CNN based facial aesthetics analysis through dynamic robust losses and ensemble regression

Fares Bougourzi, Fadi Dornaika, Nagore Barrena, Cosimo Distante, Abdelmalik Taleb-Ahmed

Applied Intelligence
https://doi.org/10.1007/s10489-022-03943-0

CNN based facial aesthetics analysis through dynamic robust losses and ensemble regression

Fares Bougourzi[1] · Fadi Dornaika[2,3,4] · Nagore Barrena[3] · Cosimo Distante[1] · Abdelmalik Taleb-Ahmed[5]

Accepted: 7 June 2022

Abstract
In recent years, estimating beauty of faces has attracted growing interest in the fields of computer vision and machine learning. This is due to the emergence of face beauty datasets (such as SCUT-FBP, SCUT-FBP5500 and KDEF-PT) and the prevalence of deep learning methods in many tasks. The goal of this work is to leverage the advances in Deep Learning architectures to provide stable and accurate face beauty estimation from static face images. To this end, our proposed approach has three main contributions. To deal with the complicated high-level features associated with the FBP problem by using more than one pre-trained Convolutional Neural Network (CNN) model, we propose an architecture with two backbones (2B-IncRex). In addition to 2B-IncRex, we introduce a parabolic dynamic law to control the behavior of the robust loss parameters during training. These robust losses are ParamSmoothL1, Huber, and Tukey. As a third contribution, we propose an ensemble regression based on five regressors, namely Resnext-50, Inception-v3 and three regressors based on our proposed 2B-IncRex architecture. These models are trained with the following dynamic loss functions: Dynamic ParamSmoothL1, Dynamic Tukey, Dynamic ParamSmoothL1, Dynamic Huber, and Dynamic Tukey, respectively. To evaluate the performance of our approach, we used two datasets: SCUT-FBP5500 and KDEF-PT. The dataset SCUT-FBP5500 contains two evaluation scenarios provided by the database developers: 60-40% split and five-fold cross-validation. Our approach outperforms state-of-the-art methods on several metrics in both evaluation scenarios of SCUT-FBP5500. Moreover, experiments on the KDEF-PT dataset demonstrate the efficiency of our approach for estimating facial beauty using transfer learning, despite the presence of facial expressions and limited data. These comparisons highlight the effectiveness of the proposed solutions for FBP. They also show that the proposed Dynamic robust losses lead to more flexible and accurate estimators.

Keywords Facial beauty prediction · Convolutional neural network · Deep learning · Ensemble regression · Robust loss functions

1 Introduction

The search for beauty has been pursued by mankind since its beginnings. Attempting to discover the secret of beauty has been a goal of philosophers, artists, and scientists throughout human history [1]. Even in ancient Greece, for example, beauty was associated with symmetry. Nowadays, the beauty of the face receives even more interest due to the rapid development of plastic surgery and the cosmetics industry [2].

Computer science is also not unaware of this fact, therefore facial beauty has become an interesting research topic in computer vision and machine learning [3, 4]. This research mainly focuses on facial beauty estimation and classification/prediction which can be useful in various applications such as cosmetic recommendations [5], plastic surgery planning [6], facial beautification [7], and social network services (SNS) (such as Facebook, Instagram, and dating websites) [8]. In addition, automatic facial beauty prediction (FBP) may find application when attractiveness is a basic requirement, such as in advertising, magazine covers, and in the selection of applicants for certain professions, such as in the entertainment industry and modeling business [6].

Motivation Over the past decade, CNNs have become the dominant solution for most computer vision and machine learning tasks [9, 10]. Despite these tremendous advances

✉ Fadi Dornaika
fadi.dornaika@ehu.eus

Extended author information available on the last page of the article.

Published online: 26 August 2022

in deep learning methods, FBP has not been able to benefit much from deep learning. One of the goals of this work is to leverage some of the recent powerful CNN architectures to develop an accurate and robust solution for FBP.

Developing such an accurate solution for facial beauty estimation is a difficult task because facial beauty is a subjective task that changes from person to person, and facial attributes (gender, ethnicity, age...) also affect facial beauty evaluation. In addition, the person's internal state (facial expression) can also influence facial beauty evaluation [11]. Although Deep Learning, especially CNN architectures, have made significant progress in facial beauty assessment and prediction, it is noted that more labelled data is needed to train Deep CNNs. To deal with the aforementioned data limitation, we use active data augmentation. Moreover, pre-trained models on ImageNet database [12] are used to extract high level features.

In this paper, we present a Deep Learning approach for predicting the beauty of faces. The presented approach is based on three main contributions. First, we propose a two-backbone architecture where two different CNN architectures are fused into a single architecture that is trained in an end-to-end method. Second, we propose a dynamic robust loss function for training the deep regressors. Third, we propose an ensemble of regressions where the final prediction is given by the average of all predictions without retraining the final solution with a new validation set. In the ensemble solution, each model is trained separately. The ensemble consists of single-branch architectures (ResneXt-50 and Inception-v3) and our proposed two-backbones architecture (2B-IncRex) with different loss functions. In this approach, three robust loss functions are made dynamic: ParametricSmoothL1, Huber and Tukey.

In the following, the most important contributions of the proposed solution are explained one after the other.

- ParamSmoothL1 regression loss function and a dynamic law that changes the parameters of the robust loss function during training. For this purpose, we use the parabolic law with the following robust loss functions: ParamSmoothL1, Huber and Tukey, to be able to solve the problem of complexity in finding the best loss function parameter. Moreover, these dynamic losses improve the training convergence compared to the standard loss functions (MSE and L_1) and the robust loss functions (SmoothL1, Huber and Tukey) that assume a fixed parameter.
- A network with two backbones (2B-IncRex) based on ResneXt-50 and Inception-v3 architectures is proposed for face beauty prediction.
- regression for face beauty estimation by fusing the predicted values of one-branch networks (ResneXt-50 and Inception-v3) and two-backbones networks (2B-IncRex) is proposed. This ensemble of five CNNs is trained with these dynamic loss functions: Dynamic ParamSmoothL1, Dynamic Tukey, Dynamic ParamSmoothL1, Dynamic Huber, and Dynamic Tukey, respectively. Although the individual regression models are trained separately with the same fixed hyperparameters, the estimates produced by the resulting ensemble regression are more accurate compared to the individual models as well as to the state-of-the-art solutions. The code to train and test our approach is publicly available at. https://github.com/faresbougourzi/Dynamic_ER-CNN. (Last accessed on March, 25^{th} 2022)

The paper is divided into the following sections: Section 2 presents some related work on facial beauty prediction. In Section 3, we explain the backbone CNN architectures used, the proposed approach, and the proposed dynamic robust losses. Section 4 contains: the description of the databases and evaluation metrics used, and the experimental setup. Section 5 presents the performance evaluations for the SCUT-FBP5500 dataset. Section 6 presents the performance evaluation for the KDEF-PT dataset. Section 7 provides a discussion and comparison with state-of-the-art methods. Finally, Section 8 concludes the paper.

2 Related work

Automatic prediction of the beauty of faces is still a young problem, but it is becoming increasingly important in the field of machine learning and computer vision. There is a unified concept of facial beauty that enables the automation of this prediction [13, 14]. In this way, the classification of facial beauty and the prediction of attractiveness score were developed to allow the association of facial attractiveness and image features in a quantitative mode [15]. The first database created to treat FBP as a regression task dates back to 2015 [16]. In fact, two main methods for performing FBP can be distinguished in the literature: hand-crafted [17–24] and deep learning [18, 25–27]. Similarly, hand-crafted methods are classified as geometry-based or appearance-based [22].

Before the heyday of deep learning architectures, hand-crafted methods were commonly used for FBP. Aarabi et al. [21] and H. Yan [22] presented work dealing with appearance-based hand-crafted methods. In the first work, an automatic system for evaluating the beauty of faces was developed. It is based on the ratios between facial features (face, eyes, eyebrows and moth concretely) with the K-nearest neighbor algorithm to learn the beauty assignment. H. Yan [22], on the other hand, proposed a new CSOR

(Cost-Sensitive Ordinal Regression) method to measure the importance of samples in different classes. The CSOR is applied to four types of characteristics: Intensity, LBP [28], SIFT [29], and LE [30]. A typical geometry-based hand-crafted method was described by Zhang et al. [20] presented. This technique uses a huge amount of data (tens of thousands of face images, both female and male). These are mapped to a human face shape subspace, and a quantitative method is used to analyze the effects of facial geometry on the beauty of the human face. The analysis was performed using the transformation invariant shape distance measurement. On the other hand, Liang et al. proposed a mixed technique combining geometric-based and appearance-based hand-crafted methods. It is based on the use of geometric features (extracted 18-dimensional ratio features of faces) and appearance features (40 Gabor feature maps), with apartment predictors that are linear regression (LR) and Support Vector Regression (SVR).

Most of the hand-crafted methods listed above have been tested using the SCUT-FBP5500 database. This database includes 5500 frontal, neutral-looking, and unclouded faces of individuals aged 15 to 60 years [18]. On the other hand, it should be mentioned that the introduction of deep learning methods in computer vision and especially in FBP has surpassed the results obtained with hand-crafted methods. As a result, in recent years, deep learning architectures have been widely used for evaluating the beauty of faces.

In [18], Liang et al. presented their face beauty database (SCUT-FBP5500) with two evaluation protocols (60-40% split and five-fold cross validation). They tested three CNN architectures (Alexnet [31], Resnet-18 [32], and ResneXt-50 [33]). Their results show that the ResneXt-50 architecture outperforms the other two deep architectures. In terms of the improvement that Deep Learning methods represent over hand-crafted methods, it should also be noted that all of the deep neural networks tested in their work (including Alexnet and Resnet-18) performed better than the hand-crafted methods tested with various shallow regressors. Cao et al. used a residual-in-residual (RIR) block to build a deeper network with multilevel skip connections to achieve better gradient transmission flow. In addition, they used both channel-wise and space-wise attention mechanisms to find the inherent correlation between feature maps. Their approach was tested on the SCUT-FBP5500 database [18] and showed good performance. Lin et al. [27] proposed an R^3CNN architecture. It consists of two components: a regression component and a ranking component. The regression component has a Siamese network (two identical regression sub-networks) to consistently map each face image to a beauty value. The ranking component, on the other hand, uses the Siamese network for a few images and provides an additional task to improve the learning process of the regression subnets. The idea is that the ranking network learns the pairwise ranking of beauty for two images. Their architecture showed promising results on the SCUT-FBP [16] and SCUT-FBP5500 [18] databases. Dornaika et al. [34] introduced a multi-layer local discriminative embedding algorithm that integrates feature selection as the main step. Feature selection captures the most relevant and discriminative features of an input face image or face descriptor. All the methods mentioned so far are supervised learning methods. However, the work presented in [35] proves that semi-supervised learning also yields promising results in facial beauty estimation.

3 Methodology

This section focuses on presenting the CNN architectures used, our proposed approach, and the proposed dynamic robust losses.

3.1 Backbone CNN Architectures

The use of CNN architectures in FBP has become increasingly popular since Deep Learning methods have demonstrated their efficient performance [31].

This work is also based on CNN and the architecture presented is a combination of ResneXt-50 [33] and Inception-v3 [36]. However, the approach is open to using other backbone architectures. In addition, pre-trained models are used, trained with the ImageNet challenge database [12].

To keep the paper self-contained, this section briefly introduces the two CNNs (ResneXt-50 and Inception-v3) used as backbone architectures in our proposed solution.

ResneXt-50 Architecture: ResneXt-50 architecture [33] is a variation of the popular Resnet [37] architecture. The main idea is to modify the residue blocks and add parallel convolutional layers with a smaller number of filters. The outputs of these filters are combined by summation and serve as input for the next residual block.

Inception-v3 Architecture: Inception-v3 [36] is an evolution of the GoogLeNet architecture [38], in which the Inception module was introduced. The main idea of this module is to use parallel convolutional layers with different kernel sizes as well as pooling layers. In this way, different receptive fields can be applied to the input in an efficient way.

3.2 Our approach

Our method is described in Fig. 1. The predicted score of beauty is the mean of multiple scores, which means that we

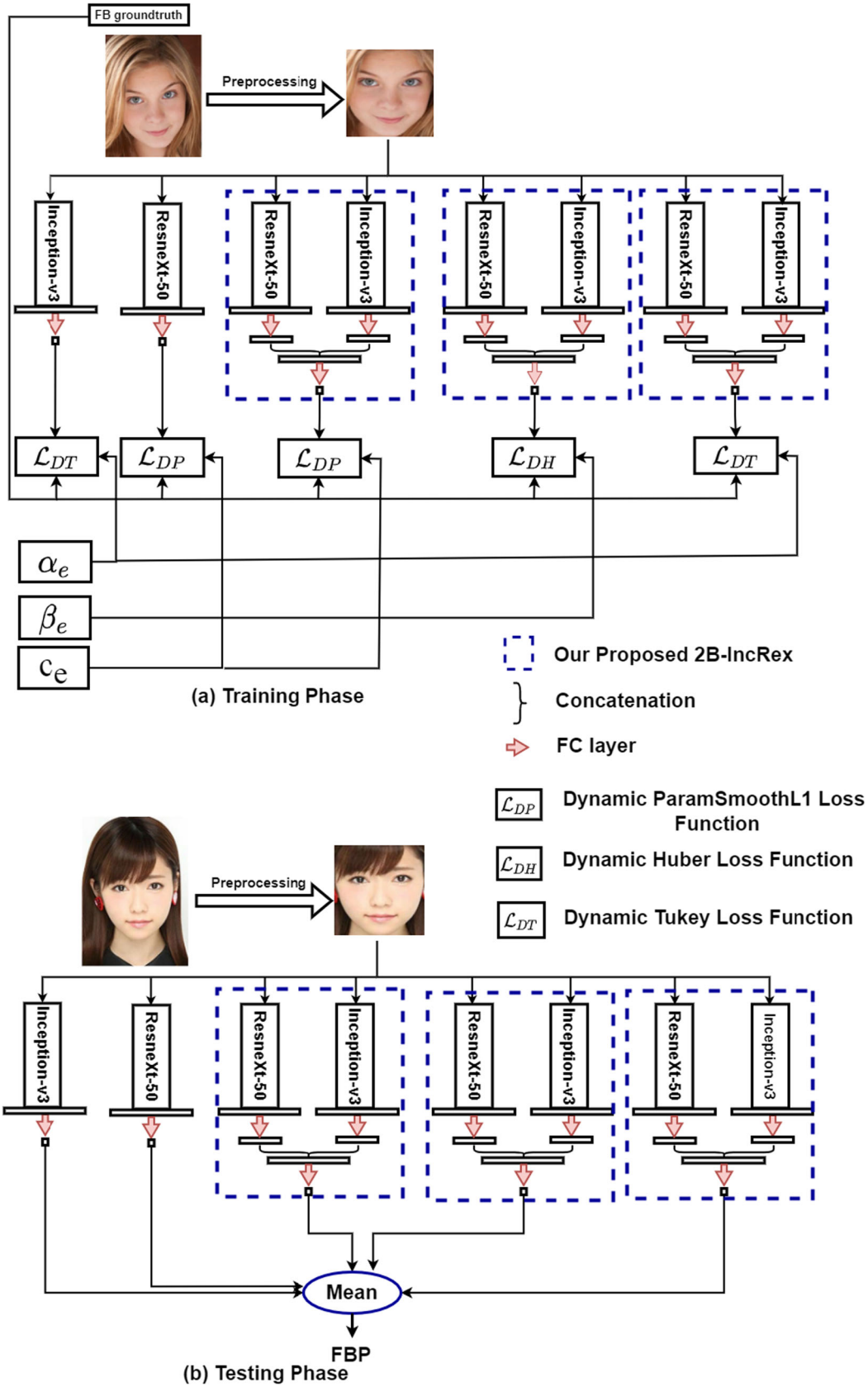

Fig. 1 General structure of the proposed approach (Dynamic ER-CNN). Note that every model in this set of five solutions is trained separately using a given regression loss

employ an ensemble of multiple regression models, each of which independently provides an individual score. In our implementation, we use five models. There are two main contributions to this ensemble: (i) the deep network with two backbones (2B-IncRex) (see Section 3.4) and (ii) the dynamic robust loss functions (see Section 3.5).

The first two scores are predicted by the trained Inception-v3 and ResneXt-50 deep networks using the Dynamic Tukey loss function and the Dynamic ParamSmoothL1 loss function, respectively. The selection of the associated loss function for each backbone was empirically determined when these backbones were evaluated in Section 5.1.2. The remaining three scores are estimated after training the proposed deep network with two backbones (2B-IncRex) using the three dynamic loss functions: Dynamic ParamSmoothL1, Dynamic Huber and Dynamic Tukey. The deep network with two backbones consists of ResneXt-50 and inception-v3, which are merged into a single architecture. As will be seen in the experimental section, the performance using the two contributions without the ensemble is better than that of the state-of-the-art methods. The use of the ensemble shown in Fig. 1 will further improve the results.

3.3 Face preprocessing

In the preprocessing phase of the faces, we adopted the 2D alignment scheme described in [39] and [40]. This scheme is summarized in Fig. 2. To obtain a rectified and cropped face region, we apply three steps to the raw face image. First, the face image is rotated so that the two eyes have the same vertical coordinates. For the SCUT-FBP5500 dataset [18], we used the face landmarks provided by the authors of this dataset. For the KDEF-PT dataset [11], we used the Dlib library [41] to obtain these landmarks. Once the image and its associated detected points are rotated in the image plane, the three furthest face points in the left, right, and bottom directions are selected as the three boundaries of the face. We denote the distance from the lower boundary to the vertical position of the eyes as d_1. The upper boundary of the face is set at a distance d_2 from the eyes, which is set to $d_2 = 0.6\ d_1$. It is worth noting that the distance d_2 determines the region of the forehead included in the cropped face. Empirically, we found that $d_2 = 0.6\ d_1$ works well. Finally, the face ROI is obtained by cropping the face using the four specified boundaries. The obtained ROI is then resized to a fixed size that depends on the input size of the corresponding convolutional neural network.

3.4 Two branches architecture

Recently, many successful deep architectures have been proposed for many computer vision tasks. In our solution, we employ two dual architectures to exploit the different capabilities of deep neural networks. Since FBP image data is limited, we propose to exploit the low-level and high-level feature extraction capability of two powerful architectures jointly. Figure 3 summarizes our introduced architecture with two branches. The first and second branches are the ResneXt-50 and Inception-v3 architectures, respectively, with the decision layers removed. In our proposed architecture with two backbones, we added the FC1 layer, which maps the encoded deep features of the ResneXt-50 branch (vector of dimension 2048) to 1024 neurons. Similarly, we added layer FC2, which maps the embedded deep features of the Inception-v3 branch (vector of dimension 2048) to 1024 neurons. FC1 and FC2 were concatenated into a single vector FC, which is followed by the FC3 layer that performs the regression, namely the beauty score. Note that the weights of both branches are the weights of the pre-trained ResneXt-50 and Inception-v3 models (trained on the ImageNet Challenge database [12].),

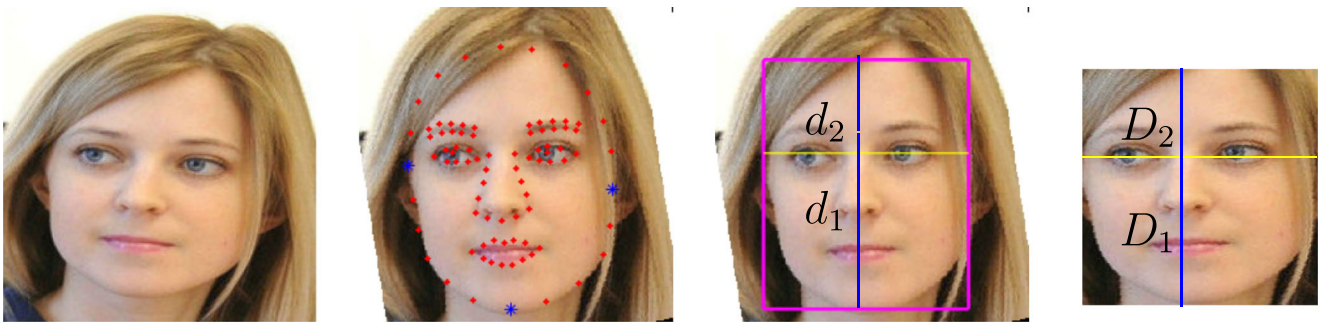

Fig. 2 Face Region of Interest. The left image is an original image from the database SCUT-FBP5500 [18]. The second image is the rotated face with its 86 detected landmarks used to estimate the three face boundary lines (right, left, and bottom). These boundaries correspond to the three points $*$ marked in blue. The third image shows how the upper boundary of the face is determined. It is located at a distance $d_2 = 0.6d_1$ from the vertical position of the two eyes. The fourth image shows the cropped and rescaled face image with 224 $\times$ 224 pixels. Note that the distances D_1 and D_2 are constant for all cropped faces

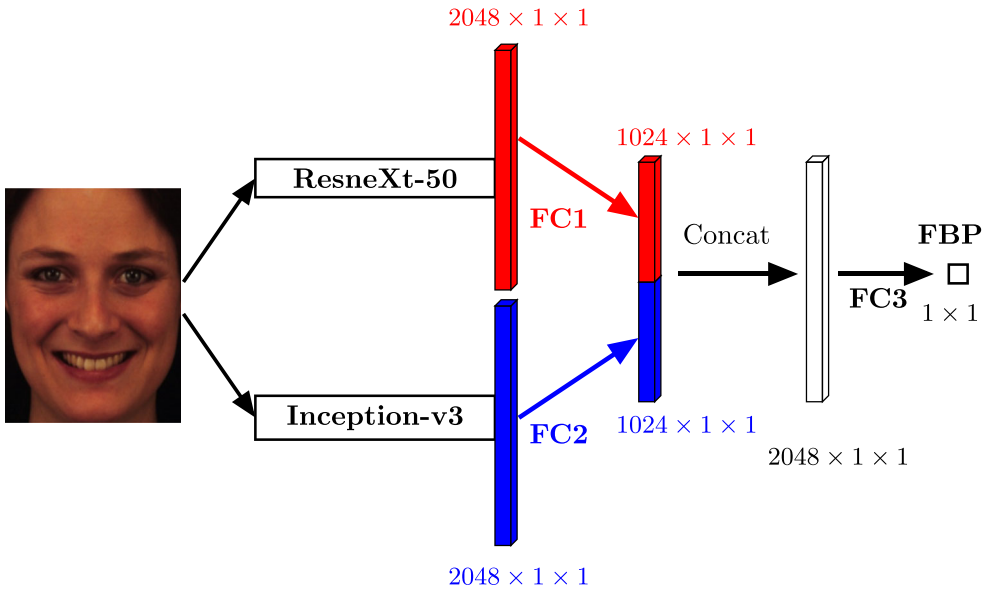

Fig. 3 Our proposed two branches network 2B-IncRex

while the FC1, FC2 and FC3 layers are randomly initialized. Our proposed network with two branches is called 2B-IncRex architecture. In the training phase, we will fine-tune this architecture for FBP.

3.5 Loss Functions: the use of dynamic robust losses

During convolutional network training, the loss function measures the error (the loss) between the ground truth and the current predicted values. Training a CNN aims to minimize the loss based on the gradients of the loss function used to update the weights of the network. In this section, we will describe the loss functions we used for training our proposed architectures. We will also introduce a dynamic law that adjusts the parameters of these robust losses during training. The losses are computed for a batch of N face images. Let y_i denote the ground truth score of the i^{th} image, and $\hat{y}_i$ denote the predicted score.

3.5.1 Dynamic Parameterized SmoothL1 (ParamSmoothL1) loss function

The loss function SmoothL1 produces a criterion that uses a quadratic term when the absolute element-wise error falls below 1, and the absolute error otherwise. It is commonly used for training deep CNN-based regressions because it is less sensitive to the presence of outliers than the Mean Square Error loss function and in some cases prevents exploding gradients [42]. The SmoothL1 loss function of N images is defined by:

$$L_{SmoothL1} = \frac{1}{N}\sum_{i=1}^{N} z_i \tag{1}$$

where N is the batch size and z_i is given by:

$$z_i = \begin{cases} 0.5\,(y_i - \hat{y}_i)^2, & \text{if } |y_i - \hat{y}_i| < 1 \\ |y_i - \hat{y}_i| - 0.5, & \text{otherwise} \end{cases} \tag{2}$$

In this work, we introduce the Dynamic Parametrized SmoothL1 loss function. First, we present the Parametrized SmoothL1 loss. We then present its dynamic variant.

Since the threshold can be different from one task to another, we proposed a Parameterized SmoothL1 loss function which is defined by:

$$L_{ParamSmoothL1} = \frac{1}{N}\sum_{i=1}^{N} z_i \tag{3}$$

where N is the batch size and z_i is given by:

$$z_i = \begin{cases} 0.5\,(y_i - \hat{y}_i)^2, & \text{if } |y_i - \hat{y}_i| \leq \alpha \\ |y_i - \hat{y}_i| + 0.5\,\alpha^2 - \alpha, & \text{otherwise} \end{cases} \tag{4}$$

where α is a tunable parameter. Our proposed dynamic robust loss functions are motivated by the following observation. During the training of CNNs, the robust loss functions can be adjusted as the training progresses. Namely, during training, the model evolves and the trained outlier examples may vary. In the early stages of training, the model is usually neither very stable nor accurate enough to handle the outlier examples correctly. Therefore, it is advantageous to use a quadratic loss function. At the end of the training, the model may be more or less accurate to handle the outliers. Therefore, it is useful to use a more rigorous robust loss function where the range of non-outlier errors is relatively small. In other words, we can time the parameter of the robust loss function (ParamSmoothL1) so that it is initialized with a maximum value and decreases monotonically as training progresses. From a practical point of view, it is extremely difficult to know the best value for

α in advance. However, the variation interval $[\alpha_{min}, \alpha_{max}]$ can be known in advance. Therefore, to better fit the robust loss function to the training progress, we propose a dynamic parameter α that decreases according to a parabolic law as a function of the epoch number. The current value of α is given by:

$$\alpha_e = \alpha_{max} - (\alpha_{max} - \alpha_{min}) \left(\frac{e}{n_e}\right)^2 \tag{5}$$

where α_e is the value of α in the current epoch (e) varying between 1 and the total number of epochs (n_e). α_{max} and α_{min} are the maximum and minimum of the α value. In this paper, we denote the proposed Dynamic Parameterized SmoothL1 by Dynamic ParamSmoothL1. Figure 4 illustrates the variation of α using the proposed dynamic law ((5)) as a function of epoch number. Here α_{max} and α_{min} are fixed at 0.7 and 0.3, respectively. Our introduced dynamic law was inspired by dynamic laws used to control the learning rate during training in stochastic gradient descent methods [43].

3.5.2 Dynamic Huber loss function

Huber is another robust loss function that is less sensitive to outliers in the data than the Mean Square Error loss function. For N training images, the Huber loss function is defined by [44]:

$$L_{Huber} = \frac{1}{N} \sum_{i=1}^{N} z_i \tag{6}$$

where N is the batch size and z_i is defined by:

$$z_i = \begin{cases} 0.5\,(y_i - \hat{y}_i)^2, & \text{if } |y_i - \hat{y}_i| \le \beta \\ \beta\,|y_i - \hat{y}_i| - 0.5\,\beta^2, & \text{otherwise} \end{cases} \tag{7}$$

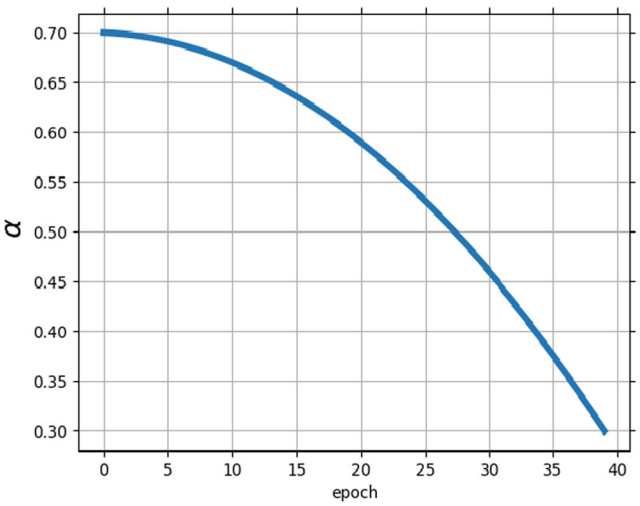

Fig. 4 Dynamic parameter α that decreases from 0.7 to 0.3

where β is a controlled parameter. Figure 5, shows a visualization of Huber loss function with four β values (0.7, 0.5, 0.3 and 0.1) and L_2 loss function.

Similar to ParamSmoothL1 loss, we suggest using dynamic β during training according to the equation:

$$\beta_e = \beta_{max} - (\beta_{max} - \beta_{min}) \left(\frac{e}{n_e}\right)^2 \tag{8}$$

where β_e is the value of β in the current epoch (e), where e increases from 1 to the total number of epochs (n_e). β_{max} and β_{min} are the defined maximum and minimum of β value.

3.5.3 Dynamic Tukey loss function

The Tukey loss function [45] suppresses the influence of outlier data during backpropagation by reducing the magnitude of its gradient toward zero. Another interesting property of this loss function is its smooth transition between inliers and outliers [46]. The Tukey loss function is defined by:

$$L_{Tukey} = \frac{1}{N} \sum_{i=1}^{N} z_i \tag{9}$$

where N is the batch size and z_i is given by:

$$z_i = \begin{cases} \frac{c^2}{6} \left[1 - \left(1 - \left(\frac{|y_i - \hat{y}_i|}{c}\right)^2\right)^3\right], & \text{if } |y_i - \hat{y}_i| \le c \\ \frac{c^2}{6}, & \text{otherwise} \end{cases} \tag{10}$$

where c is an adjustable parameter. Similar to ParamSmoothL1 and Huber losses, we propose to use dynamic c during training through the equation:

$$c_e = c_{max} - (c_{max} - c_{min}) \left(\frac{e}{n_e}\right)^2 \tag{11}$$

where c_e is the value of c in the current epoch (e), with e increasing from 1 to the total number of epochs (n_e). c_{max} and c_{min} are the maximum and minimum of c value.

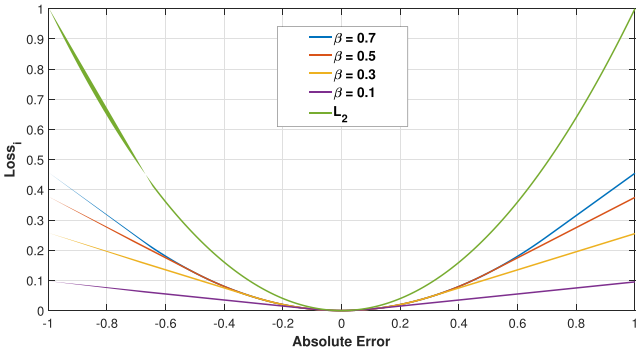

Fig. 5 Illustration of two loss functions: the Mean Sqaure Error loss (L_2 loss), and the Huber loss function with four β values (0.7, 0.5, 0.3 and 0.1)

4 Experimental setting

4.1 Database and evaluation protocols

To evaluate the performance of our approach, we used the SCUT-FBP5500 [18] database. It consists of 5500 frontal faces of subjects with different attributes: age (from 15 to 60), gender (male/ female), and ethnicity (Asian/ Caucasian). Each facial image was labelled with beauty score in the range [1-5] by 60 volunteers. In addition, each facial image has 86 facial landmarks. Figures 6 and 7 show some facial samples with their corresponding face beauty score. The creators of the SCUT-FBP5500 database provided two evaluation scenarios [18]. In the first scenario, the data were divided into a training split and test split (60 - 40%). In the second scenario, the data were divided into 5 folds to perform a five-fold cross-validation. In our evaluations, we will use both scenarios.

In addition to the SCUT-FBP5500 dataset, the KDEF-PT dataset [11] was used to evaluate the performance of our approach in the presence of facial expressions. KDEF-PT consists of 70 subjects (35 females and 35 males). Each subject performs three facial expressions, namely joy, neutrality, and anger. To determine facial attractiveness, each image was labelled by the participants and they were asked to indicate the extent of attractiveness on a 7-point rating scale (1 = not at all attractive to 7 = very attractive). Each image was rated by a varying number of subjects (from 34 to 42 subjects). The attractiveness score is the average of the subjects' ratings, Fig. 7 shows two examples from the KDEF-PT dataset. In our experiments, we split the 70 subjects into a training set and a validation set (80% and 20%) to avoid using the same subject in both the training and validation sets. Since there are only 168 training images, we used the trained models from the first fold of the SCUT-FBP5500 dataset and then performed

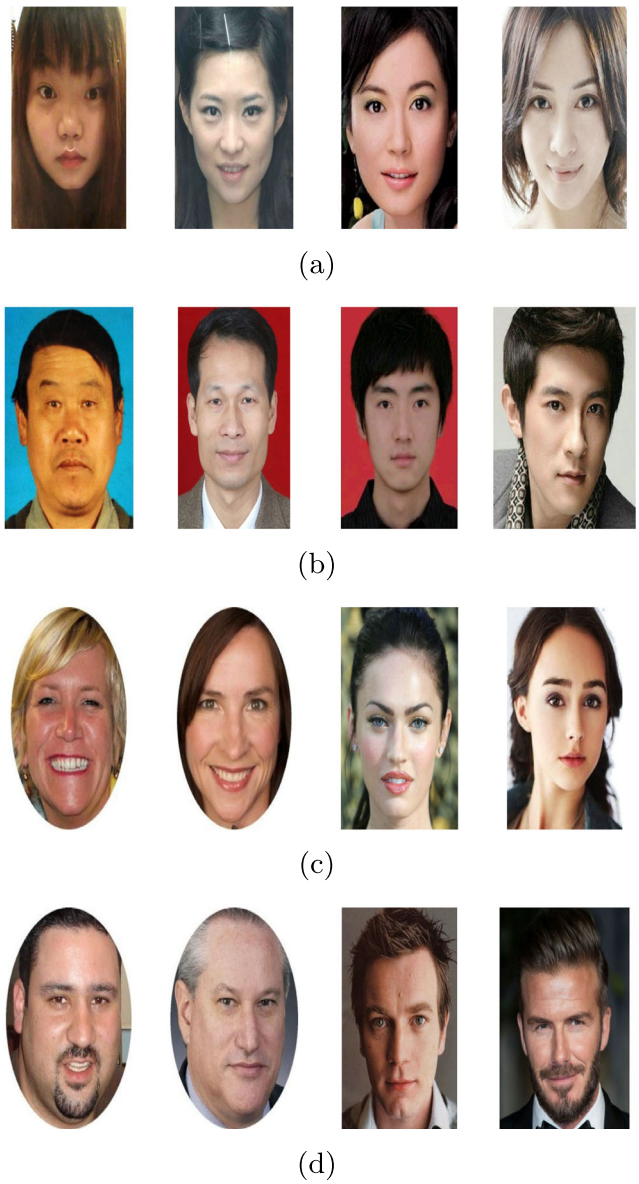

Fig. 6 Facial beauty samples from the SCUT-FBP5500 database, (a) Female Assian samples their score from left to the right are: 1.88, 3.00, 3.93 and 4.28. (b) Male Assian samples their score from left to the right are: 1.73, 2.48, 3.53 and 4.43. (c) Female Caucasian samples their score from left to the right are: 1.93, 2.87, 3.63 and 4.7. (d) Male Caucasian samples their score from left to the right are: 1.88, 2.67, 3.27 and 4.43

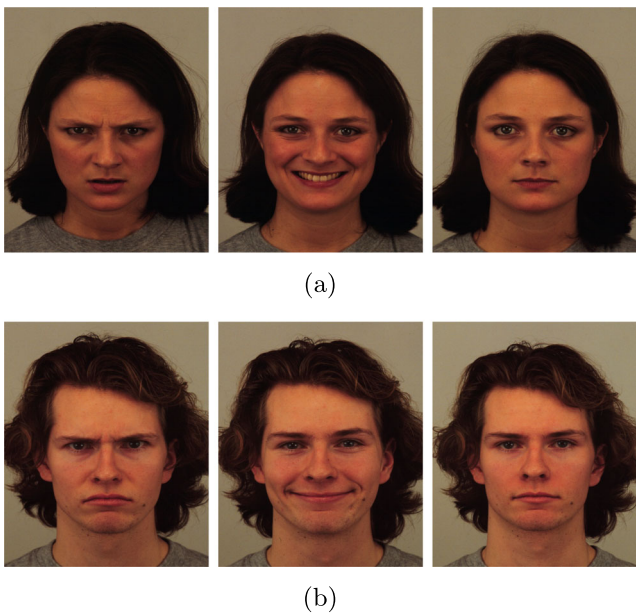

Fig. 7 Facial beauty samples from the KDEF-PT dataset, (a) Female samples their score from left to the right are: 2.78, 4.81 and 3.47, for anger, happy and neutral faces, respectively. (b) Male samples their score from left to the right are: 2.43, 3.34 and 3.24, for anger, happy and neutral faces, respectively

transfer learning (model fine-tuning) with the training part of the KDEF-PT dataset.

4.2 Evaluation metrics

To evaluate the performance of each model, four evaluation metrics are used which are: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Pearson Correlation coefficient (PC) and the ϵ-error. Let consider $Y = (y_1, y_2, ..., y_n)$ the ground-truth scores of the tested n images and $\hat{Y} = (\hat{y_1}, \hat{y_2}, ..., \hat{y_n})$ are their corresponding estimated scores. Here, n denotes the number of the tested face images. The evaluation metrics are defined as follows:

Mean Absolute Error (MAE): MAE is defined by:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{12}$$

MAE is scale-dependent accuracy measurement, this means MAE uses the same scale as the data being measured.

Root Mean Square Error (RMSE): RMSE is defined by:

$$RMSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{13}$$

The RMSE is another scale-dependent accuracy measure. Unlike MAE, the effect of any error on the RMSE is proportional to the squared error; thus, larger errors have a disproportionately large effect on the final RMSE. Consequently, the RMSE is sensitive to outliers.

Pearson Correlation coefficient (PC): PC was developed by Karl Pearson [47] and it is defined by:

$$PC = \frac{\sum_{i=1}^{n}(y_i - \overline{y_i})\left(\hat{y}_i - \overline{\hat{y}_i}\right)}{\sqrt{\sum_{\mathrm{i}=1}^{\mathrm{n}}\left(\mathrm{y_i} - \overline{\mathrm{y_i}}\right)^2}\sqrt{\sum_{i=1}^{n}\left(\hat{y}_i - \overline{\hat{y}_i}\right)^2}} \tag{14}$$

where $\overline{y_i}$ and $\overline{\hat{y}_i}$ are the mean of the ground-truth scores and the estimated scores, respectively. PC has a value between +1 and -1, it is a statistic that measures linear correlation between two variables Y and $\hat{Y}$. A value of +1 means total positive linear correlation, 0 means no linear correlation, and -1 means total negative linear correlation.

ϵ-error: ϵ-error is defined by:

$$\epsilon - error = \frac{1}{n}\sum_{i=1}^{n}\left(1 - exp\left(\frac{\left(y_i - \hat{y}_i\right)^2}{2\,\sigma_i^2}\right)\right) \tag{15}$$

where σ_i is the standard deviation of the scores of all raters of the image i. The value of ϵ-error is the accumulation of each image i error which based on the term $\epsilon - error_i = 1 - exp\,(\frac{(y_i - \hat{y}_i)^2}{2\,\sigma_i^2})$. When the absolute error of image i goes toward zero (i.e., $y_i = \hat{y}_i$), $\epsilon - error_i$ is zero. In contrast, when the absolute error is large $\epsilon - error$ takes into account the uncertainty of the rate which is represented by σ_i^2. In more details, the division by the term σ_i^2 provides a smaller contribution to the value of the ϵ error when the uncertainty in the rate is large and vice versa.

4.3 Experimental setup

All experiments are carried out on Pytorch library [48] with NVIDIA GPU Device GeForce TITAN RTX 24 GB. All Networks are trained for 40 epochs using Adam optimizer [49] and batch size of 15. The initial learning rate is 1e-4 for 20 epochs, then leaning rate decreases to 1e-5 for next 10 epochs, for the last 10 epochs the learning rate decreases to 1e-6. Active data augmentation is performed by rotating the input face by an angle between [-5, 5]. For all experiments, the reported results correspond to the best PC of the test data during the training/testing of the 40 epochs.

5 Performance evaluation on SCUT-FBP5500 dataset

5.1 Experimental results on the 60-40% split scenario

In this section, we limit the study to the provided 60-40% split.

5.1.1 Raw input vs the proposed face preprocessing

Preprocessing of faces is considered an important step for face analysis by machine learning. However, Deep Learning architectures are capable of learning high-level features in scenarios with shape and rotation variations. In this section, we investigate the impact of the preprocessing step on estimating the beauty of a face. For this purpose, we used ResneXt-50 and Inception-v3 with the default MSE loss function and considered two input scenarios : raw face images and aligned face images. Table 1 shows the results obtained. From this table, it can be seen that preprocessing the face image provides a significant improvement for the ResneXt-50 architecture. In contrast, this improvement is small for Inception-v3. In general, face alignment and cropping can support the training of CNN architectures by discarding the background features and prioritizing the face features.

5.1.2 Dynamic vs fixed loss function parameter

To investigate the effectiveness of the proposed dynamic law for the parametric robust loss functions, we use ResneXt-50 in two cases: (i) a loss adopting a fixed parameter and (ii) a loss adopting a dynamic parameter

Table 1 Face beauty prediction using ResneXt-50 and Inception-v3 networks with MSE loss function and two input image scenarios (The raw image and the detected face with our preprocessing scheme)

CNN architecture	Pre-processing	PC ↑	MAE ↓	RMSE ↓	ϵ-error ↓
ResneXt-50	Raw Image	0.9119	0.2126	0.2845	0.0853
ResneXt-50	Face Detection	0.9137	0.2107	0.2774	0.0807
Inception-v3	Raw Image	0.9108	0.2150	0.2831	0.0873
Inception-v3	Face Detection	0.9112	0.2147	0.2814	0.0833

using the parabolic law. The considered parametric loss functions are the proposed ParamSmoothL1, Huber and Tukey. For each parametric loss, we choose an interval for the dynamic law. For the fixed values, we choose the left and right limits of the interval and a set of values within the interval. The interval for the dynamic law is chosen experimentally and determined independently for each loss function. We compare the dynamic law not only with the fixed values, but also with their average.

The results obtained are summarized in Table 2. For both ParamSmoothL1 and Huber loss functions, the dynamic law interval is set to [0.7-0.3] and the fixed values are {0.7, 0.6, 0.5, 0.4, 0.3}. Table 2 shows that PramSmoothL1 and the Huber loss function with the proposed dynamic law perform better than the fixed values and their average. On the other hand, the dynamic interval of the c parameter of the Tukey loss function is set to [2-1.5] and the fixed c values are {2, 1.7, 1.5}. Similar to ParamSmoothL1 and Huber, the dynamic Tukey loss function using the parabolic law achieves better performance than using the fixed c values and their average. For the Tukey loss function, the dynamic interval of [2-1] achieved better performance than the Dynamic Tukey loss function of interval [2-1.5], as shown in Table 2. Based on these results, the dynamic intervals for the parameters α, β, and c are set to [0.7-0.3], [0.7-0.3], and [2-1] for ParamSmoothL1, Huber and Tukey, respectively.

Table 2 Comparison between dynamic and fixed parameters of ParamSmoothL1, Huber and Tukey loss functions using ResneXt-50 network

Loss Function	Parameter	PC ↑	MAE ↓	RMSE ↓	ϵ-error ↓
ParamSmoothL1	$\alpha = 0.7$	0.9122	0.2127	0.2799	0.0814
	$\alpha = 0.6$	0.9141	0.2098	0.2772	0.0803
	$\alpha = 0.5$	0.9132	0.2110	0.2780	0.0815
	$\alpha = 0.4$	0.9101	0.2146	0.2825	0.0833
	$\alpha = 0.3$	0.9116	0.2150	0.2810	0.0831
	Mean	0.9123	0.2126	0.2797	0.0819
	dynamic α (0.7-0.3)	**0.9150**	**0.2085**	**0.2744**	**0.0796**
Huber	$\beta = 0.7$	0.9126	0.2114	0.2796	0.0812
	$\beta = 0.6$	0.9130	0.2107	0.2780	0.0804
	$\beta = 0.5$	0.9144	0.2111	0.2770	0.0808
	$\beta = 0.4$	0.9124	0.2122	0.2783	0.0839
	$\beta = 0.3$	0.9110	0.2155	0.2811	0.0845
	Mean	0.9127	0.2122	0.2788	0.0822
	dynamic β (0.7-0.3)	**0.9149**	**0.2105**	**0.2757**	**0.0809**
Tukey	$c = 2.0$	0.9128	0.2155	0.2810	0.0837
	$c = 1.7$	0.9116	0.2133	0.2805	0.0821
	$c = 1.5$	0.9126	0.2129	0.2808	0.0824
	Mean	0.9123	0.2139	0.2808	0.0827
	dynamic c (2-1.5)	0.9138	0.2114	0.2778	0.0810
	dynamic c (2-1)	**0.9146**	**0.2093**	**0.2757**	**0.0798**

Best results are shown in bold

Table 3 Facial Beauty Prediction using ResneXt-50 Network with five loss functions (L_1, MSE, Dynamic ParamSmoothL1, Dynamic Huber and Dynamic Tukey losses)

Loss Function	PC ↑	MAE ↓	RMSE ↓	ϵ-error ↓
L_1	0.9126	0.2113	0.2783	0.0810
MSE	0.9137	0.2107	0.2774	0.0807
Dyn. ParamSmoothL1 (0.7-0.3)	**0.9150**	**0.2085**	**0.2744**	**0.0796**
Dyn. Huber (0.7-0.3)	0.9149	0.2105	0.2757	0.0809
Dyn. Tukey (2-1)	0.9146	0.2093	0.2757	0.0798

Best results are shown in bold

5.1.3 Two branches vs one branch

The goal of this section is to compare our proposed architecture with two backbones (2B-IncRex) with the pre-trained CNNs used to create 2B-IncRex. To this end, we used five loss functions (L_1, MSE, Dynamic ParamSmoothL1, Dynamic Huber, and Dynamic Tukey losses) to test ResneXt-50, Inception-v3, and 2B-IncRex, as shown in Tables 3, 4, and 5. respectively. In addition to the comparison between our proposed two-backbone architecture and the individual backbones, these results also show the comparison between our proposed dynamic loss functions and the standard loss functions (L_1 and MSE).

Based on the results of ResneXt-50 in Table 3, we can see that our proposed dynamic loss function ParamSmoothL1 achieves the best performance. Moreover, the other two dynamic loss functions (Huber and Tukey) obtained similar results to the Dynamic ParamSmoothL1 loss function and better performance than L_1 and MSE. This proves the efficiency of using the dynamic law not only compared to fixed parametric losses (as in Table 2), but also compared to other loss functions. From the Inception-v3 results in Table 4, we can also see that the dynamic loss functions give better results than L_1 and MSE. For the Inception-v3 architecture, the Dynamic Tukey loss function achieved the best performance.

The results of 2B-IncRex using the five loss functions are summarized in Table 5. Again, we note that the proposed dynamic loss functions achieve better performance than L_1 and MSE. On the other hand, the proposed Dynamic ParamSmoothL1 achieved the best performance for our proposed 2B-IncRex architecture. From the results of ResneXt-50, Inception-v3 and 2B-IncRex (from Tables 3, 4 and 5), we conclude that the proposed 2B-IncRex converges to the lowest error compared to ResneXt-50 and Inception-v3. This proves the effectiveness of our proposed CNN architecture with two backbones and the effectiveness of the proposed dynamic law for the robust loss functions.

5.1.4 CNN ensemble

The goal of this section is to use the trained models from Section 5.1.3 to improve the performance of FBP. To this end, we select the best models for the two individual backbones (ResneXt-50 and Inception) and the three best models of the proposed 2B-IncRex. In summary, three ensemble scenarios were tested. First, we combine the ensemble of the single backbones (ResneXt-50 and Inception). Second, the three best models of the proposed 2B-IncRex are combined. The third scenario is the combination of five models (best individual backbones and the best three 2B-IncRex, which corresponds to the trained models with the dynamic robust losses), the results are shown in Table 6. Since the creators of the SCUT -FBP5500 dataset provided two evaluation scenarios (60-40% and five fold cross-validation), each considering only training and test splits, we considered the last model after it was trained with 40 epochs. The goal of selecting the last

Table 4 Facial Beauty Prediction using Inception-v3 Network with five loss functions (L_1, MSE, Dynamic SmoothL1, Dynamic Huber and Dynamic Tukey)

Loss Function	PC ↑	L_1 ↓	RMSE ↓	ϵ-error ↓
L_1	0.9103	0.2152	0.2832	0.0848
MSE	0.9112	0.2147	0.2814	0.0833
Dyn. ParamSmoothL1 (0.7-0.3)	0.9132	0.2136	0.2803	0.0832
Dyn. Huber (0.7-0.3)	0.9127	0.2143	0.2793	0.0838
Dyn. Tukey (2-1)	**0.9149**	**0.2132**	**0.2781**	**0.0829**

Best results are shown in bold

Table 5 Facial Beauty Prediction using the proposed two backbones Network (2B-IncRex) with five loss functions (L_1, MSE, Dynamic ParamSmoothL1, Dynamic Huber and Dynamic Tukey losses)

Loss Function	PC ↑	MAE ↓	RMSE ↓	ϵ-error ↓
L_1	0.9123	0.2133	0.2802	0.0818
MSE	0.9145	0.2090	0.2755	0.0796
Dyn. ParamSmoothL1 (0.7-0.3)	**0.9171**	**0.2072**	**0.2716**	**0.0783**
Dyn. Huber (0.7-0.3)	0.9161	0.2082	0.2741	0.0791
Dyn. Tukey (2-1)	0.9164	0.2081	0.2734	0.0787

Best results are shown in bold

model in our ensemble approach is to use the test data only once. For an input image fed with regressors (trained CNN models), the ensemble is obtained by computing the average of the different regressors; this average is considered as the ensemble prediction. From the results of Table 6, we can observe the following:

- The ensemble of the two individual backbones outperforms the individual CNN backbones.
- The ensemble of our proposed 2B-IncRex trained with the proposed dynamic losses outperforms the ensemble of the two individual backbones.
- Finally, we find that the ensemble of the two single backbones and the three 2B-IncRex models improves the results compared to the previous ensemble schemes. Based on these results, we consider this last ensemble as our proposed solution for FBP. We refer to it as Dynamic ER-CNN, since it benefits from the proposed dynamic loss functions and ensemble of CNN architectures for the regression task.

From the above results, we conclude that all ensemble scenarios improve the face beauty estimation. Although the second ensemble scenario achieves a better result than the first, the combination of both ensemble scenarios further improves the results. This proves that the individual backbones can provide a diversity estimator for the proposed Dynamic ER-CNN solution.

5.2 Experimental results using the five fold cross-validation scenario

In addition to the 60-40% evaluation scheme of the SCUT -FBP5500 dataset, the creator of this dataset has provided five-fold cross-validation splits. In this section, we will test the best identified solutions from 60-40% for one and two backbones. Specifically, these are the following solutions: ResneXt-50 trained with Dynamic ParamSmoothL1, Inception-v3 trained with Dynamic Tukey and 2B-IncRex trained with the three dynamic robust loss functions (ParamSmoothL1, Huber and Tukey). The obtained results are summarized in Table 7. For each architecture and corresponding loss function, we report the results using four evaluation metrics (PC, MAE, RMSE and ϵ-error) for each fold and its average over the five folds.

Similar to the results of 60-40% split, two backbones architecture achieve higher performance than the single backbones, again this proves the efficiency of the proposed 2B-IncRex architecture. On the other hand, we notice that the two backbones architecture achieves close performance using different dynamic robust losses, with small preference for the Dynamic ParamSmoothL1 loss function based on MAE, RMSE and ϵ-error metrics. Similar to the 60-40% split results, the architecture with two backbones achieves higher performance than the one with one backbone, which again proves the efficiency of the proposed 2B-IncRex architecture. On the other hand, we find that the architecture with two backbones achieves similar performance when using different dynamic robust losses, slightly favoring the Dynamic ParamSmoothL1 loss function based on MAE, RMSE and ϵ error metrics.

Similar to Section 5.1.4, we tested three ensemble scenarios, (i) the ensemble of the single backbones (ResneXt-50 and Inception), (ii) the ensemble of the proposed 2B-IncRex architecture trained with the three dynamic robust losses, (iii) the ensemble of all models used in the first two scenarios (i and ii), denoted by Dynamic CNN-ER. Table 8 summarizes the obtained results of the three ensemble scenarios for each fold and their mean. From

Table 6 Facial Beauty Prediction using the proposed CNN ensemble of different trained models on 60-40% data split

Fusion scheme	PC ↑	MAE ↓	RMSE ↓	ϵ-error ↓
One Backbone (2 models)	0.9189	0.2071	0.2711	0.0785
Two Backbones (3 models)	0.9201	0.2045	0.2685	0.0765
Dynamic ER-CNN (Mixture models)	**0.9212**	**0.2037**	**0.2672**	**0.0762**

Best results are shown in bold

Table 7 Five folds cross-validation of Facial Beauty Prediction using single backbone networks (Inception-v3 with Dynamic Tukey loss and ResneXt-50 with Dynamic ParamSmoothL1 loss) and two backbones networks (2B-IncRex wit Dynamic ParamSmoothL1, Dynamic Huber, and Dynamic Tukey losses)

Architecture	Fold	PC ↑	MAE ↓	RMSE ↓	ϵ-error ↓
Inception-v3 with Dynamic Tukey loss function	Fold 1	0.9139	0.2138	0.2792	0.0803
	Fold 2	0.9147	0.2062	0.2794	0.0792
	Fold 3	0.9188	0.2124	0.2774	0.0791
	Fold 4	0.9239	0.2094	0.2686	0.0797
	Fold 5	0.9214	0.2066	0.2675	0.0763
	Mean	**0.9185**	**0.2097**	**0.2744**	**0.0789**
ResneXt-50 with Dynamic ParamSmoothL1 loss function	Fold 1	0.9176	0.2094	0.2740	0.0780
	Fold 2	0.9155	0.2067	0.2780	0.0789
	Fold 3	0.9214	0.2080	0.2739	0.0774
	Fold 4	0.9218	0.2051	0.2688	0.0765
	Fold 5	0.9196	0.2073	0.2713	0.0777
	Mean	**0.9192**	**0.2073**	**0.2732**	**0.0777**
2B-IncRex with Dynamic ParamSmoothL1 loss function	Fold 1	0.9175	0.2084	0.2762	0.0770
	Fold 2	0.9166	0.2073	0.2789	0.0786
	Fold 3	0.9220	0.2072	0.2733	0.0775
	Fold 4	0.9279	0.2013	0.2604	0.0737
	Fold 5	0.9266	0.1959	0.2596	0.0703
	Mean	**0.9221**	**0.2040**	**0.2697**	**0.0754**
2B-IncRex with Dynamic Huber loss function	Fold 1	0.9206	0.2046	0.2706	0.0753
	Fold 2	0.9196	0.2083	0.2725	0.0783
	Fold 3	0.9254	0.2049	0.2687	0.0755
	Fold 4	0.9250	0.2036	0.2637	0.0752
	Fold 5	0.9220	0.2053	0.2694	0.0775
	Mean	**0.9225**	**0.2053**	**0.2690**	**0.0763**
2B-IncRex with Dynamic Tukey loss function	Fold 1	0.9216	0.2060	0.2699	0.0761
	Fold 2	0.9138	0.2097	0.2821	0.0806
	Fold 3	0.9251	0.2088	0.2733	0.0775
	Fold 4	0.9247	0.2040	0.2653	0.0766
	Fold 5	0.9243	0.2014	0.2633	0.0734
	Mean	**0.9219**	**0.2060**	**0.2708**	**0.0768**

Best results are shown in bold

the results for one and two backbones (Table 7) and the ensemble scenarios (Table 8), we notice the following:

- The fusion of the individual backbones (scenario (i)) performs better than the individual backbone networks (Inception-v3 with Dynamic Tukey and ResneXt-50 with Dynamic PramSmoothL1 loss function).
- The second ensemble scenario shows that the ensemble of 2B-IncRex outperforms all the results obtained by the single two backbone networks (2B-IncRex with the three dynamic robust loss functions).
- Our proposed ensemble approach Dynamic CNN-ER (scenario (iii)) outperforms not only single and 2B-IncRex networks, but also their combination.

The comparison between the results of Tables 7 and 8 proves the effectiveness of the proposed Dynamic ER-CNN for the assessment of the beauty of the face.

6 Performance evaluation on KDEF-PT dataset

In this experiment, we used the KDEF-PT dataset [11], which contains ratings of facial beauty in the presence of facial expressions. Table 9 summarizes the results obtained with the selected individual CNN architectures in our ensemble trained with the proposed dynamic loss functions.

Table 8 Five folds cross-validation of Facial Beauty Prediction using the proposed CNN ensemble of different trained models

Fusion scheme	Fold	PC ↑	MAE ↓	RMSE ↓	ϵ-error ↓
One Branch (2 models)	Fold 1	0.9203	0.2071	0.2704	0.0761
	Fold 2	0.9198	0.2009	0.2723	0.0757
	Fold 3	0.9243	0.2056	0.2695	0.0752
	Fold 4	0.9274	0.2007	0.2604	0.0741
	Fold 5	0.9243	0.2023	0.2636	0.0739
	Mean	**0.9232**	**0.2033**	**0.2672**	**0.075**
Two Branches (3 models)	Fold 1	0.9236	0.2015	0.2663	0.0732
	Fold 2	0.9204	0.2032	0.2721	0.0762
	Fold 3	0.9278	0.2028	0.2666	0.0742
	Fold 4	0.9293	0.1986	0.2580	0.0727
	Fold 5	0.9277	0.1959	0.2579	0.0706
	Mean	**0.9257**	**0.2004**	**0.2642**	**0.0734**
Dynamic ER-CNN (Mixture 5 models)	Fold 1	0.9240	0.2022	0.2656	0.0731
	Fold 2	0.9216	0.2004	0.2701	0.0750
	Fold 3	0.9279	0.2023	0.2657	0.0736
	Fold 4	0.9299	0.1978	0.2568	0.0722
	Fold 5	0.9275	0.1966	0.2583	0.0709
	Mean	**0.9262**	**0.1998**	**0.2633**	**0.0730**

Best results are shown in bold

Similar to the ensemble experiments in the SCUT-FBP5500 dataset, three ensemble scenarios are evaluated. From the results of Table 9, we can make the following observations:

- The proposed 2B-IncRex trained by various dynamic robust losses performs better than the individual backbones and their ensemble.
- The fusion of individual backbones performs better than the individual backbone networks (Inception-v3 with Dynamic Tukey and ResneXt-50 with Dynamic PramSmoothL1 loss function).
- 2B-IncRex-based ensemble scenario outperforms all results obtained by the single two backbone networks (2B-IncRex with the three dynamic robust loss functions).
- Our proposed ensemble approach Dynamic CNN-ER outperforms not only single and 2B-IncRex networks, but also their combination.

Despite the presence of facial expressions and a limited amount of data, our approach can achieve very good performance in estimating facial beauty using transfer learning (model fine-tuning). Moreover, the ϵ-error shows

Table 9 Facial Beauty Prediction using single backbone CNN architectures (Inception-v3 and Resnext-50) and our proposed 2B-IncRex architecture. Furthermore, the ensemble of these approaches and our proposed Dynamic ER-CNN approach are evaluated. All these methods are tested on KDEF-PT dataset

Architecture	PC ↑	MAE ↓	RMSE ↓	ϵ-error ↓
Inception-v3 with Dynamic Tukey loss	0.9068	0.3402	0.4474	0.0519
Resnext-50 with Dynamic ParamSmoothL1 loss	0.9031	0.3500	0.4267	0.0496
2B-IncRex with Dynamic ParamSmoothL1 loss	0.9153	0.2864	0.3769	0.0404
2B-IncRex with Dynamic Huber loss	0.9132	0.3130	0.3934	0.0428
2B-IncRex with Dynamic Tukey loss	0.9151	0.3099	0.4060	0.0444
Ensemble of Single Backbones (2 models)	0.9111	0.3311	0.4270	0.0482
Ensemble of Two Backbones (3 models)	0.9232	0.2988	0.3805	0.0396
Dynamic ER-CNN	**0.9260**	**0.2839**	**0.3638**	**0.0371**

Best results are shown in bold

Table 10 Comparison with the sate-of-the-arts methods using the 60-40% split. Dynamic ParamSmoothL* is our 2B-IncRex network that was trained using the Dynamic ParamSmoothL1 loss function

Method	PC ↑	MAE ↓	RMSE ↓
LR (2018) [18]	0.5948	0.4289	0.5531
GR (2018) [18]	0.6738	0.3914	0.5085
SVR (2018) [18]	0.6668	0.3898	0.5132
Alexnet (2018) [18]	0.8298	0.2938	0.3819
Resnet-18 (2018) [18]	0.8513	0.2818	0.3703
ResneXt-50 (2018) [18]	0.8777	0.2518	0.3325
CNN with SCA (2020) [25]	0.8780	0.2517	0.3320
Dynamic ParamSmoothL1* (Ours)	**0.9171**	**0.2072**	**0.2716**
Dynamic ER-CNN (Ours)	**0.9212**	**0.2037**	**0.2672**

Best results are shown in bold

Table 11 Comparison with the sate-of-the-arts methods the five folds cross-validation scenario. $^{+}$ the authors of [27] used ResNeXt-50 as the backbone network to re-implement the methods [50] and [51] on the newly-constructed SCUT-FBP5500 dataset. Dynamic ParamSmoothL* is our 2B-IncRex network that was trained using the Dynamic ParamSmoothL1 loss function

	1	2	3	4	5	Mean
PC ↑						
Alexnet (2018) [18]	0.8667	0.8645	0.8615	0.8678	0.8566	0.8634
Resnet-18 (2018) [18]	0.8847	0.8792	0.8929	0.8932	0.9004	0.8900
ResneXt-50 (2018) [18]	0.8985	0.8932	0.9016	0.899	0.9064	0.8997
CNN with SCA (2020) [25]	0.8990	0.8939	0.9020	0.8999	0.9067	0.9003
PI-CNN (2017) [51]$^{+}$	-	-	-	-	-	0.8978
CNN + LDL (2017) [50]$^{+}$	-	-	-	-	-	0.9031
ResNet-18 based AaNet (2019) [26]	-	-	-	-	-	0.9055
ResneXt-50-R^{3}CNN (2019) [27]	0.9143	0.9066	0.9136	0.9146	0.9217	0.9142
Dynamic ParamSmoothL1* (Ours)	0.9175	0.9166	0.9220	0.9279	0.9266	0.9221
Dynamic ER-CNN (Ours)	**0.9240**	**0.9216**	**0.9279**	**0.9299**	**0.9275**	**0.9262**
MAE ↓						
Alexnet (2018) [18]	0.2633	0.2605	0.2681	0.2609	0.2728	0.2651
Resnet-18 (2018) [18]	0.2480	0.2459	0.243	0.2383	0.2383	0.2419
ResneXt-50 (2018) [18]	0.2306	0.2285	0.226	0.2349	0.2258	0.2291
CNN with SCA (2020) [25]	0.2300	0.2284	0.2257	0.2345	0.2251	0.2287
PI-CNN (2017) [51]$^{+}$	-	-	-	-	-	0.2267
CNN + LDL (2017) [50]$^{+}$	-	-	-	-	-	0.2201
ResNet-18 based AaNet (2019) [26]	-	-	-	-	-	0.2236
ResneXt-50-R^{3}CNN (2019) [27]	0.2109	0.2152	0.2126	0.2130	0.2085	0.2120
Dynamic ParamSmoothL1* (Ours)	0.2084	0.2073	0.2072	0.2013	0.1959	0.2040
Dynamic ER-CNN (Ours)	**0.2022**	**0.2004**	**0.2023**	**0.1978**	**0.1966**	**0.1998**
RMSE ↓						
Alexnet (2018) [18]	0.3408	0.3449	0.3538	0.3438	0.3576	0.3481
Resnet-18 (2018) [18]	0.3258	0.3286	0.3184	0.3107	0.2994	0.3166
ResneXt-50 (2018) [18]	0.3025	0.3084	0.3016	0.3044	0.2918	0.3017
CNN with SCA (2020)[25]	0.3020	0.3081	0.3013	0.3039	0.2916	0.3014
PI-CNN (2017) [51]$^{+}$	-	-	-	-	-	0.3016
CNN + LDL (2017) [50]$^{+}$	-	-	-	-	-	0.2940
ResNet-18 based AaNet (2019) [26]	-	-	-	-	-	0.2954
ResneXt-50-R^{3}CNN (2019) [27]	0.2767	0.2895	0.2837	0.2804	0.2701	0.2800
Dynamic ParamSmoothL1* (Ours)	0.2762	0.2789	0.2733	0.2604	0.2596	0.2697
Dynamic ER-CNN (Ours)	**0.2656**	**0.2701**	**0.2657**	**0.2568**	**0.2583**	**0.2633**

that our approach achieves very good performance despite the high labeling uncertainty of the ground truth. From the above results and discussion, it is clear that all of our proposed elements (2B-IncRex, Dynamic Law for Robust Loss Function, and the Ensemble) and their combination prove their efficiency for FBP. As far as we know, this is the first time that facial beauty estimation has been evaluated using machine learning methods in the presence of facial expressions on the dataset KDEF-PT.

7 Discussion and comparison

The goal of this section is to compare the performance of our proposed solutions with state-of-the-art approaches. In summary, this comparison examines the two evaluation scenarios of SCUT-FBP5500 (60-40% and five-fold cross-validation). Table 10 summarizes the comparison with the state-of-the-art approaches using the first evaluation scenario (60-40% split). The comparison consists of two parts. First, we compare our proposed Dynamic ER-CNN with the state-of-the-art approaches in three evaluation metrics (PC, MAE, and RMSE). This comparison shows that our proposed Dynamic ER-CNN outperforms state-of-the-art methods. In addition to the first comparison, our proposed 2B-IncRex architecture trained with the proposed Dynamic PramSmoothL1 loss function achieves better performance than the state-of-the-art approaches. The above comparisons prove that the superiority of our approach over the state-of-the-art methods is not only due to the ensemble of models, but that both the proposed 2B-IncRex network and the dynamic parabolic law for the robust loss functions played a crucial role in achieving such performance.

Similar to the comparison with state-of-the-art approaches for the 60-40% assessment scenario, we used our proposed Dynamic ER-CNN and Dynamic PramSmoothL1 of the 2B-IncRex architecture for the five-fold scenario. From Table 11, our approach (Dynamic ER-CNN) outperforms all state-of-the-art methods on the three evaluation metrics. Moreover, our Dynamic ER-CNN not only outperforms the state-of-the-art methods, but also our two proposed backbones with Dynamic ParamSmoothL1 achieve better performance than the state-of-the-art methods. This confirms that the strength of our proposed Dynamic ER-CNN is not only due to the ensemble of different regressors, but both the proposed 2B-IncRex network and the dynamic loss functions play a crucial role in outperforming state-of-the-art methods. The efficiency of our proposed Dynamic ER-CNN has been demonstrated in both 60-40% and five-fold cross-validation.

8 Conclusion

In this paper, we presented a framework based on an ensemble of regression CNNs (Dynamic ER-CNN). Our proposed approach averages the output of five trained CNN architectures. The CNNs used are ResneXt-50, Inception-v3, and the proposed 2B-IncRex architectures. These architectures were trained with the proposed Dynamic ParamSmoothL1, Dynamic Huber, and Dynamic Tukey. For these dynamic loss functions (ParamSmoothL1, Huber and Tukey), a parabolic law is proposed to reduce the parameter of the loss. The dynamic schemes were found to be very efficient both in terms of performance and in avoiding the grid search for the best value, which has a high computational cost. Moreover, the dynamic loss functions performed better than two standard loss functions, namely L_1 and MSE.

The obtained results show the superiority of the proposed 2B-IncRex over ResneXt-50 and Inception-v3 networks. Moreover, the proposed approach (Dynamic ER-CNN) outperformed not only one and two branches networks, but also their fused models. On the other hand, the proposed approach performed better than the state-of-the-art methods in both the 60-40% and cross-validation experiments for the three evaluation metrics (PC, MAE and RMSE) onthe SCUT-FBP5500 dataset. The experimental results on KDEF-PT proved the efficiency of our approach for estimating facial beauty with adopting transfer learning, despite the presence of facial expressions and limited data. We also found that using the proposed dynamic robust loss functions generally leads to better estimates

Acknowledgment This work was partially funded by the Spanish Ministerio de Ciencia, Innovación y Universidades, Programa Estatal de I+D+i Orientada a los Retos de la Sociedad, RTI2018-101045-B-C21.

The authors thank Arturo Argentieri of CNR-ISASI, Italy, for his assistance with the multi-GPU computing facilities.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

References

1. Dion K, Berscheid E, Walster E (1972) What is beautiful is good. J Personal Soc Psychol 24(3):285. Publisher: American Psychological Association

2. Gan J, Xiang L, Zhai Y, Mai C, He G, Zeng J, Bai Z, Labati RD, Piuri V, Scotti F (2020) 2M BeautyNet: facial beauty prediction based on multi-task transfer learning. IEEE Access 8:20245–20256. Publisher: IEEE
3. Eisenthal Y, Dror G, Ruppin E (2006) Facial attractiveness: Beauty and the machine. Neural Comput 18(1):119–142. Publisher: MIT Press
4. Liu X, Li T, Peng H, Ouyang IC, Kim T, Wang R (2019) Understanding beauty via deep facial features. In: 2019 IEEE/CVF Conference on computer vision and pattern recognition workshops (CVPRW), pp 246–256
5. Alashkar T, Jiang S, Fu Y (2017) Rule-based facial makeup recommendation system. In: IEEE. 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)
6. Laurentini A, Bottino A (2014) Computer analysis of face beauty: A survey. Comput Vis Image Underst 125:184–199. Publisher: Elsevier
7. Liang L, Jin L, Li X (2014) Facial skin beautification using adaptive region-aware masks. IEEE Trans Cybern 44(12):2600–2612. Publisher: IEEE
8. Xu L, Fan H, Xiang J (2019) Hierarchical multi-task network for race, gender and facial attractiveness recognition. In: 2019 IEEE International Conference on Image Processing (ICIP), pp 3861–3865. IEEE
9. Vantaggiato E, Paladini E, Bougourzi F, Distante C, Hadid A, Taleb-Ahmed A (2021) COVID-19 recognition using ensemble-CNNs in two new chest X-ray databases. Sensors 21(5):1742. Publisher: Multidisciplinary Digital Publishing Institute. Accessed 2022-03-24
10. Bougourzi F, Distante C, Ouafi A, Dornaika F, Hadid A, Taleb-Ahmed A (2021) Per-COVID-19: A Benchmark Dataset for COVID-19 Percentage Estimation from CT-Scans. Journal of Imaging 7(9):189. https://doi.org/10.3390/jimaging7090189. Publisher: Multidisciplinary Digital Publishing Institute. Accessed 2022-03-24
11. Garrido MV, Prada M (2017) KDEF-PT: valence, emotional intensity, familiarity and attractiveness ratings of angry, neutral, and happy faces
12. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M (2015) Imagenet large scale visual recognition challenge. Int J Comput Vis 115(3):211–252. Publisher: Springer
13. Bottino A, Laurentini A (2010) The analysis of facial beauty: an emerging area of research in pattern analysis. In: International conference image analysis and recognition, pp 425–435. Springer
14. Gan J, Zhou L, Zhai Y (2015) A study for facial beauty prediction model. In: 2015 International conference on wavelet analysis and pattern recognition (ICWAPR), pp 8–13. IEEE
15. Rhazi ME, Zarghili A, Majda A, Bouzalmat A, Oufkir AA (2019) Facial beauty analysis by age and gender. Int J Intell Syst Technol Appl 18(1-2):179–203
16. Xie D, Liang L, Jin L, Xu J, Li M (2015) Scut-fbp: A benchmark dataset for facial beauty perception
17. Xu L, Xiang J, Yuan X (2018) Transferring rich deep features for facial beauty prediction
18. Liang L, Lin L, Jin L, Xie D, Li M (2018) SCUT-FBP5500: a diverse benchmark dataset for multi-paradigm facial beauty prediction
19. Gray D, Yu K, Xu W, Gong Y (2010) Predicting facial beauty without landmarks
20. Zhang D, Zhao Q, Chen F (2011) Quantitative analysis of human facial beauty using geometric features. Pattern Recogn 44(4):940–950
21. Aarabi P, Hughes D, Mohajer K, Emami M (2001) The automatic measurement of facial beauty
22. Yan H (2014) Cost-sensitive ordinal regression for fully automatic facial beauty assessment. Neurocomputing 129:334–342. Publisher: Elsevier
23. Chiang W-C, Lin H-H, Huang C-S, Lo L-J, Wan S-Y (2014) The cluster assessment of facial attractiveness using fuzzy neural network classifier based on 3D Moiré features. Pattern Recogn 47(3):1249–1260. https://doi.org/10.1016/j.patcog.2013.09.007. Accessed 2021-01-23
24. Fan J, Chau KP, Wan X, Zhai L, Lau E (2012) Prediction of facial attractiveness from facial proportions. Pattern Recogn 45(6):2326–2334. https://doi.org/10.1016/j.patcog.2013.09.007. Accessed 2021-01-23
25. Cao K, Choi K-n, Jung H, Duan L (2020) Deep learning for facial beauty prediction. Information 11(8):391. Publisher: Multidisciplinary Digital Publishing Institute
26. Lin L, Liang L, Jin L, Chen W (2019) Attribute-aware convolutional neural networks for facial beauty prediction. In: IJCAI, pp 847–853
27. Lin L, Liang L, Jin L (2019) Regression guided by relative ranking using convolutional neural network (R3CNN) for facial beauty prediction
28. Ahonen T, Hadid A, Pietikainen M (2006) Face description with local binary patterns: Application to face recognition. IEEE Trans Pattern Anal Mach Intell 28(12):2037–2041
29. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110. Publisher: Springer
30. Cao Z, Yin Q, Tang X, Sun J (2010) Face recognition with learning-based descriptor
31. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105
32. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
33. Xie S, Girshick R, Dollár P, Tu Z, He K (2017) Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1492–1500
34. Dornaika F, Moujahid A, Wang K, Feng X (2020) Efficient deep discriminant embedding: Application to face beauty prediction and classification. Eng Appl Artif Intell 95:103831. https://doi.org/10.1016/j.engappai.2020.103831. Accessed 2020-08-30
35. Dornaika F, Wang K, Arganda-Carreras I, Elorza A, Moujahid A (2020) Toward graph-based semi-supervised face beauty prediction. Expert Syst Appl 142:112990. Publisher: Elsevier
36. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2818–2826
37. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
38. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9
39. Bougourzi F, Mokrani K, Ruichek Y, Dornaika F, Ouafi A, Taleb-Ahmed A (2019) Fusion of transformed shallow features for facial expression recognition. IET Image Process 13(9):1479–1489. https://doi.org/10.1049/iet-ipr.2018.6235. Publisher: IET Digital Library. Accessed 2020-10-19
40. Bougourzi F, Dornaika F, Mokrani K, Taleb-Ahmed A, Ruichek Y (2020) Fusing Transformed Deep and Shallow features (FTDS) for image-based facial expression recognition. Expert Syst Appl 156:113459

41. King DE (2009) Dlib-ml: A machine learning toolkit. J Mach Learn Res 10(Jul):1755–1758
42. Girshick R (2015) Fast r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision, pp 1440–1448
43. Loshchilov I, Hutter F (2017) SGDR: Stochastic gradient descent with warm restarts. In: International conference on learning representation
44. Huber PJ (1992) Robust estimation of a location parameter
45. Black MJ, Rangarajan A (1996) On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. Int J Comput Vis 19(1):57–91. Publisher: Springer
46. Belagiannis V, Rupprecht C, Carneiro G, Navab N (2015) Robust optimization for deep regression. In: Proceedings of the IEEE international conference on computer Vision, pp 2830–2838
47. Pearson K (1895) VII. Note On regression and inheritance in the case of two parents. Proc R Soc London 58(347-352):240–242
48. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L (2019) Pytorch: An imperative style, high-performance deep learning library. In: Advances in neural information processing systems, pp 8026–8037
49. Kingma DP, Ba J (2014) Adam: A, method for stochastic optimization
50. Fan Y-Y, Liu S, Li B, Guo Z, Samal A, Wan J, Li SZ (2017) Label distribution-based facial attractiveness computation by deep residual learning. IEEE Trans Multimedia 20(8):2196–2208
51. Xu J, Jin L, Liang L, Feng Z, Xie D, Mao H (2017) Facial attractiveness prediction using psychologically inspired convolutional neural network (PI-CNN)

Affiliations

Fares Bougourzi[1] · Fadi Dornaika[2,3,4] · Nagore Barrena[3] · Cosimo Distante[1] · Abdelmalik Taleb-Ahmed[5]

Fares Bougourzi
fares.bougourzi@isasi.cnr.it

Nagore Barrena
nagore.barrena@ehu.eus

Cosimo Distante
cosimo.distante@cnr.it

Abdelmalik Taleb-Ahmed
Abdelmalik.Taleb-Ahmed@uphf.fr

1 Institute of Applied Sciences and Intelligent Systems, National Research Council of Italy, Lecce, 73100, Italy

2 Henan Key Laboratory of Big Data Analysis and Processing, Henan University, Kaifeng, China

3 University of the Basque Country UPV/EHU, San Sebastian 20018, Basque Country, Spain

4 IKERBASQUE, Basque Foundation for Science, Bilbao, 48012, Basque Country, Spain

5 Universite Polytechnique Hauts-de-France, Université de Lille, 969 CNRS, Valenciennes, 59313, Hauts-de-France, France