR18) Dong et al. (2018a)     | 1.91                     | Wing          | 2.07                     | 1.81        | 1.73        | 1.64        |
| GoDP (IVC18) Wu et al. (2018b)       | 1.84                     | RWing         | <b>2.04</b>              | <b>1.77</b> | <b>1.71</b> | <b>1.63</b> |

Bold values indicate the best performing loss function on CNN-5/6/7/8

For CNN-5/6/7/8, each model was trained for 100 epochs with a batch size of 16. The learning rate was fine-tuned for each combination of a plain CNN architecture and a loss function. Five baseline algorithms, including DAC-CSR (Feng et al. 2017b), TR-DRN (Lv et al. 2017), CPM+SBR (Dong et al. 2018b), SAN (Dong et al. 2018a) and GoDP (Wu et al. 2018b), are used for further comparison

Regression (DAC-CSR) (Feng et al. 2017b). The other four baseline approaches are all deep-learning-based, including the Two-stage Re-initialisation Deep Regression Network (TR-DRN) (Lv et al. 2017), Supervision-by-Registration (SBR) (Dong et al. 2018b), Style Aggregated Network (SAN) (Dong et al. 2018a) and the Globally Optimised Dual Pathway neural network (GoDP) (Wu et al. 2018b). A comparison with more state-of-the-art algorithms on the AFLW dataset is reported in Sect. 8.

The results are reported in Table 2. The L2 loss function, which has been widely used for facial landmark localisation, obtains competitive results as compared with the baseline methods. Surprisingly, by simply switching the loss function from L2 to L1 or smooth L1, the landmarking error can be significantly reduced. CNN-7 outperforms DAC-CSR, TR-DRN, CPM+SBR and SAN in terms of accuracy and performs equally well as the GoDP approach. The combination of CNN-8 with L1 or smooth L1 beats all the baseline approaches. The NME of CNN-8 using the L1 loss function is  $1.72 \times 10^{-2}$ , which is around 7% lower than GoDP that has the NME of  $1.84 \times 10^{-2}$ .

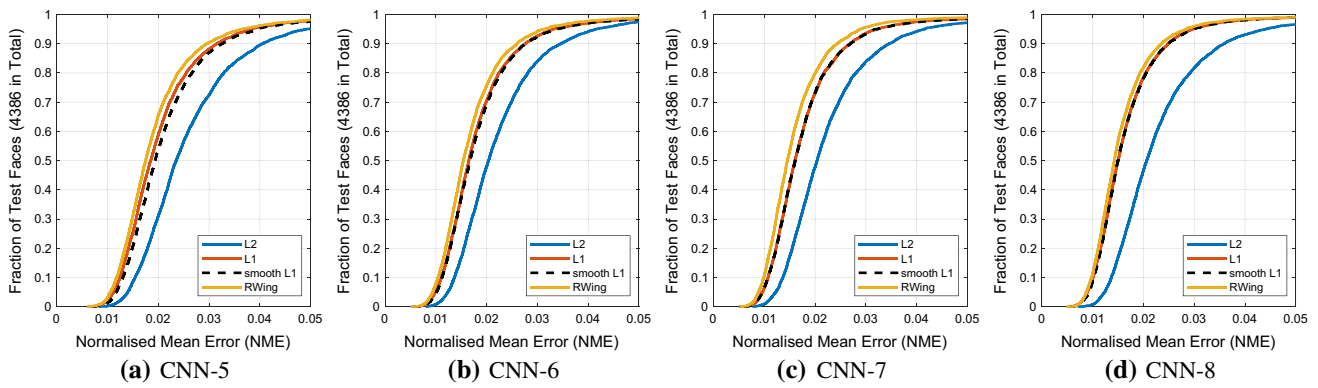
Another conclusion is that, a deeper network with higher resolution input images usually performs better in accuracy. This finding has also been validated in many other CNN-based computer vision and pattern recognition tasks, *e.g.* in VGG (Parkhi et al. 2015) and ResNet (He et al. 2017). To boost the performance in accuracy, more powerful network architectures are suggested, such as our coarse-to-fine landmark localisation framework presented in Sect. 7, VGG and ResNet. We will report the results of these advanced network architectures in Sect. 8.2.1. But the use of deeper and wider neural networks increases the computational complexity dramatically. For example, the model parameter and model size increase around four times by upgrading each plain network to a higher level, *e.g.* from CNN-6 to CNN-7, as shown in Table 1. Accordingly, the Floating Point Operations (FLOPs) increase around five times. In the next section, we propose a new loss function that brings further performance boosting for lightweight networks.

## 5 Rectified Wing Loss

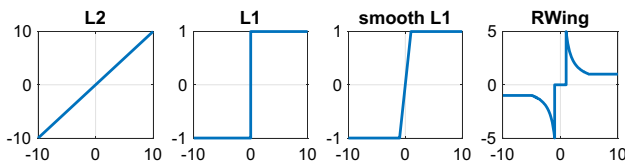
As analysed in the last section, the design of a proper loss function is crucial for regression-based facial landmark localisation with CNNs. However, predominantly the L2 loss has been used in existing deep-neural-network-based facial landmarking systems, in spite of the findings supporting the use of the L1 and smooth L1 loss functions (Girshick 2015; Rashid et al. 2017). Inspired by our analysis, we propose a new loss function, namely Rectified Wing (RWing) loss, to further improve the accuracy of a CNN-based facial landmark localisation system.

We first compare the results obtained on the AFLW dataset using four plain CNN architectures and three different loss functions (L2, L1 and smooth L1) in Fig. 4 by plotting the Cumulative Error Distribution (CED) curves. On one hand, we can see that all the loss functions analysed in the last section perform well for large errors, regardless of the choice of the CNN architecture. This indicates that the training of a neural network should pay more attention to the samples with small or medium range errors. On the other hand, it is very hard to achieve very small errors even for large-capacity networks, *e.g.* CNN-7 and CNN-8. The main reason stems from the residual noise in the ground truth labelling of the training data. These inaccuracies suggest that we should ignore very small errors in CNN training. To satisfy these two observations, we propose the RWing loss for CNN-based facial landmark localisation.

In order to motivate the new loss function, we provide an intuitive analysis of the properties of the classical loss functions, as shown in Fig. 3. We also plot their corresponding influence functions (derivatives) in Fig. 5. As shown in the figure, the magnitude of the gradients of the L1 and L2 loss functions is 1 and  $|x|$  respectively, and the magnitude of the corresponding optimal step sizes should be  $|x|$  and 1. Finding the minimum in either case is straightforward. However, the situation becomes more complicated when we try to optimise simultaneously the location of multiple points, as in our problem of facial landmark localisation formulated in



**Fig. 4** Cumulative error distribution curves comparing different loss functions on the AFLW dataset, using different plain CNN architectures



**Fig. 5** Plots of the influence functions (derivatives) of different loss functions. For the RWing loss function, we set the parameters  $r = 1$ ,  $w = 5$  and  $\epsilon = 1$

Eq. (5). In both cases the update towards the solution will be dominated by larger errors. In the case of L1, the magnitude of the gradient is the same for all the points, but the step size is disproportionately influenced by larger errors. For L2, the step size is the same but the gradient will be dominated by large errors. Thus in both cases it is hard to correct relatively small displacements.

The influence of small errors can be enhanced by an alternative loss function, such as  $\ln x$ . Its gradient, given by  $1/x$ , increases as we approach zero error. The magnitude of the optimal step size is  $x^2$ . When compounding the contributions from multiple points, the gradient will be dominated by small errors, but the step size by larger errors. This restores the balance between the influence of errors of different sizes. However, to prevent making large update steps in a potentially wrong direction, it is important not to overcompensate the influence of small localisation errors. This can be achieved by opting for a logarithm function with a positive offset. In addition, to eliminate the effects posed by noise, we rectify the loss function for very small values.

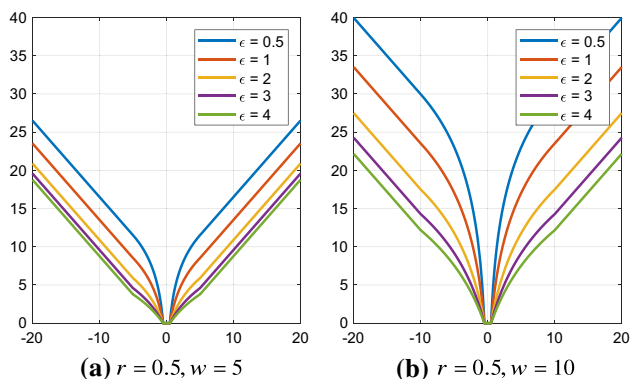
This type of loss function shape is appropriate for dealing with relatively small localisation errors. However, in facial landmark localisation of unconstrained faces we may be dealing with extreme appearance variations, e.g. pose, where initially the localisation errors can be very large. In such a regime the loss function should promote a fast recovery from these large errors for network training. This suggests that the loss function should behave more like L1 or L2. As L2 is sensitive to outliers, we favour L1.

The above intuitive argument points to a loss function which for very small errors should have the value of zero, for small medium range errors behave as a logarithm function with an offset, and for larger errors as L1. Such a loss function can be piecewise defined as:

$$RWing(x) = \begin{cases} 0 & \text{if } |x| < r \\ w \ln(1 + (|x| - r)/\epsilon) & \text{if } r \leq |x| < w \\ |x| - C & \text{otherwise} \end{cases}, \tag{9}$$

where the non-negative parameter  $r$  sets the range of rectified region to  $(-r, r)$  for very small values. For small medium range values with the absolute value in  $[r, w]$ , we use a modified logarithm function, where  $\epsilon$  limits the curvature of the nonlinear region and  $C = w - w \ln(1 + (w - r)/\epsilon)$  is a constant that smoothly links the piecewise-defined linear and nonlinear parts. Note that we should not set  $\epsilon$  to a very small value because this would make the training of a network very unstable and cause the exploding gradient problem for small errors. In fact, the nonlinear part of our RWing loss function just simply takes a part of the curve of  $\ln(x)$  and scales it along both the X-axis and Y-axis. Also, we apply translation along the Y-axis to allow  $RWing(\pm r) = 0$  and to impose continuity on the loss function at  $\pm w$ . Figure 6 depicts our RWing loss using different parameter settings.

We compare our RWing loss with other loss functions in Table 2 and Fig. 4. According to the figure, our RWing loss outperforms L2, L1 and smooth L1 in terms of accuracy for all the plain networks, i.e. CNN-5/6/7/8. Although the improvement for CNN-8 in Fig. 4 seems not obvious, the actual NME is reduced from  $1.72 \times 10^{-2}$  of L1 to  $1.63 \times 10^{-2}$ , which is around 6% lower than the best result obtained in the last section for CNN-8 (Table 2), and 11% lower than the best baseline approach, i.e. GoDP (Wu et al. 2018b). Additionally, by virtue of the proposed RWing loss, smaller networks are able to perform equally well or even better than larger networks. For example, CNN-6 is four times smaller



**Fig. 6** Our RWing loss function (Eq. 9) plotted with different parameter settings, where  $r$  and  $w$  limit the range of the non-linear part and  $\epsilon$  controls the curvature. By design, we amplify the impact of the samples with small and medium range errors and omit the impact of the samples with very small errors on the network training

in size as compared with CNN-7. But the NME of CNN-6 powered by our RWing loss is  $1.77 \times 10^{-2}$ , which is smaller than the NMEs of CNN-7 trained with L2 loss ( $2.35 \times 10^{-2}$ ), L1 loss ( $1.85 \times 10^{-2}$ ) and smooth L1 loss ( $1.85 \times 10^{-2}$ ). This validates the effectiveness of the proposed RWing loss for the training of lightweight CNNs.

As the facial landmarks of a training image are labelled by human annotators, they will be subject to individual biases of different annotators. Moreover, if we ask the same annotator to label the facial landmarks of the same image twice, the results will be slightly different, even for some landmarks with a clear semantic meaning such as eye corner or mouth corner. Note that the Wing loss function without rectification will have the highest gradient when a training sample has a very small error that might be caused by annotation noise. As aforementioned, for network training, a sample with very small errors should be ignored in back propagation. This observation motivates the idea of rectifying the Wing loss for very small errors. To validate the effectiveness of the rectification measure, we compared the performance of RWing loss and the Wing loss function without rectification. The results are reported in Table 2. According to the table, both Wing loss functions (with and without rectification) outperform L2, L1 and smooth L1 in accuracy, regardless of the network architecture. However, our rectified Wing loss has a slight performance edge over the Wing loss function without rectification. This confirms the merit of the proposed RWing loss function for regression-based facial landmark localisation.

### 6 Pose-Based Data Balancing

Extreme pose variations are very challenging for robust facial landmark localisation. To mitigate this problem, we propose

a simple but very effective Pose-based Data Balancing (PDB) strategy. We argue that the difficulty for accurately localising faces with large poses is mainly due to data imbalance, which is a well-known problem in many computer vision applications (Shrivastava et al. 2016). For example, given a training dataset, most samples in it are likely to be near-frontal faces. The network trained on such a dataset is dominated by frontal faces. By over-fitting to the frontal pose it cannot adapt well to faces with large poses. In fact, the difficulty of training and testing on merely frontal faces should be similar to that on profile faces. This is the main reason why a view-based face analysis algorithm usually works well for pose-varying faces. As an evidence, even the classical view-based active appearance model can localise faces with large poses very well (up to  $90^\circ$  in yaw) Cootes et al. (2000).

To perform PDB, we first align all the training shapes to a reference shape using Procrustes Analysis. Then we apply Principal Component Analysis (PCA) to the aligned shapes and project them to the one dimensional space defined by the shape eigenvector (pose space) controlling pose variations. To be more specific, for a training dataset  $\{\mathbf{s}_i\}_{i=1}^N$  with  $N$  samples, where  $\mathbf{s}_i \in \mathbb{R}^{2L}$  is the  $i$ th training shape vector consisting of the 2D coordinates of all the  $L$  landmarks, the use of Procrustes Analysis aligns all the training shapes to a reference shape, *i.e.* the mean shape, using rigid transformations. Then we can approximate any training shape or a new shape,  $\mathbf{s}$ , using a statistical linear shape model:

$$\mathbf{s} \approx \bar{\mathbf{s}} + \sum_{j=1}^{N_s} p_j \mathbf{s}_j^*, \tag{10}$$

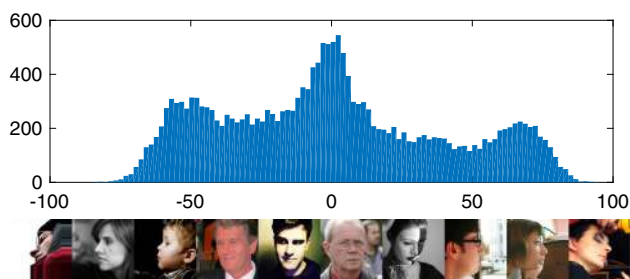
where  $\bar{\mathbf{s}} = \frac{1}{N} \sum_{i=1}^N \mathbf{s}_i$  is the mean shape over all the training samples,  $\mathbf{s}_j^*$  is the  $j$ th eigenvector obtained by applying PCA to all the aligned training shapes and  $p_j$  is the coefficient of the  $j$ th shape eigenvector. Among those shape eigenvectors, we can find an eigenvector, usually the first one, that controls the pose variation. We denote this eigenvector as  $\hat{\mathbf{s}}$ . Then we can obtain the pose coefficient of each training sample  $\mathbf{s}_i$  as:

$$\hat{p}_i = \hat{\mathbf{s}}^T (\mathbf{s}_i - \bar{\mathbf{s}}). \tag{11}$$

We plot the distribution of the pose coefficients of all the AFLW training samples in Fig. 7. According to the figure, we can see that the AFLW dataset is not well-balanced in terms of pose variation.

With the pose coefficients of all the training samples, we first categorise the training dataset into  $K$  subsets. Then we balance the training data by duplicating the samples falling into the subsets of lower cardinality. To be more specific, we denote the number of training samples of the  $k$ th subset as  $B_k$  and the maximum size of the  $K$  subsets as  $B^*$ . To balance the whole training dataset in terms of pose variation, we add





**Fig. 7** Distribution of the pose coefficients of the AFLW training samples obtained by projecting their shapes to the 1-D pose space

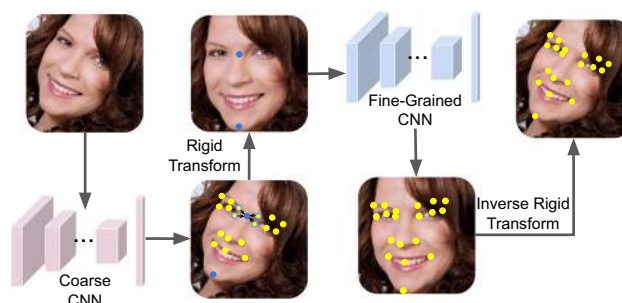
**Table 3** A comparison of the performance of different loss functions and CNN architectures using our PDB strategy, measured in terms of the Normalised Mean Error (NME) ( $\times 10^{-2}$ ) on AFLW

| Method      | Loss |      |           |             |
|-------------|------|------|-----------|-------------|
|             | L2   | L1   | smooth L1 | RWing       |
| CNN-5       | 2.73 | 2.16 | 2.24      | 2.04        |
| CNN-5 + PDB | 2.53 | 2.12 | 2.19      | <b>2.01</b> |
| CNN-6       | 2.33 | 1.91 | 1.93      | 1.77        |
| CNN-6 + PDB | 2.11 | 1.82 | 1.85      | <b>1.75</b> |
| CNN-7       | 2.35 | 1.85 | 1.85      | 1.71        |
| CNN-7 + PDB | 2.09 | 1.76 | 1.77      | <b>1.65</b> |
| CNN-8       | 2.41 | 1.72 | 1.73      | 1.63        |
| CNN-8 + PDB | 2.22 | 1.63 | 1.64      | <b>1.59</b> |

Bold values indicate the best performing loss function on CNN-5/6/7/8. It should be noted that data augmentation is applied to all the settings

more training samples to the  $k$ th subset by randomly sampling  $B^* - B_k$  samples from the original  $k$ th subset. Then all the subsets have the size of  $B^*$  and the total number of training samples is increased from  $\sum_{k=1}^K B_k$  to  $KB^*$ . It should be noted that we perform pose-based data balancing before network training by randomly duplicating some training samples of each subset of lower occupancy. Additionally, we modify each duplicated training image online with random image rotation, bounding box perturbation and other data augmentation approaches, as introduced in Sect. 8.1. After pose-based data balancing, the training samples of each mini-batch is randomly sampled from the balanced training dataset for network training. As the samples with different poses have the same probability to be sampled for a mini-batch, the network training is pose balanced.

We compare the performance of the four plain CNN architectures on the AFLW dataset in Table 3, using four different loss functions as well as the proposed PDB strategy. Note that, for a fair comparison, we also apply data augmentation to the training samples when we train a network without PDB. From the table, we can see that PDB improves the performance of all different CNN architectures in accuracy, in spite of the choice of loss functions.



**Fig. 8** The coarse-to-fine facial landmark localisation framework

### 7 Coarse-to-Fine Localisation Network

Besides out-of-plane head rotations, the accuracy of a facial landmark localisation algorithm can be degraded by other factors, such as in-plane rotations and inaccurate bounding boxes output from a poor face detector. To address these issues, we can stack or cascade multiple networks to form a coarse-to-fine structure. In fact, this technique has been widely used in the community. For example, Huang et al. (2015) proposed to use a global network to obtain coarse facial landmarks for transforming a face to the canonical view and then applies multiple networks trained on different facial parts for landmark refinement. Similarly, both Yang et al. (2017a) and Deng et al. (2019b) proposed to train a network that predicts a small number of facial landmarks (5 or 19) to transform the face to a canonical view. Because the first network can be trained on a large-scale dataset, such as CelebA (Liu et al. 2015) and UMDFaces (Bansal et al. 2017), it performs well for unconstrained faces with in-plane head rotation, scale and translation. With the normalised faces from the first stage, the performance of subsequent networks trained on a small dataset with all the facial landmarks is boosted. However, there are two outstanding issues in the use of a multi-stage network. First, one should question its effectiveness. Does a multi-stage network perform better than a single large-capacity network that has more parameters? The second important issue is whether stacking multiple networks would slow down the speed of the network. In other words, how can a multi-stage network be used in the most efficient way?

In this section, we answer these two questions using a coarse-to-fine network as depicted in Fig. 8. Given a fixed neural network architecture, the network trained on a dataset exhibiting wide diversity usually has a better generalisation capacity but achieves lower accuracy. In contrast, the network trained on a dataset with less diversity usually performs better for the cohorts involved in the training but is not able to generalise well for untrained cohorts. To achieve good performance in terms of both generalisation capability and accuracy, we need a large-capacity model and a large-scale

**Table 4** A comparison of the performance of the single-stage and coarse-to-fine network architectures, measured in terms of the Normalised Mean Error (NME) ( $\times 10^{-2}$ ) on AFLW

| Model | Single stage |       |       |       | Coarse-to-fine |         |         |         |         |         |
|-------|--------------|-------|-------|-------|----------------|---------|---------|---------|---------|---------|
|       | CNN-5        | CNN-6 | CNN-7 | CNN-8 | CNN-5/5        | CNN-6/6 | CNN-7/7 | CNN-8/8 | CNN-6/7 | CNN-6/8 |
| NME   | 2.01         | 1.75  | 1.65  | 1.59  | 1.94           | 1.65    | 1.57    | 1.54    | 1.57    | 1.54    |

The RWing loss and pose-based data balancing are used for network training

dataset with a large number of labelled training samples. However, the collection of such a face dataset with manually annotated facial landmarks is very expensive and tedious. An alternative is to train multiple stacked networks, *e.g.* the proposed coarse-to-fine localisation network.

The coarse network is trained on a dataset with very heavy data augmentation by randomly rotating an original training image between  $[-180^\circ, 180^\circ]$  and perturbing the bounding box with 20% of the original bounding box size. Such a trained network is able to perform well for large in-plane head rotations as well as low-quality face bounding boxes. For the second network training, we feed each heavily augmented training sample to the first trained network and obtain its facial landmarks. Then two anchor points (blue points in Fig. 8) are defined using these landmarks to perform rigid transformation. For AFLW, the mean of four inner eye and eyebrow corners is used as the first anchor point and the landmark on the chin is used as the second anchor point. After that, we inject a light data augmentation by randomly rotating the image between  $[-10^\circ, 10^\circ]$  and perturbing the bounding box with 10% of the bounding box size. Then, the second network is trained using a dataset with less in-plane rotations and high-quality face bounding boxes hence it is able to perform better in terms of accuracy. The joint use of these two networks in a coarse-to-fine fashion is instrumental in enhancing the generalisation capacity as well as accuracy.

We compare the four single-stage CNN plain networks with our coarse-to-fine networks in Table 4, in terms of NME. We can see that the use of our coarse-to-fine framework improves the accuracy of the original plain network at the expense of doubling the network inference time. The speed of each network is reported in Table 6. In addition, the use of two small networks performs better than a single large-capacity network. For example, the model sizes of CNN-6 and CNN-7 are 40MB and 160MB respectively (Table 1). The size of CNN-7 is four times that of CNN-6. When we stack two CNN-6 networks, the size of CNN-7 is still twice that of CNN-6/6. However, the accuracy obtained by the coarse-to-fine CNN-6/6 is better than the single CNN-7 network. The same conclusion can be drawn from the comparison between CNN-7/7 and CNN-8. Moreover, we do not, in fact, need a large-capacity network for the first stage because we only use it to perform coarse facial landmark localisation. We can use a lightweight network, *e.g.* CNN-6, for the first stage and

then cascade a large-capacity network for landmark refinement. According to Table 4, CNN-6/7 and CNN-6/8 perform as well as CNN-7/7 and CNN-8/8.

## 8 Experimental Results

In this section, we first introduce the implementation details and experimental settings of the proposed method. Second, we conduct an ablation study of its different components. Last, we compare our method with the state-of-the-art algorithms on four well-known benchmarking datasets, *i.e.* the Caltech Occluded Faces in the Wild (COFW) dataset (Burgos-Artizzu et al. 2013), the Annotated Facial Landmarks in the Wild (AFLW) dataset (Koestinger et al. 2011), the Wider Facial Landmarks in-the-wild (WFLW) dataset (Wu et al. 2018) and the 300 faces in-the-Wild (300W) dataset (Sagonas et al. 2016).

### 8.1 Implementation Details

For our experiments, we adopted Matlab 2019a and the MatConvNet toolbox<sup>2</sup> for network training and evaluation. The experiments were conducted on a server running Ubuntu 16.04 with  $2 \times$  Intel Xeon Gold 6134 CPU @3.20 GHz, 188 GB RAM and three NVIDIA GeForce RTX 2080Ti cards. Note that we only use one GPU card for measuring the speed of a network with a batch size of 1. Additionally, due to the low efficiency of MatConvNet on new GPU devices and CUDA versions, our speed benchmarking was measured by using PyTorch.

For network training, we set the weight decay to  $5 \times 10^{-4}$ , momentum to 0.9 and batch size to 16. In our plain networks, *i.e.* CNN-5/6/7/8, the standard ReLU function was chosen for nonlinear activation, and the 2D  $2 \times 2$  Max pooling with the stride of 2 was applied to downsize the feature maps. For a convolutional layer, we used  $3 \times 3$  kernels with the stride of 1. All the networks, including CNN-5/6/7/8, MobileNet-V1 Howard et al. (2017), MobileNet-V2 Sandler et al. (2018), VGG-16 Parkhi et al. (2015) and ResNet-50 He et al. (2017), were trained from scratch without any pre-training on any other dataset. This is different from the original Wing loss

<sup>2</sup> <http://www.vlfeat.org/matconvnet/>

paper, in which the ResNet-50 model was pre-trained on ImageNet Feng et al. (2018b). For the proposed PDB strategy, the number of bins  $K$  was set to 18.

The learning rate was fine-tuned for each network and loss function. To be more specific, we set the initial learning rate to a suitable value and then reduce it linearly across all the epochs to a value that is  $10^{-2}$  of the initial learning rate. For example, for CNN-6, we reduced the learning rate from  $3 \times 10^{-4}$  to  $3 \times 10^{-6}$  for L2 and from  $3 \times 10^{-3}$  to  $3 \times 10^{-5}$  for the other loss functions. The parameters of the RWing loss were set to  $w = 5/10/20$  and  $\epsilon = 0.5/1/2.5$  for CNN-5/6/7,  $w = 40$  and  $\epsilon = 5$  for CNN-8, MobileNet-V1, MobileNet-V2, VGG-16 and ResNet-50. The parameter used for the rectified region,  $r$ , was set to 0.5% of the size of an input image of each network.

To perform online data augmentation, we randomly applied image rotation, bounding box perturbation, left-right image flipping, Gaussian blur, etc. to each training image with the probability of 50%. For bounding box perturbation, we applied random translations to the upper-left and bottom-right corners of the original face bounding box given for a training sample.

To evaluate the performance of a facial landmark localisation algorithm, we adopted the widely used Normalised Mean Error (NME) metric. For the COFW dataset, the NME metric was normalised by the inter-pupil distance. For the AFLW dataset, we followed the protocol used in Zhu et al. (2016a), in which the NME was normalised by the face bounding box size. For the WFLW dataset, we followed the protocol used in Wu et al. (2018), in which the inter-ocular distance is used to perform normalisation. For the 300W dataset, NME was normalised by the outer eye corner distance. Additionally, the Area Under the Curve (AUC) and failure rate metrics were also used for benchmarking an algorithm on WFLW, 300W and COFW. AUC is defined as the area under the cumulative error distribution curve. The failure rate is defined as the proportion of the test images with NME higher than  $10 \times 10^{-2}$  NME.

## 8.2 Ablation Study

In this section, we perform an ablation study of the proposed method. Note that, some results have already been reported in Sects. 4–7 to validate the effectiveness of each component of the proposed method, namely the new RWing loss, Pose-based Data Balancing (PDB) and the coarse-to-fine network architecture.

### 8.2.1 RWing Loss for Other Network Architectures

In Sect. 5, we demonstrate that the use of our proposed RWing loss function improves the performance of different plain CNN networks, *i.e.* our CNN-5/6/7/8, in terms of

**Table 5** A comparison of the performance of MobileNet-V1 (Howard et al. 2017), MobileNet-V2 (Sandler et al. 2018), VGG-16 (Simonyan et al. 2014) and ResNet-50 (He et al. 2017) using different loss functions

| Method       | NME ( $\times 10^{-2}$ ) |      |           |             |
|--------------|--------------------------|------|-----------|-------------|
|              | L2                       | L1   | smooth L1 | RWing       |
| <b>AFLW</b>  |                          |      |           |             |
| MobileNet-V1 | 1.86                     | 1.77 | 1.77      | <b>1.71</b> |
| MobileNet-V2 | 1.96                     | 1.82 | 1.83      | <b>1.78</b> |
| VGG-16       | 2.27                     | 1.79 | 1.81      | <b>1.62</b> |
| ResNet-50    | 1.77                     | 1.57 | 1.58      | <b>1.51</b> |
| <b>WFLW</b>  |                          |      |           |             |
| MobileNet-V1 | 6.12                     | 5.55 | 5.51      | <b>5.27</b> |
| MobileNet-V2 | 6.32                     | 5.54 | 5.46      | <b>5.30</b> |
| VGG-16       | 9.09                     | 6.18 | 6.15      | <b>5.57</b> |
| ResNet-50    | 6.05                     | 5.26 | 5.25      | <b>4.99</b> |

Bold values indicate the best performing loss function on various CNN architectures

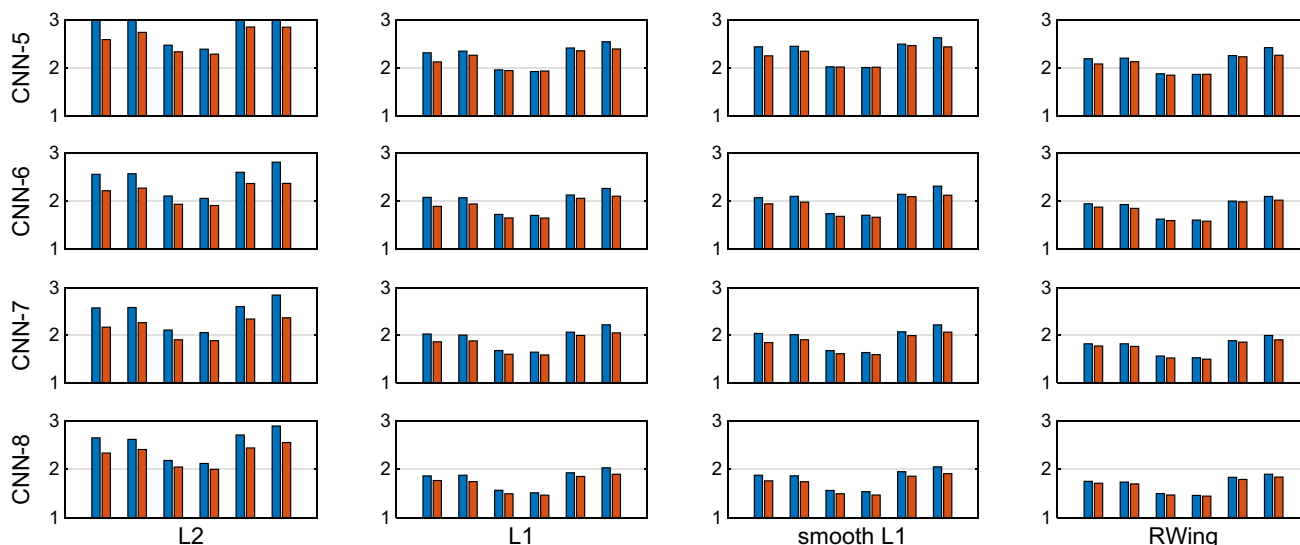
The performance was evaluated on the AFLW and WFLW datasets in terms of the Normalised Mean Error (NME). Note that all these networks were trained from scratch

accuracy. However, one may question the effectiveness of RWing loss for other CNN architectures, especially for some newly developed lightweight networks and large-capacity networks. To close this gap, we evaluate the performance of RWing loss using MobileNet-V1 Howard et al. (2017), MobileNet-V2 Sandler et al. (2018) VGG-16 Simonyan et al. (2014) and ResNet-50 He et al. (2017) on the AFLW and WFLW datasets. The input for MobileNet-V1/V2, VGG-16 and ResNet-50 is a  $224 \times 224 \times 3$  colour image. All these four networks were trained from scratch using the training samples of AFLW or WFLW only. Data augmentation was performed online for all the samples in each mini-batch as introduced in Sect. 8.1.

The results are reported in Table 5. As shown in the table, the use of the newly proposed RWing loss outperforms all the other loss functions in terms of accuracy, which further demonstrates the generalisation capacity of our RWing loss to other network architectures, including both lightweight networks, *i.e.* MobileNet-V1 and V2, and large-capacity networks, *i.e.* VGG-16 and ResNet-50. In particular, for the VGG-16 network, the use of our RWing loss reduces the error by around 30% as compared with L2 and 10% as compared with the L1 and smooth L1 loss functions.

### 8.2.2 Pose-based data balancing for near frontal faces

The aim of Pose-based Data Balancing (PDB) presented in Sect. 6 is to deal with extreme out-of-plane pose variations. In fact, PDB increases the proportion of large poses in the population during training. With this technique, one may wonder



**Fig. 9** A comparison of different network architectures and loss functions using the normalised mean error ( $\times 10^{-2}$ ) parameterised by pose. We split the test set into 6 cohorts,  $[-90, -60]$ ,  $[-60, -30]$ ,  $[-30, 0]$ ,  $[0, 30]$ ,  $[30, 60]$  and  $[60, 90]$ , using their projected pose space coef-

ficients. For each cohort, the left blue bar stands for a model trained without the PDB strategy, and the right red bar for a model trained with PDB

whether it will degrade the performance of a trained network for the test samples with small out-of-plane head rotations. To examine this, we perform an evaluation using four different plain CNN networks as well as four different loss functions on the AFLW dataset. The evaluation is conducted by splitting the 4386 test images of AFLW-Full into six different cohorts based on their projected pose coefficients.

The evaluation results are shown in Fig. 9. From the figure, we can confidently say that the proposed PDB approach is not only able to increase the accuracy of the trained network for the test samples with large out-plane head rotations, but also to maintain or even increase the performance for the test samples with small pose variations.

### 8.2.3 Balancing the Speed and Accuracy

Facial landmark localisation has been widely used in many real-time practical applications, hence the speed together with accuracy of an algorithm is crucial for the deployment of the algorithm in commercial use cases. However, the use of a more accurate model usually brings the increase in the cost of the inference time. In this section, we compare the performance of different networks on the AFLW dataset in terms of both accuracy and speed. The aim is to provide a better guidance for the selection of a proper model for a specific practical application. To this end, we compare different networks in terms of network parameters, model size, FLOPs, speed as well as accuracy in Table 6. The speed of each model was tested on CPU, GPU and two mobile devices as listed in the table. For each model, the proposed RWing loss function and PDB strategy were used for model training.

According to the results reported in Table 6, for a real-time application deployed on a device without GPU support, we suggest the CNN-6 model. The CNN-6 model has the accuracy of  $1.75 \times 10^{-2}$  in terms of NME that is even better than most of the state-of-the-art methods in Tables 2 and 9. More importantly, CNN-6 is super fast, which runs at 2200 fps on an NVIDIA GeForce RTX 2080Ti GPU and 170 fps on an Intel Xeon Gold 6134 CPU. CNN-6 is much faster than most existing DNN-based facial landmark localisation approaches such as MobileNets and TR-DRN Lv et al. (2017). The speed of TR-DRN is only 83 fps on an NVIDIA GeForce GTX Titan X card. Even on mobile devices, CNN-6 still able to run at a good speed, such as 370/13.8 fps on the GPU/CPU of NVIDIA Jetson TX2. However, for some low-cost mobile devices such as Raspberry Pi 4, none of the models listed in the table is able to run in real time. In this case, we have to further sacrifice the performance to improve the speed. For example, one can use the CNN-5 model that runs at 19 fps on Raspberry Pi 4 and 25 fps on the CPU of NVIDIA Jetson TX2.

For a software that does not require the real-time inference speed, the ResNet-50 trained using our RWing loss and PDB strategy is advocated because it brings the best accuracy in facial landmark detection. A well-balanced model is our coarse-to-fine CNN-6/8 model that has similar performance as ResNet-50 in accuracy but runs much faster than ResNet-50 on GPU. Suppose we have a real-time application running on a device with a powerful GPU, our CNN-6/8 would be the best choice. It runs at 1010 fps on an NVIDIA GeForce RTX 2080Ti card whereas ResNet-50 only runs at 154 fps. In general, a real-time facial analysis system usually

**Table 6** A comparison of different networks, in terms of the number of model parameters, model size, floating point operations (FLOPs), speed on four types of devices and accuracy, conducted on the AFLW dataset using the AFLW-Full protocol

| Network      | Model parameters | Model size | FLOPs   | Speed (fps) |     |               |            |      | NME ( $\times 10^{-2}$ ) |
|--------------|------------------|------------|---------|-------------|-----|---------------|------------|------|--------------------------|
|              |                  |            |         | GPU         | CPU | Mobile Device |            |      |                          |
|              |                  |            |         |             |     | Raspberry Pi4 | Jetson TX2 |      |                          |
|              |                  |            |         | GPU         | CPU |               |            |      |                          |
| CNN-5        | 2.6 M            | 10 MB      | 0.06 G  | 2600        | 310 | 19            | 460        | 25   | 2.01                     |
| CNN-6        | 10 M             | 40 MB      | 0.33 G  | 2200        | 170 | 5.6           | 370        | 13.8 | 1.75                     |
| CNN-7        | 42 M             | 160 MB     | 1.61 G  | 2020        | 50  | 1.47          | 350        | 3.62 | 1.65                     |
| CNN-8        | 168 M            | 640 MB     | 7.71 G  | 1860        | 10  | 0.39          | 300        | 0.83 | 1.59                     |
| CNN-5/5      | 5.2 M            | 20 MB      | 0.12 G  | 1300        | 155 | 9.5           | 230        | 12.5 | 1.94                     |
| CNN-6/6      | 20 M             | 80 MB      | 0.66 G  | 1100        | 85  | 2.8           | 185        | 6.93 | 1.65                     |
| CNN-7/7      | 84 M             | 320 MB     | 3.22 G  | 1010        | 25  | 0.74          | 175        | 1.81 | 1.57                     |
| CNN-8/8      | 336 M            | 1280 MB    | 15.42 G | 930         | 5   | 0.19          | 150        | 0.42 | 1.54                     |
| CNN-6/7      | 52 M             | 200 MB     | 1.94 G  | 1050        | 38  | 1.16          | 180        | 2.87 | 1.57                     |
| CNN-6/8      | 178 M            | 680 MB     | 8.04 G  | 1010        | 10  | 0.36          | 165        | 0.78 | 1.54                     |
| MobileNet-V1 | 4.3 M            | 16.5 MB    | 0.55 G  | 330         | 32  | 0.22          | 66         | 0.25 | 1.71                     |
| MobileNet-V2 | 3.6 M            | 14 MB      | 0.32 G  | 170         | 19  | 0.15          | 34         | 0.17 | 1.78                     |
| VGG-16       | 134 M            | 512 MB     | 16.79 G | 950         | 13  | 0.24          | 202        | 0.61 | 1.62                     |
| ResNet-50    | 32 M             | 122 MB     | 3.87 G  | 154         | 12  | 0.62          | 26         | 0.7  | 1.51                     |

The accuracy is measured in terms of Normalised Mean Error (NME) ( $\times 10^{-2}$ ). The speed of each network was evaluated by PyTorch and only the network inference time was used to calculate the speed. The speed on GPU was tested using an NVIDIA GeForce RTX 2080Ti card and the speed on CPU was tested using Intel Xeon Gold 6134 @3.2GHz. For mobile devices, we used two power-efficient embedded computing devices: the Raspberry Pi-4 platform that has only a quad-core Broadcom BCM2711 Cortex-A72 CPU @1.5GHz with 4GB RAM; the NVIDIA Jetson TX2 card that has a 256-core GPU @1.3GHz, a quad-core ARM Cortex-A57 CPU @2GHz+ and a dual-core NVIDIA Denver2 CPU @2GHz with 8GB RAM

has to perform multiple tasks, such as face detection and face recognition, jointly. Additionally, a video frame may contain multiple faces. In such a case, the joint use of all those components may not be able to achieve video rate if we use ResNet-50. Despite the significant difference between CNN-6/8 and ResNet-50 in speed (GPU), the accuracy of CNN-6/8 is comparable with ResNet-50.

#### 8.2.4 Sensitivity Analysis of the RWing Parameters

The key innovation of the proposed RWing loss function is the non-linear region that boosts the impact of the training samples with small-medium errors. The extent of this impact is controlled by two parameters,  $w$  and  $\epsilon$ , that change the width and curvature of the non-linear region, respectively. As mentioned in Sect. 5, we should not set  $\epsilon$  to a very small value because it makes the training of a network unstable and may cause the exploding gradient problem for small errors. However, a pertinent question is: what does constitute a proper value for  $\epsilon$ ?

To answer this question, we compared the performance of different parameter settings for  $\epsilon$  and  $w$ , using the CNN-6 model. The experiments were conducted on the AFLW dataset and measured in terms of NME. The results are

**Table 7** A comparison of the impact of different parameter settings for  $w$  and  $\epsilon$  on the performance, evaluated on the AFLW dataset using the CNN-6 model in terms of NME ( $\times 10^{-2}$ )

| $\epsilon$ | $w$  |      |             |             |             |             |             |
|------------|------|------|-------------|-------------|-------------|-------------|-------------|
|            | 4    | 6    | 8           | 10          | 12          | 14          | 16          |
| 0.5        | 1.80 | 1.79 | <b>1.77</b> | 1.78        | 1.79        | 1.80        | 1.82        |
| 1          | 1.81 | 1.79 | 1.78        | <b>1.77</b> | <b>1.77</b> | <b>1.77</b> | <b>1.77</b> |
| 2          | 1.87 | 1.82 | 1.80        | 1.79        | <b>1.77</b> | <b>1.77</b> | <b>1.77</b> |
| 3          | 1.91 | 1.86 | 1.82        | 1.81        | 1.79        | 1.78        | <b>1.77</b> |
| 4          | 1.95 | 1.89 | 1.84        | 1.82        | 1.81        | 1.79        | 1.79        |

Bold values indicate the best performing parameter pairs,  $\epsilon$  and  $w$   
Note that we did not use PDB for network training in this experiment

reported in Table 7. We can see that, almost all the combinations of different values of the two parameters perform better than the classical L2 ( $2.33 \times 10^{-2}$ ), L1 ( $1.91 \times 10^{-2}$ ) and smooth L1 ( $1.93 \times 10^{-2}$ ) loss functions as reported in Table 2, in terms of NME. More importantly, the best result ( $1.77 \times 10^{-2}$ ) can be found by evaluating many different combinations of the two parameter values. The sensitivity analysis demonstrates that the behaviour of the network is quite stable as a function of the loss function parameters.

### 8.3 Comparison with the State-of-The-Art Methods

In this section, we compare the proposed method with the state-of-the-art approaches on four benchmarks, *i.e.* COFW Burgos-Artizzu et al. (2013), AFLW Koestinger et al. (2011), WFLW Wu et al. (2018) and 300W Sagonas et al. (2016). To this end, we use three different CNN architectures, including the single-stage CNN-6 model, the coarse-to-fine CNN-6/8 model and the ResNet-50 model. All these three models were trained from scratch using the training set provided by each benchmark. Note that no external data was used for our network training.

#### 8.3.1 Evaluation on COFW

We first evaluate our methods on the Caltech Occluded Faces in the Wild (COFW) dataset (Burgos-Artizzu et al. 2013) that is a widely used benchmark for facial landmark localisation algorithms. COFW is an extension of the original Labelled Facial Parts in the Wild (LFPW) dataset (Belhumeur et al. 2011), by adding more training and test examples with heavy occlusions. The COFW benchmark has 1345 training and 507 test images. Each facial image in COFW was manually annotated with 29 landmarks. We followed the standard protocol of COFW and report the performance of our approaches in Table 8 using two different metrics, *i.e.* normalised mean error and failure rate.

As shown in Table 8, our simple and fast CNN-6 model powered by the RWing loss and PDB strategy outperforms all the other state-of-the-art approaches in terms of both NME and failure rate. Note that COFW focuses on benchmarking the robustness of a facial landmark localisation algorithm for in-the-wild faces with heavy occlusions. Many state-of-the-art approaches listed in the table, *e.g.* RSR Cui et al. (2019), HOSRD Xing et al. (2018) and RAR Xiao et al. (2016), use some specific techniques to deal with the challenge posed by occlusions. In contrast, our CNN models do not use any specific trick to address the occlusion problem. This further illustrates the advantages of the proposed approach. Last, with the coarse-to-fine CNN-6/8 model and the ResNet-50 model, we further improve the performance on the COFW dataset. The NME is reduced by around 10% and 15% as compared with the best state-of-the-art result achieved by RSR when using ResNet-50 and CNN-6/8 powered by our RWing loss and the PDB strategy.

One interesting finding is that our CNN-6/8 performs much better than ResNet-50 on COFW. The reason for this is twofold. First, the training set of COFW has only 1345 facial images, which is a typical small sample size problem for CNN training. In such a case, the use of our coarse-to-fine network strategy is superior over a large-capacity network, *e.g.* ResNet, that usually requires a large number of training samples for successful network training. Second, the face

**Table 8** A comparison of our CNN models powered by RWing loss and PDB with state-of-the-art methods, including Zhu et al. (2012), Cao et al. (2014), RCPR Burgos-Artizzu et al. (2013), HPM Ghiasi and Fowlkes (2014), RCRC Feng et al. (2015b), CCR Feng et al. (2015a), Zeng et al. (2018), HOSRD Xing et al. (2018), Wu et al. (2017a), RAR Xiao et al. (2016), DAC-CSR Feng et al. (2017b) and RSR Cui et al. (2019), on COFW, measured in terms of NME ( $\times 10^{-2}$ ) by face size, failure rate (%) and speed (fps)

| Method                     | Metric      |              |             |             |
|----------------------------|-------------|--------------|-------------|-------------|
|                            | NME         | Failure Rate | Speed (fps) |             |
|                            |             |              | CPU         | GPU         |
| Zhu et. al.                | 14.4        | 80           | 0.025       | –           |
| Cao et. al.                | 11.2        | 36           | 67          | –           |
| RCPR                       | 8.50        | 20           | 3           | –           |
| HPM                        | 7.50        | 13           | 0.03        | –           |
| RCRC                       | 7.30        | 12           | 22          | –           |
| CCR                        | 7.03        | 10.9         | 69          | –           |
| RAR (ECCV-16)              | 6.03        | 4.14         | –           | 4           |
| Wu et. al. (CVPR-17)       | 6.40        | –            | –           | –           |
| DAC-CSR (CVPR-17)          | 6.03        | 4.73         | 10          | –           |
| Zeng et. al. (TIP-18)      | 8.10        | 19           | –           | 157         |
| HOSRD (TPAMI-18)           | 6.80        | 13           | 7           | –           |
| RSR (TPAMI-19)             | 5.63        | –            | –           | –           |
| CNN-6 (RWing+PDB)          | 5.44        | 3.75         | <b>170</b>  | <b>2200</b> |
| <b>CNN-6/8 (RWing+PDB)</b> | <b>4.80</b> | <b>3.16</b>  | 10          | 1010        |
| ResNet-50 (RWing+PDB)      | 5.07        | <b>3.16</b>  | 12          | 154         |

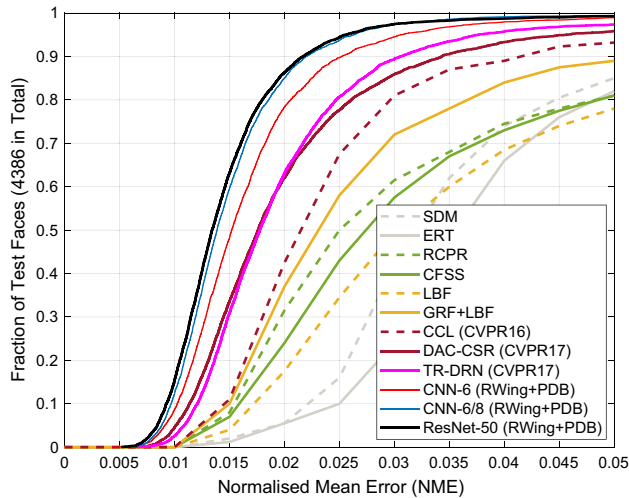
Bold values indicate the best performance

bounding boxes of the test samples of COFW are very different from those of the training samples. Our coarse-to-fine network can deal with this problem effectively. This further demonstrates the merit of the proposed coarse-to-fine landmark localisation system.

#### 8.3.2 Evaluation on AFLW

For the AFLW dataset Koestinger et al. (2011), we follow the protocol used in Zhu et al. (2016a). AFLW is a very challenging dataset that has been widely used for benchmarking facial landmark localisation algorithms. The images in AFLW consist of a wide range of pose variations in yaw (from  $-90^\circ$  to  $90^\circ$ ), as shown in Fig. 7. The protocol used in Zhu et al. (2016a) defines 20,000 training and 4,386 test images, and each image has 19 manually annotated facial landmarks. The evaluation is performed using two different settings: AFLW-Full and AFLW-Frontal. AFLW-Full evaluates an algorithm using all the test images, whereas AFLW-Frontal evaluates an algorithm using only near-frontal faces.

For the AFLW-Full setting, we first compare the proposed method with a set of state-of-the-art approaches in terms of accuracy in Fig. 10, using the Cumulative Error Distribution (CED) curve. Second, a further comparison with more



**Fig. 10** A comparison of the CED curves on the AFLW dataset with the AFLW-Full protocol. We compare our method with a set of state-of-the-art approaches, including SDM Xiong and Torre (2013), ERT Kazemi and Sullivan (2014), RCPR Burgos-Artizzu et al. (2013), CFSS Zhu et al. (2015), LBF Ren et al. (2016), GRF Hara and Chellappa (2014), CCL Zhu et al. (2016a), DAC-CSR Feng et al. (2017b) and TR-DRN Lv et al. (2017)

approaches is reported in Table 9, using both the AFLW-Full and AFLW-Frontal settings.

As shown in Fig. 10, again, our very simple and fast CNN-6 network outperforms all the other approaches. Second, by using the proposed coarse-to-fine network, *i.e.* CNN-6/8, the performance has been significantly improved in accuracy. The performance of our CNN-6/8 is very close to ResNet-50.

**Table 9** A comparison of our CNN models powered by RWin loss and PDB with state-of-the-art methods, including RCPR Burgos-Artizzu et al. (2013), CCL Zhu et al. (2016a), DAC-CSR Feng et al. (2017b), TR-DRN Lv et al. (2017), Zeng et al. (2018), CPM+SBR Dong et al. (2018b), SAN Dong et al. (2018a), GoDP Wu et al. (2018b), LAB Wu et al. (2018), ODN Zhu et al. (2019) and GHCU Liu et al. (2019), on AFLW, measured in terms of NME and speed

| Method                      | Metric                   |              |             |             |
|-----------------------------|--------------------------|--------------|-------------|-------------|
|                             | NME ( $\times 10^{-2}$ ) |              | Speed (fps) |             |
|                             | AFLW-full                | AFLW-frontal | CPU         | GPU         |
| RCPR                        | 3.73                     | 2.87         | 3           | –           |
| CCL                         | 2.72                     | 2.17         | <b>350</b>  | –           |
| DAC-CSR (CVPR-17)           | 2.27                     | 1.81         | 10          | –           |
| TR-DRN (CVPR-17)            | 2.17                     | –            | –           | 83          |
| Zeng et al. (TIP-18)        | 2.60                     | –            | –           | 157         |
| CPM+SBR (CVPR-18)           | 2.14                     | –            | –           | –           |
| SAN (CVPR-18)               | 1.91                     | 1.85         | –           | –           |
| LAB (CVPR-18)               | 1.85                     | 1.62         | –           | 17          |
| LAB* (with external data)   | <i>1.25</i>              | <i>1.14</i>  | –           | 17          |
| GoDP (IVC-18)               | 1.84                     | –            | –           | –           |
| ODN (CVPR-19)               | 1.63                     | 1.38         | –           | –           |
| GHCU (CVPR-19)              | 1.60                     | –            | –           | 7           |
| CNN-6 (RWin+PDB)            | 1.75                     | 1.47         | 170         | <b>2200</b> |
| CNN-6/8 (RWin+PDB)          | 1.54                     | 1.30         | 10          | 1010        |
| <b>ResNet-50 (RWin+PDB)</b> | <b>1.51</b>              | <b>1.27</b>  | 12          | 154         |

Bold values indicate the best performance

The values in italic draw attention to the use of external data

Note that the ResNet-50 model was trained using our RWin loss and PDB strategy. Otherwise, ResNet-50 would perform worse than our coarse-to-fine CNN-6/8, as evidenced by the results in Tables 5 and 4.

In Table 9, we compare our method with more state-of-the-art approaches on both the AFLW-Full and AFLW-Frontal settings. The proposed method improves the performance in accuracy over the state-of-the-art approaches. For example, in contrast to GHCU Liu et al. (2019), the ResNet-50 model trained with our RWin loss and PDB strategy reduces the NME from  $1.60 \times 10^{-2}$  to  $1.51 \times 10^{-2}$ , which is circa a 6% decrease in normalised mean error on the AFLW dataset. Note that the best result is achieved by LAB\* Wu et al. (2018) but it was trained with external data to obtain boundary heatmaps.

### 8.3.3 Evaluation on WFLW

The WFLW dataset is a newly annotated dataset for facial landmark localisation Wu et al. (2018). The whole WFLW dataset has 10,000 facial images, in which 7500 images are used as the training set and the remaining 2500 images are used for test. Each image in the WFLW dataset was manually annotated with 98 facial landmarks. Considering the number of facial landmarks, WFLW is the current largest dataset that has 980 K manually annotated landmarks, which is higher than the  $19 \times 24386 \approx 460K$  landmarks of AFLW. To benchmark a facial landmark localisation approach on the WFLW dataset, three evaluation metrics are used, namely AUC,

**Table 10** A comparison of the performance of our CNN models powered by the proposed RWing loss function and PDB strategy with a number of state-of-the-art methods on the WFLW dataset, measured in terms of three evaluation metrics: AUC (the higher the better), NME (the lower the better) and failure rate (the lower the better)

| Metric                       | Method                       | Full Set     | Subset       |              |              |              |              |              |
|------------------------------|------------------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|                              |                              |              | Pose         | Expression   | Illumination | Makeup       | Occlusion    | Blur         |
| AUC@0.1 ( $\times 10^{-2}$ ) | ESR Cao et al. (2014)        | 27.74        | 1.77         | 19.81        | 29.53        | 24.85        | 19.46        | 22.04        |
|                              | SDM Xiong and Torre (2013)   | 30.02        | 2.26         | 22.93        | 32.37        | 31.25        | 20.60        | 23.98        |
|                              | CFSS Zhu et al. (2015)       | 36.59        | 6.32         | 31.57        | 38.54        | 36.91        | 26.88        | 30.37        |
|                              | DVLN Wu et al. (2017b)       | 45.51        | 14.74        | 38.89        | 47.43        | 44.94        | 37.94        | 39.73        |
|                              | LAB Wu et al. (2018)         | 53.23        | 23.45        | 49.51        | 54.33        | 53.94        | 44.90        | 46.30        |
|                              | CNN-6 (RWing+PDB)            | 50.80        | 26.80        | 46.29        | 51.61        | 48.75        | 44.69        | 46.11        |
|                              | CNN-6/8 (RWing+PDB)          | 51.82        | 28.95        | 46.48        | 51.83        | 51.02        | 45.55        | 45.62        |
|                              | <b>ResNet-50 (RWing+PDB)</b> | <b>55.85</b> | <b>33.09</b> | <b>49.79</b> | <b>56.31</b> | <b>54.60</b> | <b>49.85</b> | <b>50.10</b> |
| NME ( $\times 10^{-2}$ )     | ESR Cao et al. (2014)        | 11.13        | 25.88        | 11.47        | 10.49        | 11.05        | 13.75        | 12.20        |
|                              | SDM Xiong and Torre (2013)   | 10.29        | 24.10        | 11.45        | 9.32         | 9.38         | 13.03        | 11.28        |
|                              | CFSS Zhu et al. (2015)       | 9.07         | 21.36        | 10.09        | 8.30         | 8.74         | 11.76        | 9.96         |
|                              | DVLN Wu et al. (2017b)       | 6.08         | 11.54        | 6.78         | 5.73         | 5.98         | 7.33         | 6.88         |
|                              | LAB Wu et al. (2018)         | 5.27         | 10.24        | 5.51         | 5.23         | <b>5.15</b>  | 6.79         | 6.32         |
|                              | CNN-6 (RWing+PDB)            | 5.92         | 10.46        | 6.41         | 5.82         | 6.23         | 7.54         | 6.61         |
|                              | CNN-6/8 (RWing+PDB)          | 5.60         | 9.79         | 6.16         | 5.54         | 6.65         | 7.05         | 6.41         |
|                              | <b>ResNet-50 (RWing+PDB)</b> | <b>4.99</b>  | <b>8.43</b>  | <b>5.21</b>  | <b>4.88</b>  | 5.26         | <b>6.21</b>  | <b>5.81</b>  |
| Failure Rate (%)             | ESR Cao et al. (2014)        | 35.24        | 90.18        | 42.04        | 30.80        | 38.84        | 47.28        | 41.40        |
|                              | SDM Xiong and Torre (2013)   | 29.40        | 84.36        | 33.44        | 26.22        | 27.67        | 41.85        | 35.32        |
|                              | CFSS Zhu et al. (2015)       | 20.56        | 66.26        | 23.25        | 17.34        | 21.84        | 32.88        | 23.67        |
|                              | DVLN Wu et al. (2017b)       | 10.84        | 46.93        | 11.15        | 7.31         | 11.65        | 16.30        | 13.71        |
|                              | LAB Wu et al. (2018)         | 7.56         | 28.83        | 6.37         | 6.73         | <b>7.77</b>  | 13.72        | 10.74        |
|                              | CNN-6 (RWing+PDB)            | 9.88         | 38.65        | 12.10        | 9.31         | 11.65        | 18.21        | 12.94        |
|                              | CNN-6/8 (RWing+PDB)          | 8.24         | 34.36        | 9.87         | 7.16         | 9.71         | 15.22        | 10.61        |
|                              | <b>ResNet-50 (RWing+PDB)</b> | <b>5.64</b>  | <b>23.31</b> | <b>4.14</b>  | <b>4.87</b>  | 8.74         | <b>11.69</b> | <b>7.50</b>  |

Bold values indicate the best performance

NME and failure rate, as introduced at the end of Sect. 8.1. In addition, the WFLW dataset further divides the whole test set into 6 different subsets labelled by different challenging attributes, including pose, expression, illumination, makeup, occlusion and image blur. This provides a better understanding of the behaviour of a facial landmark localisation method under different challenging scenarios.

We compare the proposed method with a number of state-of-the-art approaches in Table 10, in terms of AUC, NME and failure rate. As shown in the table, our single-stage CNN-6 and coarse-to-fine CNN-6/8 networks, equipped with the RWing loss and PDB, perform well on both the full set and subset evaluations. CNN-6 and CNN-6/8 outperform most of the existing state-of-the-art approaches in terms of AUC, NME and failure rate, but worse than LAB that has the best performance reported in the existing literature on the WFLW dataset. However, the network architecture of LAB is very complicated. LAB runs at around 16 fps on a Titan X GPU, which is much slower than the speed of our CNN-6 (2200

fps) or CNN-6/8 (1010 fps). When we switch the backbone network to ResNet-50, that has the speed of 154 fps on GPU, we are able to beat all the other approaches on the full set and most of the subsets in terms of all the three evaluation metrics. It should be noted that the training of ResNet-50 is based on the proposed RWing loss and PDB. Without those two innovative elements, the performance of ResNet-50 would be worse than LAB. The results obtained on the WFLW dataset demonstrate the efficiency and robustness of the proposed method further.

### 8.3.4 Evaluation on 300W

The 300W dataset is a collection of multiple face datasets, including LFPW Belhumeur et al. (2011), HELEN Le et al. (2012), AFW Zhu et al. (2012), FRGC Phillips et al. (2005), XM2VTS Messer et al. (1999) and another 135 unconstrained faces collected from the Internet. For testing, 600 unconstrained facial images, including 300 indoor and 300



**Table 11** A comparison of our CNN models powered by RWing loss and PDB with the state-of-the-art methods, including Uříčář et al. (2016), Čech et al. (2016), Martinez and Valstar (2016), Deng et al. (2016), Fan and Zhou (2016), MDM Trigeorgis et al. (2016), Densereg+MDM Alp Guler et al. (2017), LAB Wu et al. (2018) and JMFA Deng et al. (2019b), on the 300W dataset, evaluated in terms of three metrics: AUC ( $\times 10^{-2}$ ), failure rate (%) and speed (fps)

| Method                            | Metric       |              | Speed      |             |
|-----------------------------------|--------------|--------------|------------|-------------|
|                                   | AUC          | Failure rate | CPU        | GPU         |
| Uricar et. al.                    | 21.09        | 32.17        | 10         | –           |
| Cech et. al.                      | 22.18        | 33.83        | –          | –           |
| Martinez et. al.                  | 37.79        | 16.0         | –          | –           |
| Deng et. al.                      | 47.52        | 5.50         | 20         | –           |
| Fan et. al.                       | 48.02        | 14.83        | –          | –           |
| MDM (CVPR-16)                     | 45.32        | 6.80         | –          | –           |
| Densereg+MDM (CVPR-17)            | 52.19        | 3.67         | –          | –           |
| LAB (CVPR-18)                     | 58.85        | 0.83         | –          | 17          |
| JMFA (TIP-19)                     | 54.85        | 1.00         | –          | 50          |
| <i>JMFA* (with external data)</i> | <i>60.71</i> | <i>0.33</i>  | –          | 50          |
| CNN-6 (RWing+PDB)                 | 53.40        | 2.50         | <b>170</b> | <b>2200</b> |
| CNN-6/8 (RWing+PDB)               | 57.79        | 0.67         | 10         | 1010        |
| ResNet-50 (RWing+PDB)             | <b>59.23</b> | <b>0.50</b>  | 12         | 154         |

Bold values indicate the best performance

The values in italic draw attention to the use of external data

outdoor images, were collected. The face images involved in the 300W dataset were semi-automatically annotated by 68 facial landmarks Sagonas et al. (2013). We used the 600 300W test images to evaluate the proposed method and compared it with a number of state-of-the-art approaches, in terms of the Area Under the Curve (AUC), failure rate and speed. The results are reported in Table 11.

As shown in Table 11, even the simple CNN-6 model trained with our RWing loss and PDB achieves comparable results. The CNN-6 model only performs worse than LAB Wu et al. (2018) and JMFA Deng et al. (2019b) in terms of AUC and failure rate, but runs much faster than these two methods. The use of the proposed coarse-to-fine network, *i.e.* CNN6/8, improves the performance significantly in terms of AUC and failure rate. CNN6/8 only performs worse than LAB in terms of AUC and failure rate but with much faster inference speed. The ResNet-50 model trained with RWing loss and PDB beats JFMA and LAB in terms of all the evaluation metrics. However, the best result is achieved by JMFA\* that was trained with external data. JMFA\* achieves the highest AUC score but the training of JMFA\* uses the Menpo dataset that has 10993 near frontal faces and 3852 profile faces Deng et al. (2018).

## 9 Conclusion

In this paper, we analysed different loss functions that can be used for the task of regression-based facial landmark localisation. We found that L1 and smooth L1 loss functions perform much better in accuracy than the L2 loss function. Motivated by our analysis of these loss functions, we proposed

a new, RWing loss performance measure. The key idea of the RWing loss criterion is to increase the contribution of the samples with small and medium size errors to the training of the regression network. To prove the effectiveness of the proposed RWing loss function, extensive experiments have been conducted using several CNN network architectures. As shown in our experiments, by equipping a lightweight CNN network with the proposed RWing loss, it is able to achieve as good performance as large-capacity networks. Furthermore, a Pose-based Data Balancing (PDB) strategy and a coarse-to-fine landmark localisation framework were advocated to improve the accuracy of CNN-based facial landmark localisation further. We found that the proposed PDB strategy and coarse-to-fine framework can effectively deal with the difficulties posed by large-scale head rotations and small sample size problems, respectively. By evaluating our algorithm on multiple well-known benchmarking datasets, we demonstrated the merits of the proposed approach.

**Acknowledgements** This work was supported in part by the EPSRC Programme Grant (FACER2VM) EP/N007743/1, EPSRC/dstl/MURI project EP/R018456/1, the National Natural Science Foundation of China (61672265, U1836218, 61876072, 61902153).

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the

permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Alp Guler, R., Trigeorgis, G., Antonakos, E., Snape, P., Zafeiriou, S., & Kokkinos, I. (2017). Densereg: Fully convolutional dense shape regression in-the-wild. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 6799–6808).
- Bansal, A., Nanduri, A., Castillo, C. D., Ranjan, R., & Chellappa, R. (2017). Umdfaces: An annotated face dataset for training deep networks. In *IEEE international joint conference on biometrics (IJCB)* (pp. 464–473).
- Belhumeur, P. N., Jacobs, D. W., Kriegman, D., & Kumar, N. (2011). Localizing parts of faces using a consensus of exemplars. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 545–552).
- Bhagavatula, C., Zhu, C., Luu, K., & Savvides, M. (2017). Faster than real-time facial alignment: A 3D spatial transformer network approach in unconstrained poses. In *IEEE international conference on computer vision (ICCV)*.
- Bulat, A., & Tzimiropoulos, G. (2017a). Binarized convolutional landmark localizers for human pose estimation and face alignment with limited resources. In *IEEE international conference on computer vision (ICCV)*.
- Bulat, A., & Tzimiropoulos, G. (2017b). How far are we from solving the 2D & 3D face alignment problem? And a dataset of 230,000 3D facial landmarks. In *IEEE international conference on computer vision (ICCV)*.
- Burgos-Artizzu, X. P., Perona, P., & Dollár, P. (2013). Robust face landmark estimation under occlusion. In *IEEE international conference on computer vision (ICCV)* (pp. 1513–1520).
- Cao, X., Wei, Y., Wen, F., & Sun, J. (2014). Face alignment by explicit shape regression. *International Journal of Computer Vision (IJCV)*, 107(2), 177–190.
- Čech, J., Franc, V., Uříčář, M., & Matas, J. (2016). Multi-view facial landmark detection by using a 3d shape model. *Image and Vision Computing*, 47, 60–70.
- Cootes, T., Taylor, C., Cooper, D., Graham, J., et al. (1995). Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1), 38–59.
- Cootes, T. F., Walker, K., & Taylor, C. J. (2000). View-based active appearance models. In *IEEE international conference on automatic face and gesture recognition (FG)* (pp. 227–232).
- Cootes, T. F., Edwards, G., & Taylor, C. J. (2001). Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6), 681–685.
- Cootes, T. F., Wheeler, G. V., Walker, K. N., & Taylor, C. J. (2002). View-based active appearance models. *Image and Vision Computing*, 20(9), 657–664.
- Cristinacce, D., & Cootes, T. F. (2006). Feature Detection and Tracking with Constrained Local Models. *British Machine Vision Conference (BMVC)*, 3, 929–938.
- Cui, Z., Xiao, S., Niu, Z., Yan, S., & Zheng, W. (2019). Recurrent Shape Regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(5), 1271–1278.
- Deng, J., Liu, Q., Yang, J., & Tao, D. (2016). M3csr: Multi-view, multi-scale and multi-component cascade shape regression. *Image and Vision Computing*, 47, 19–26.
- Deng, J., Roussos, A., Chrysos, G., Ververas, E., Kotsia, I., Shen, J., & Zafeiriou, S. (2018). The menpo benchmark for multi-pose 2D and 3D facial landmark localisation and tracking. In *international journal of computer vision (IJCV)* (pp. 1–26).
- Deng, J., Guo, J., Xue, N., & Zafeiriou, S. (2019a). Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4690–4699).
- Deng, J., Trigeorgis, G., Zhou, Y., & Zafeiriou, S. (2019b). Joint multi-view face alignment in the wild. *IEEE Transactions on Image Processing*, 28(7), 3636–3648.
- Dollár, P., Welinder, P., & Perona, P. (2010). Cascaded pose regression. In *IEEE conference on computer vision and pattern recognition (CVPR)*, *IEEE* (pp. 1078–1085).
- Dong, X., Yan, Y., Ouyang, W., & Yang, Y. (2018a). Style aggregated network for facial landmark detection. In *IEEE conference on computer vision and pattern recognition (CVPR)*.
- Dong, X., Yu, S. I., Weng, X., Wei, S.E., Yang, Y., & Sheikh, Y. (2018b). Supervision-by-Registration: An unsupervised approach to improve the precision of facial landmark detectors. In *IEEE conference on computer vision and pattern recognition (CVPR)*.
- Dong, Y., & Wu, Y. (2015). Adaptive cascade deep convolutional neural networks for face alignment. *Computer Standards and Interfaces*, 42, 105–112.
- Fabian Benitez-Quiroz, C., Srinivasan, R., & Martinez, A. M. (2016). Emotionet: An accurate, real-time algorithm for the automatic annotation of a million facial expressions in the wild. In *IEEE conference on computer vision and pattern recognition (CVPR)*.
- Fan, H., & Zhou, E. (2016). Approaching human level facial landmark localization by deep learning. *Image and Vision Computing*, 47, 27–35.
- Feng, Z. H., Hu, G., Kittler, J., Christmas, W., & Wu, X. J. (2015a). Cascaded collaborative regression for robust facial landmark detection trained using a mixture of synthetic and real images with dynamic weighting. *IEEE Transactions on Image Processing*, 24(11), 3425–3440.
- Feng, Z. H., Huber, P., Kittler, J., Christmas, W., & Wu, X. (2015b). Random Cascaded-Regression Cope for Robust Facial Landmark Detection. *IEEE Signal Processing Letters*, 22(1), 76–80.
- Feng, Z. H., Kittler, J., Awais, M., Huber, P., & Wu, X. J. (2017a). Face detection, bounding box aggregation and pose estimation for robust facial landmark localisation in the wild. In *IEEE conference on computer vision and pattern recognition workshops (CVPRW)* (pp. 160–169).
- Feng, Z. H., Kittler, J., Christmas, W., Huber, P., & Wu, X. J. (2017b). Dynamic attention-controlled cascaded shape regression exploiting training data augmentation and fuzzy-set sample weighting. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2481–2490).
- Feng, Z. H., Huber, P., Kittler, J., Hancock, P., Wu, X. J., Zhao, Q., Koppen, P., & Rätzsch, M. (2018a). Evaluation of Dense 3D Reconstruction from 2D Face Images in the Wild. In *IEEE conference on automatic face and gesture recognition (FG)* (pp. 780–786).
- Feng, Z. H., Kittler, J., Awais, M., Huber, P., & Wu, X. J. (2018b). Wing loss for robust facial landmark localisation with convolutional neural networks. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2235–2245).
- Feng, Z. H., Kittler, J., & Wu, X. J. (2019). Mining Hard Augmented Samples for Robust Facial Landmark Localisation with CNNs. *IEEE Signal Processing Letters*, 26(3), 450–454.
- Gao, X., Su, Y., Li, X., & Tao, D. (2010). A review of active appearance models. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(2), 145–158.
- Ghiasi, G., & Fowlkes, C. C. (2014). Occlusion coherence: Localizing occluded faces with a hierarchical deformable part model. In *IEEE conference on computer vision and pattern recognition*.
- Girshick, R. (2015). Fast R-CNN. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1440–1448).

- Hara, K., & Chellappa, R. (2014). Growing regression forests by classification: Applications to object pose estimation. In *European conference on computer vision, (ECCV)* (pp. 552–567). Springer.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 770–778).
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint [arXiv:1704.04861](https://arxiv.org/abs/1704.04861)
- Huang, Z., Zhou, E., & Cao, Z. (2015). Coarse-to-fine face alignment with multi-scale local patch regression. arXiv preprint [arXiv:1511.04901](https://arxiv.org/abs/1511.04901)
- Huber, P. J., et al. (1964). Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1), 73–101.
- Jourabloo, A., & Liu, X. (2016). Large-pose face alignment via CNN-based dense 3D model fitting. In *IEEE conference on computer vision and pattern recognition (CVPR)*.
- Jourabloo, A., & Liu, X. (2017). Pose-invariant face alignment via CNN-based dense 3D model fitting. *International Journal of Computer Vision (IJCV)*, 124(2), 187–203.
- Jourabloo, A., Ye, M., Liu, X., & Ren, L. (2017). Pose-invariant face alignment with a single cnn. In *IEEE international conference on computer vision (ICCV)*.
- Kazemi, V., & Sullivan, J. (2014). One millisecond face alignment with an ensemble of regression trees. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1867–1874).
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)
- Kittler, J., Huber, P., Feng, Z., Hu, G., & Christmas, W. (2016). 3D morphable face models and their applications. In *International conference on articulated motion and deformable objects* (pp. 185–206). Springer.
- Koestinger, M., Wohlhart, P., Roth, P. M., & Bischof, H. (2011). Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *First IEEE international workshop on benchmarking Facial image analysis technologies*.
- Koppen, P., Feng, Z. H., Kittler, J., Awais, M., Christmas, W., Wu, X. J., et al. (2018). Gaussian Mixture 3D Morphable Face Model. *Pattern Recognition*, 74, 617–628.
- Le, V., Brandt, J., Lin, Z., Bourdev, L., & Huang, T. S. (2012). Interactive facial feature localization. In *European conference on computer vision (ECCV)* (pp. 679–692). Springer.
- Li, S., Deng, W., & Du, J. (2017). Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild. In *IEEE conference on computer vision and pattern recognition (CVPR)*.
- Liang, Z., Ding, S., & Lin, L. (2015). Unconstrained facial landmark localization with backbone-branches fully-convolutional networks. arXiv preprint [arXiv:1507.03409](https://arxiv.org/abs/1507.03409)
- Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., & Song, L. (2017a). SpheroFace: Deep hypersphere embedding for face recognition. In *IEEE conference on computer vision and pattern recognition (CVPR)*.
- Liu, Y., Jourabloo, A., Ren, W., & Liu, X. (2017b). Dense face alignment. In *IEEE international conference on computer vision workshops (ICCVW)*.
- Liu, Z., Luo, P., Wang, X., & Tang, X. (2015). Deep learning face attributes in the wild. In *IEEE international conference on computer vision (ICCV)* (pp. 3730–3738).
- Liu, Z., Zhu, X., Hu, G., Guo, H., Tang, M., Lei, Z., Robertson, N. M., & Wang, J. (2019). Semantic alignment: Finding semantically consistent ground-truth for facial landmark detection. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 3467–3476).
- Luo, P., Wang, X., & Tang, X. (2012). Hierarchical face parsing via deep learning. In *2012 IEEE conference on computer vision and pattern recognition (CVPR)*, IEEE (pp. 2480–2487).
- Lv, J., Shao, X., Xing, J., Cheng, C., & Zhou, X. (2017). A deep regression architecture with two-stage re-initialization for high performance facial landmark detection. In *IEEE conference on computer vision and pattern recognition (CVPR)*.
- Martinez, B., & Valstar, M. F. (2016). L2, 1-based regression and prediction accumulation across views for robust facial landmark detection. *Image and Vision Computing*, 47, 36–44.
- Masi, I., Rawls, S., Medioni, G., & Natarajan, P. (2016a). Pose-aware face recognition in the wild. In *IEEE conference on computer vision and pattern recognition (CVPR)*.
- Masi, I., Tran, A. T., Hassner, T., Leksut, J. T., & Medioni, G. (2016b). Do we really need to collect millions of faces for effective face recognition? In *European conference on computer vision (ECCV)* (pp. 579–596).
- Messer, K., Matas, J., Kittler, J., Luettin, J., & Maitre, G. (1999). XM2VTSDB: The extended M2VTS database. *International Conference on Audio and Video-based Biometric Person Authentication, Citeseer*, 964, 965–966.
- Newell, A., Yang, K., & Deng, J. (2016). Stacked hourglass networks for human pose estimation. In *European conference on computer vision (ECCV)* (pp. 483–499).
- Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015). Deep face recognition. In *British machine vision conference*.
- Phillips, P. J., Flynn, P. J., Scruggs, T., Bowyer, K. W., Chang, J., Hoffman, K., Marques, J., Min, J., & Worek, W. (2005). Overview of the face recognition grand challenge. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, IEEE (Vol. 1, pp. 947–954).
- Ranjian, R., Patel, V. M., & Chellappa, R. (2017). Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. In *IEEE transactions on pattern analysis and machine, intelligence*.
- Rashid, M., Gu, X., & Jae Lee, Y. (2017). Interspecies knowledge transfer for facial keypoint detection. In *IEEE conference on computer vision and pattern recognition (CVPR)*.
- Ren, S., Cao, X., Wei, Y., & Sun, J. (2016). Face alignment via regressing local binary features. *IEEE Transactions on Image Processing*, 25(3), 1233–1245.
- Romdhani, S., Gong, S., Psarrou, A., et al. (1999). A multi-view nonlinear active shape model using kernel PCA. *British Machine Vision Conference (BMVC)*, 99, 483–492.
- Roth, J., Tong, Y., & Liu, X. (2016). Adaptive 3D face reconstruction from unconstrained photo collections. In *IEEE conference on computer vision and pattern recognition (CVPR)*.
- Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., & Pantic, M. (2013). A semi-automatic methodology for facial landmark annotation. In *IEEE conference on computer vision and pattern recognition workshops (CVPRW)* (pp. 896–903).
- Sagonas, C., Antonakos, E., Tzimiropoulos, G., Zafeiriou, S., & Pantic, M. (2016). 300 faces in-the-wild challenge: Database and results. *Image and Vision Computing*, 47, 3–18.
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4510–4520).
- Shrivastava, A., Gupta, A., & Girshick, R. (2016). Training region-based object detectors with online hard example mining. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 761–769).
- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. CoRR. [abs/1409.1556](https://arxiv.org/abs/1409.1556).

- Sun, X., Wei, Y., Liang, S., Tang, X., & Sun, J. (2015). Cascaded hand pose regression. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 824–832).
- Sun, Y., Wang, X., & Tang, X. (2013). Deep convolutional network cascade for facial point detection. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 3476–3483).
- Taigman, Y., Yang, M., Ranzato, M., & Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1701–1708).
- Trigeorgis, G., Snape, P., Nicolaou, M. A., Antonakos, E., & Zafeiriou, S. (2016). Mnemonic descent method: A recurrent process applied for end-to-end face alignment. In *IEEE conference on computer vision and pattern recognition (CVPR)*.
- Uříčák, M., Franc, V., Thomas, D., Sugimoto, A., & Hlaváč, V. (2016). Multi-view facial landmark detector learned by the structured output svm. *Image and Vision Computing*, 47, 45–59.
- Walecki, R., Rudovic, O., Pavlovic, V., & Pantic, M. (2016). Copula ordinal regression for joint estimation of facial action unit intensity. In *IEEE conference on computer vision and pattern recognition (CVPR)*.
- Wang, N., Gao, X., Tao, D., Yang, H., & Li, X. (2018). Facial feature point detection: A comprehensive survey. *Neurocomputing*, 275, 50–65.
- Weng, R., Lu, J., Tan, Y. P., & Zhou, J. (2016). Learning cascaded deep auto-encoder networks for face alignment. *IEEE Transactions on Multimedia*, 18(10), 2066–2078.
- Wu, W., & Yang, S. (2017). Leveraging intra and inter-dataset variations for robust face alignment. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 150–159).
- Wu, W., Qian, C., Yang, S., Wang, Q., Cai, Y., & Zhou, Q. (2018a). Look at boundary: A boundary-aware face alignment algorithm. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2129–2138).
- Wu, Y., & Ji, Q. (2019). Facial landmark detection: A literature survey. *International Journal of Computer Vision*, 127(2), 115–142.
- Wu, Y., Gou, C., & Ji, Q. (2017a). Simultaneous facial landmark detection, pose and deformation estimation under facial occlusion. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3471–3480).
- Wu, Y., Hassner, T., Kim, K., Medioni, G., & Natarajan, P. (2017b). Facial landmark detection with tweaked convolutional neural networks. In *IEEE transactions on pattern analysis and machine intelligence*.
- Wu, Y., Shah, S. K., & Kakadiaris, I. A. (2018b). GoDP: Globally Optimized Dual Pathway deep network architecture for facial landmark localization in-the-wild. *Image and Vision Computing*, 73, 1–16.
- Xiao, S., Feng, J., Xing, J., Lai, H., Yan, S., & Kassim, A. (2016). Robust facial landmark detection via recurrent attentive-refinement networks. In *European conference on computer vision (ECCV)* (pp. 57–72).
- Xiao, S., Feng, J., Liu, L., Nie, X., Wang, W., Yan, S., & Kassim, A. (2017). Recurrent 3D–2D dual learning for large-pose facial landmark detection. In *The IEEE international conference on computer vision (ICCV)*.
- Xing, J., Niu, Z., Huang, J., Hu, W., Zhou, X., & Yan, S. (2018). Towards robust and accurate multi-view and partially-occluded face alignment. *IEEE transactions on pattern analysis and machine intelligence*, 40(4), 987–1001.
- Xiong, X., & De la Torre, F. (2013). Supervised descent method and its applications to face alignment. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 532–539).
- Xiong, X., & De la Torre, F. (2015). Global supervised descent method. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2664–2673).
- Xu, X., & Kakadiaris, I. A. (2017). Joint head pose estimation and face alignment framework using global and local cnn features. In *IEEE international conference on automatic face gesture recognition (FG)* (pp. 642–649).
- Yang, J., Liu, Q., & Zhang, K. (2017a). Stacked hourglass network for robust facial landmark localisation. In *IEEE conference on computer vision and pattern recognition workshops (CVPRW)* (pp. 2025–2033).
- Yang, J., Ren, P., Zhang, D., Chen, D., Wen, F., Li, H., & Hua, G. (2017b). Neural aggregation network for video face recognition. In *IEEE conference on computer vision and pattern recognition (CVPR)*.
- Yu, X., Zhou, F., & Chandraker, M. (2016). Deep deformation network for object landmark localization. In *European conference on computer vision (ECCV)* (pp. 52–70).
- Zeiler, M. D. (2012). Adadelta: An adaptive learning rate method. arXiv preprint [arXiv:1212.5701](https://arxiv.org/abs/1212.5701)
- Zeng, J., Liu, S., Li, X., Mahdi, D. A., Wu, F., & Wang, G. (2018). Deep context-sensitive facial landmark detection with tree-structured modeling. *IEEE Transactions on Image Processing*, 27(5), 2096–2107.
- Zeng, Z., Pantic, M., Roisman, G. I., & Huang, T. S. (2009). A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1), 39–58.
- Zhang, J., Shan, S., Kan, M., & Chen, X. (2014). Coarse-to-Fine Auto-Encoder Networks (CFAN) for Real-Time Face Alignment. *European Conference on Computer Vision (ECCV)*, 8690, 1–16.
- Zhang, J., Kan, M., Shan, S., & Chen, X. (2016a). Occlusion-free face alignment: Deep regression networks coupled with de-corrupt autoencoders. In *IEEE conference on computer vision and pattern recognition (CVPR)*.
- Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016b). Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10), 1499–1503.
- Zhu, M., Shi, D., Zheng, M., & Sadiq, M. (2019). Robust facial landmark detection via occlusion-adaptive deep networks. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 3486–3496).
- Zhu, S., Li, C., Change Loy, C., & Tang, X. (2015). Face alignment by coarse-to-fine shape searching. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 4998–5006).
- Zhu, S., Li, C., Loy, C. C., & Tang, X. (2016a). Unconstrained face alignment via cascaded compositional learning. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 3409–3417).
- Zhu, X., & Ramanan, D. (2012). Face detection, pose estimation, and landmark localization in the wild. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2879–2886).
- Zhu, X., Lei, Z., Liu, X., Shi, H., & Li, S. Z. (2016b). Face alignment across large poses: A 3D solution. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 146–155).