Facial Beauty Prediction Using Hybrid CNN Architectures and Dynamic Robust Loss Function

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

Facial Beauty Prediction Using Hybrid CNN Architectures and Dynamic Robust Loss Function

F. Dornaika[1,2], F. Bougourzi[3,*], A. Taleb-Ahmed [4], C. Distante[3]
[1] University of the Basque Country UPV/EHU, San Sebastian, Spain
[2] IKERBASQUE, Basque Foundation for Science, Bilbao, Spain
[3] Institute of Applied Sciences and Intelligent Systems, National Research Council of Italy, 73100 Lecce, Italy;
[4] Univ. Polytechnique Hauts-de-France, Univ. Lille, CNRS, Centrale Lille, UMR 8520 - IEMN, F-59313 Valenciennes, France
[*] Corresponding Author

***Abstract*—In the last decade, several studies have shown that facial attractiveness can be learned by machines. In this paper, we address Facial Beauty Prediction from static images. The paper contains two main contributions. First, we propose a two-branch architecture (REX-INCEP) based on merging the architecture of two already trained networks to deal with the complicated high-level features associated with the FBP problem. Second, we introduce the use of a dynamic law to control the behavior of the following robust loss functions during training: new ParamSmoothL1 loss function formulas.**

Our approach is evaluated on the SCUT-FBP5500 database using the five-fold cross-validation protocol. Our proposed REX-INCEP architecture outperforms several CNN architectures and our proposed dynamic ParamSmoothL1 loss function outperforms traditional loss functions (L1 and MSE) and fixed ParamSmoothL1 loss function. On the other hand, our approach outperforms the state of the art approaches on several metrics. 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.

***Index Terms*—Facial Beauty Prediction, Convolutional Neural Network, Deep Learning, Robust Loss Functions.**

I. Introduction

The mysterious concept of beauty has always attracted people to discover its true meaning [1]. For a long time, Eastern and Western philosophers, psychologists and artists have pondered and researched the nature and composition of the beauty of the human face. In today's society the study of the beauty of the face has an important practical significance. Beauty is big business: according to a recent study, the beauty and cosmetics market generates an estimated 445 billion in annual revenue worldwide and continues to grow rapidly (an estimated 750 billion by 2024). The rapidly growing beauty market urgently needs a more precise definition of the beauty standard for faces. In animation and computer game design, research on the beauty of faces can also serve as a reference for designers creating virtual characters. Compared to other facial image analysis tasks, such as face recognition, facial expression recognition, gender classification, age estimation, and ethnicity classification, evaluating the beauty of faces is more challenging because it is difficult to apply a well-defined concept to describe the beauty information of facial images, which is a key problem in this field. Applications for facial beauty estimation and prediction include: Cosmetic recommendations [2], scheduling of aesthetic surgeries [3], facial beautification [4], and Social Networks Services (SNS) (such as Facebook, Instagram, and dating websites) [5].

Predicting facial beauty in images using machine learning got some progresses in the last decade [6],[7]. In particular, some researchers proposed deep learning solutions to solve the prediction problem in end-to-end fashion [8], [9].

In this paper, we propose a CNN-based approach which has two main contributions. First, we propose to combine two different powerful CNN architectures into a single architecture (called the two-branch architecture) that is trained end-to-end. Second, we propose to build an adaptive robust loss function for training the resulting architecture.

In summary, the main contributions of this paper are as follows:

- We propose two branches network (REX-INCEP) for face beauty estimation based on ResneXt-50 and Inception-v3 architectures. Moreover, our REX-INCEP architecture provides the right trade-off between the performance and the number of parameters for facial beauty prediction.
- We introduce ParamSmoothL1 regression loss function. This loss can change its parameter during training. This can solve the problem of complexity in finding the best loss function parameter.

The rest of the paper is structured as follows. Section II describes some related works. Section III describes the proposed method. The obtained experimental results are presented in section IV. Finally, section V concludes the paper.

II. Related Work

In the last decade, estimating the beauty of faces from static images has attracted increasing interest from the computer vision and machine learning community due to its wide range of applications. Indeed, the methods used to estimate facial

beauty can be divided into hand-crafted [10], [11], [12], [13], [14], [15], [16], [17] or deep learning methods [11], [18], [8], [9]. The hand-crafted methods are based on facial geometry [11], [14] or appearance [11], [15].

In addition to hand-crafted methods, many CNN-based approaches to FBP have been proposed. In [11], L. Liang et al. presented their facial beauty database (SCUT-FBP5500) with regression scores . They tested three CNN architectures (Alexnet [19], Resnet-18 [20], and ResneXt-50 [21]). Their results show that the ResneXt-50 architecture outperformed the other two deep architectures (Alexnet and Resnet-18). Moreover, the deep architectures performed better than the hand-crafted features they used with different shallow regressors.

K. Cao et al. [18] used a residual-in-residual (RIR) block to build a deeper network with multi-level skip connections to produce 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 [11] database and showed good performance. In [9], L. Lin et al. propose an R^3CNN architecture consisting of two main components. The first component is a regression component that contains two identical regression subnetworks that consistently map each face image to a beauty value. The second component is a ranking component that uses the Siamese network to learn a pairwise ranking that guides the beauty prediction regression. Their architecture showed promising results on the SCUT-FBP [22] and SCUT-FBP5500 [11] databases.

In addition to supervised learning, semi-supervised learning shows promising results for facial beauty estimation [6] and [7].

III. Proposed method

In this section, we will present the used CNN architectures, our proposed CNN solution and the proposed dynamic robust loss.

A. Backbone CNN Architectures

In the last decade, CNNs have become a dominant approach in many computer vision tasks. Consequently, numerous CNN architectures have been proposed. In our work, we will use two popular CNN architectures (ResneXt-50 [21] and Inception-v3 [23]) as building blocks for our REX-INCEP architecture. In our proposal, we use the above pre-trained models trained on the ImageNet challenge database [24]. In this section, we will briefly introduce ResneXt-50 and Inception-v3 architectures, which were used as backbone architectures in our proposed REX-INCEP solution.

a) ResneXt-50 Architecture: The architecture of ResneXt-50 is presented in [21], which is based on the ResneXt module. The ResneXt module performs a series of transformations, each based on a low-dimensional embedding and sharing the same topology. The results of all transformations are combined by summation.

b) Inception-v3 Architecture: The Inception-v3 architecture is presented in [23], which is based on the Inception module presented in [25]. The main idea of the inception architecture is to combine different convolutional layers with different kernel sizes and pooling layers in one inception module.

B. Our Approach

Our approach is based on two main components: (i) the deep network with two branches (REX-INCEP) (see Section III-B2) and (ii) the dynamic robust loss functions (see Section III-B3). In more details, we train the proposed two-branch deep network (REX-INCEP) using our proposed dynamic ParamSmoothL1 loss function. The two-branch deep network consists of ResneXt-50 and inception-v3, which are merged into a single architecture.

1) **Face Preprocessing***:* In the face preprocessing phase, we used the 2D alignment scheme proposed in [26] and [27]. The 2D face alignment and cropping are depicted in Figure 1. We emphasize that the facial points are provided for the FBP-SCUT5500 database.

2) **Two Branches Architecture***:* We propose a two branch architecture that will be trained in end-to-end fashion.

Since FBP data is limited, we propose to exploit the low-level and high-level feature extraction capability of two powerful architectures simultaneously: ResneXt-50 and inception-v3. Fig. 2 summarizes our proposed 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 branches, we added the layer FC1 that maps the output of the ResneXt-50 branch (vector of dimension 2048) to 1024 neurons. Similarly, we added layer FC2, which maps the output of the Inception-v3 branch (vector of dimension 2048) to 1024 neurons. FC1 and FC2 were concatenated into a single vector which is mapped by FC3 layer that performs the regression. Note that the initial weights of both branches are the weights of the pre-trained ResneXt-50 and Inception-v3 models (trained on the ImageNet challenge database [24].), while the FC1, FC2 and FC3 layers are randomly initialized. Our proposed network with two branches is called REX-INCEP architecture. In the training phase, we will fine-tune this architecture for FBP.

3) **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 estimated values. The CNNs aim 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 used in our experiments. We emphasize that three of them are robust loss functions. We will also introduce a dynamic law that adjusts the parameters of the robust losses during training. The losses are computed for the batch size N, y_i denotes the ground truth score of the i^{th} image, and $\hat{y}_i$ denotes the estimated value corresponding to the i^{th} image.

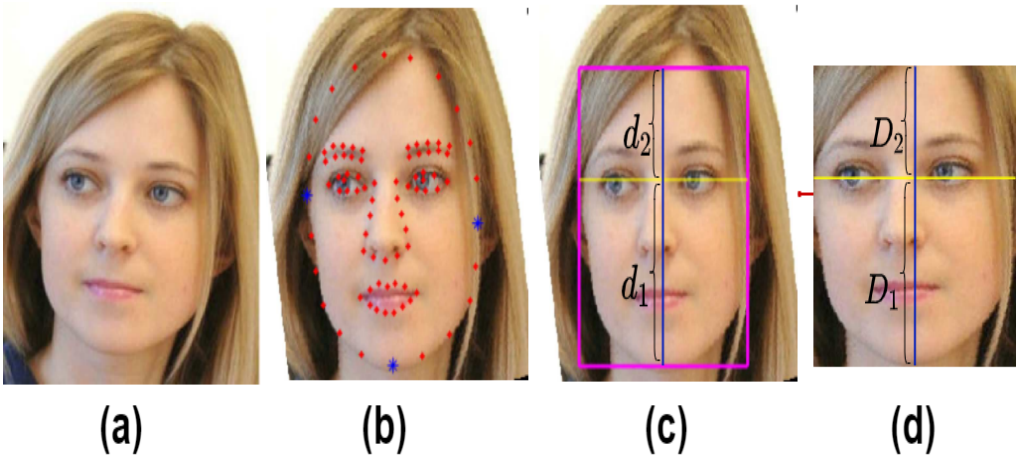

Figure 1: Face Region of Interest. (a) is an original image from the database SCUT-FBP5500 [11]. (b) is the rotated face with its 86 detected landmarks used to estimate the three face boundary lines (right, left, and bottom). These boundaries correspond to the three points $*$ marked in blue. (c) shows how the upper boundary of the face is determined. It is located at a distance $d_2 = 0.6\, d_1$ from the vertical position of the two eyes. (d) shows the cropped and rescaled face image with 224×224 pixels. Note that the distances D_1 and D_2 are constant for all cropped faces.

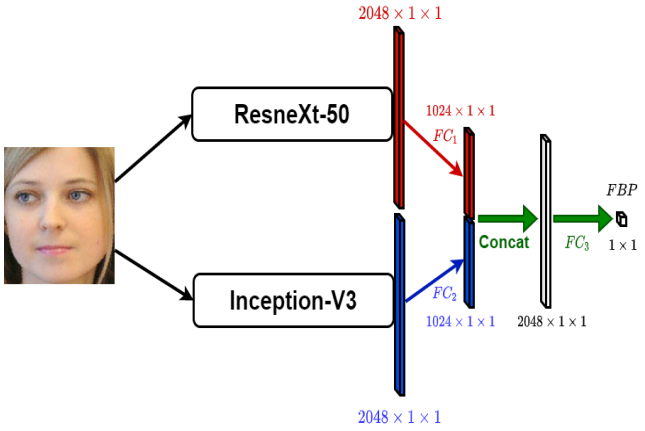

Figure 2: Our proposed two branches network REX-INCEP.

a) L_1 loss function: L_1 is one of the most commonly used loss functions. The most important property of the L_1 loss function is its robustness to outliers. For N batch size, L_1 loss function is defined by:

$$L_{L_1} = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}_i| \tag{1}$$

b) Mean Squared Error (MSE) loss function: MSE is also known as L_2 loss function, it is more sensitive to outliers compared to L_1. The MSE loss function should be used when the target data are normally distributed around a mean and when it is important to penalize outliers particularly heavily. For N predictions, the MSE loss function is defined by:

$$L_{MSE} = \frac{1}{N}\sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{2}$$

c) Dynamic Parameterized SmoothL1 (ParamSmoothL1) loss function: The loss function SmoothL1 creates a criterion that uses a quadratic term if the absolute element-wise error falls below 1, and an L_1 term otherwise. It is less sensitive to outliers than the MSE loss function, and in some cases prevents exploding gradients [28]. The SmoothL1 loss associated with N images is defined by:

$$L_{SmoothL1} = \frac{1}{N}\sum_{i=1}^{N} z_i \tag{3}$$

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

$$z_i = \begin{cases} 0.5\,(y_i - \hat{y}_i)^2, & if |y_i - \hat{y}_i| < 1 \\ |y_i - \hat{y}_i| - 0.5, & \text{otherwise} \end{cases} \tag{4}$$

Since the threshold may vary from one task to another, we proposed a Parameterized SmoothL1 loss function defined as follows:

$$L_{Para\_SmoothL1} = \frac{1}{N}\sum_{i=1}^{N} z_i \tag{5}$$

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

$$z_i = \begin{cases} 0.5\,(y_i - \hat{y}_i)^2, & if |y_i - \hat{y}_i| \le \alpha \\ |y_i - \hat{y}_i| + 0.5\,\alpha^2 - \alpha, & \text{otherwise} \end{cases} \tag{6}$$

where α is a tunable parameter. Figure 3 shows the proposed ParamSmoothL1 loss function with five α values (0.7, 0.6, 0.5, 0.4 and 0.3).

Our proposed dynamic robust loss functions are based on the following observation. During the training of ConvNets, the robust loss functions can be adjusted as the training progresses. Namely, during training, the model evolves and the outlier examples may vary. In the early stages of training,

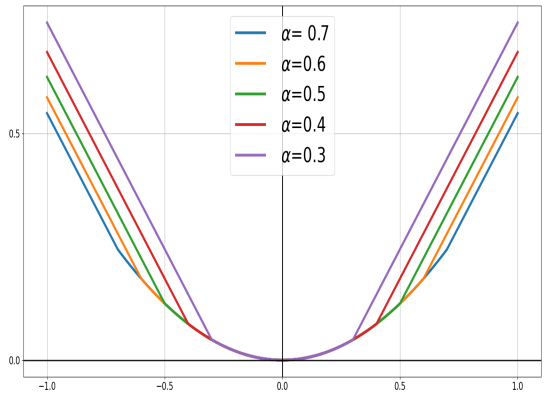

Figure 3: ParamSmoothL1 loss function with five α values (0.7, 0.6, 0.5, 0.4 and 0.3).

the model is usually neither very stable nor accurate enough to handle the outlier examples. Therefore, it is recommended to use the quadratic function of loss. At the end of the training, the model may be more or less accurate to deal with the outliers. Therefore, it is recommended to use the robust loss function where the range of non-outlier errors is relatively small. Concretely, this means that the parameter of the robust loss function, α, starts with a maximum value and decreases monotonically as the 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 make the robust loss function more adaptive to the training progress, we propose a dynamic parameter α. This parameter follows a cosine law as a function of the epoch number. The current value of α is given by:

$$\alpha_{cur} = \alpha_{min} + \frac{1}{2}(\alpha_{max} - \alpha_{min})\left(1 + \cos\left(\frac{e_{cur}}{n_e}\pi\right)\right) \quad (7)$$

where α_{cur} is the value of α at the current epoch (e_{cur}). The latter varies 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. Fig. 4 shows the values of α using the proposed law (Eq. (7)) as a function of epoch number. Here α_{max} and α_{min} are fixed at 0.7 and 0.3, respectively. Our dynamic law was inspired by the dynamic law used to control the learning rate in stochastic gradient descent methods [29].

IV. Performance Study

We used the five fold splits of SCUT-FBP5500 dataset [11] to evaluate the performance of our approach and compare with the state of the art methods.

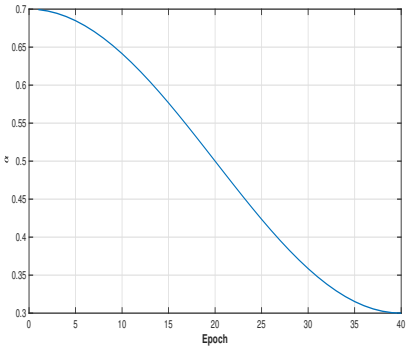

Figure 4: Dynamic ParamSmoothL1 with α that decreases from 0.7 to 0.3.

A. Database and evaluation protocols

To evaluate the performance of our approach, we used the database SCUT-FBP5500 [11]. It consists of 5500 frontal faces of subjects with different attributes: Age (from 15 to 60), gender (male/female), and ethnicity (Asian/Caucasian). Each face image was given a beauty score in the range [1-5] by 60 volunteers. In addition, each face image contains 86 facial features. The creators of the SCUT-FBP5500 database [11] provided 5 splits of the set allowing a five-fold cross-validation evaluation.

B. Evaluation Metrics

To evaluate the performance of each model, four evaluation metrics are used, namely: mean absolute error (MAE), root mean square error (RMSE), Pearson correlation coefficient (PC) [30] and the ϵ-error. We emphasize that the ϵ-error takes into account the incertitude of the ground-truth score using the standard deviation σ_i of the scores of all raters of image i as shown in equation (8):

$$\epsilon - error = \frac{1}{n}\sum_{i=1}^{n}\left(1 - exp\left(\frac{(y_i - \hat{y}_i)^2}{2\,\sigma_i^2}\right)\right) \quad (8)$$

Where $Y = (y_1, y_2, ..., y_n)$ is the ground-truth scores of the tested n images and $\hat{Y} = (\hat{y_1}, \hat{y_2}, ..., \hat{y_n})$ as their corresponding estimated scores. Where n represents the number of face images tested.

C. Experimental Setup

All experiments were performed on Pytorch [31] with an NVIDIA Geforce TITAN RTX 24 GB. All networks are trained for 40 epochs using Adam optimizer [32] and batch size of 15. The initial learning rate is 1e-4 for 20 epochs, then the learning rate decreases to 1e-5 for the next 10 epochs, and for the last 10 epochs the learning rate decreases to 1e-6. Active data augmentation is performed by rotating the input face by a samll random angle in the range [-5 deg, 5 deg]. For all experiments, the reported results correspond to the best PC of the test data during the training/testing of the 40 epochs.

D. Results

1) REX-INCEP architecture for FBP: In this section, we compare the performance of our proposed REX-INCEP with three baseline CNNs (Resnet-50 [33], Inception-v3 [23] and ResneXt-50 [21]) using the standard MSE loss function in their training. The results are summarized in Table I. From these results, we notice that our proposed REX-INCEP achieves better performance than the three CNN architectures.

As shown in Figure 2, our proposed REX-INCEP has two branches. In the first branch, our proposed REX-INCEP architecture is able to learn high-level features for FBP by using a combination of splitting, transformation and aggregation mechanisms through the ResneXt block. In the second branch, our proposed REX-INCEP architecture is able to learn high-level features for FBP by combining different convolutional layers with different kernel sizes and pooling layers through the Inception blocks. The main advantage of our REX-INCEP architecture is its ability to learn high-level FBP features using ResneXt and Inception blocks simultaneously, which proved its efficiency compared with other CNN architectures.

Table I: Comparison between three CNN architectures (Resnet-50, Inception-v3 and ResneXt-50) and our proposed REX-INCEP approach for Facial Beauty Prediction with MSE loss function.

CNN architecture	PC ↑	MAE ↓	RMSE ↓	ϵ-error ↓
Resnet-50 [33]	91.60	0.2097	0.2777	0.0796
Inception-v3 [23]	91.85	0.2089	0.2741	0.0787
ResneXt-50 [21]	91.85	0.2092	0.2751	0.0784
REX-INCEP (Our architecture)	92.08	0.2066	0.2714	0.0772

2) Dynamic Vs Fixed loss parameter: In this section, we compare the performance of Face Beauty Prediction dynamic and fixed loss functions. In this set of experiments, we use our proposed REX-INCEP architecture with the parametric robust SmoothL1 (ParamSmoothL1) loss function. We then compare the performance of two variants of the ParamSmoothL1 loss function: (i) parametric robust SmoothL1 with a fixed parameter α, and (ii) parametric robust SmoothL1 with a dynamic parameter α according to Eq. (7). To provide a fair comparison, the range of parameter variation associated with the dynamic scheme is also used by the fixed parameter loss function. This is achieved by repeating the training and testing with several fixed values in the same range.

Table II summarizes the obtained results using the five folds and their mean. For the loss of ParamSmoothL1, the interval of α is fixed to [0.7-0.3]. The fixed parameter scheme spans the following values $\{0.7, 0.6, 0.5, 0.4, 0.3\}$. Based on the results of the loss function ParamSmoothL1, we can see that the performance of the dynamic scheme is better than all performances obtained with the fixed parameters loss.

Table II: Five-fold cross-validation of facial beauty prediction using L1, MSE and dynamic smoothL1 loss.

Architecture	Fold	PC ↑	MAE ↓	RMSE ↓	ϵ-error ↓
REX-INCEP with ParamSmoothL1 α = 0.7	Fold 1	91.43	0.2081	0.2796	0.0800
	Fold 2	91.35	0.2098	.2806	0.0802
	Fold 3	92.20	0.2054	0.2731	0.0767
	Fold 4	92.54	0.2115	0.2714	0.0814
	Fold 5	92.05	0.2098	0.2716	0.0792
	Mean	**91.91**	**0.2089**	**0.2752**	**0.0795**
REX-INCEP with ParamSmoothL1 α = 0.6	Fold 1	91.60	0.2146	0.2790	0.080
	Fold 2	91.47	0.2173	0.2822	0.0842
	Fold 3	91.85	0.2196	0.2852	0.0875
	Fold 4	92.46	0.2152	0.2759	0.0839
	Fold 5	92.18	0.2040	0.2679	0.0753
	Mean	**91.91**	**0.2141**	**0.2780**	**0.0823**
REX-INCEP with ParamSmoothL1 α = 0.5	Fold 1	91.57	0.2141	0.2793	0.0807
	Fold 2	91.31	0.2104	0.2803	0.0810
	Fold 3	92.16	0.2132	0.2773	0.0808
	Fold 4	92.21	0.2178	0.2790	0.0851
	Fold 5	91.99	0.2062	0.2703	0.0761
	Mean	**91.85**	**0.2123**	**0.2772**	**0.0807**
REX-INCEP with ParamSmoothL1 α = 0.4	Fold 1	91.69	0.2119	0.2774	0.0803
	Fold 2	91.66	0.2101	0.2782	0.0804
	Fold 3	92.01	0.2148	0.2779	0.0811
	Fold 4	92.16	0.2158	0.2784	0.0854
	Fold 5	92.24	0.2092	0.2707	0.0802
	Mean	**91.95**	**0.2124**	**0.2765**	**0.0815**
REX-INCEP with ParamSmoothL1 α = 0.3	Fold 1	91.42	0.2137	0.2790	0.0800
	Fold 2	91.37	0.2122	0.2803	0.0818
	Fold 3	92.12	0.2081	0.2734	0.0776
	Fold 4	92.48	0.2038	0.2643	0.0758
	Fold 5	92.00	0.2091	0.2732	0.0801
	Mean	**91.88**	**0.2094**	**0.2741**	**0.07910**
REX-INCEP with dynamic ParamSmoothL1 loss function	Fold 1	92.02	0.2070	0.2718	0.0763
	Fold 2	91.67	0.2056	0.2766	0.0774
	Fold 3	92.20	0.2094	0.2733	0.0779
	Fold 4	92.65	0.2033	0.2634	0.0759
	Fold 5	92.38	0.2011	0.2639	0.0742
	Mean	**92.18**	**0.2052**	**0.2698**	**0.0763**

E. Dynamic ParamSmoothL1 vs Standard loss functions (L1, MSE)

In order to compare the performance of the proposed dynamic ParamSmoothL1 loss function with taht obtained bt the classic losees (L1 and MSE), we use the provided five folds to perform the cross-validation experiments. We train the proposed architecture REX-INCEP with each loss function: L1, MSE, and the proposed dynamic ParamSmoothL1 loss function.

Table III contains the results obtained with each fold, as well as the average over the five folds using two branches (REX-INCEP with L1, MSE, dynamic ParamSmoothL1 loss functions). It is worth noting that the presented result for each fold corresponds to the best result obtained by PC over the test data during the training of 40 epochs. The cross-validation results can provide a better comparison between the loss functions.

From Table III, we notice that the proposed dynamic ParamSmoothL1 loss function achieved better performance than L1 and MSE. This note is observed for both the 5-folds average results as well as for all folds results. This proves the efficiency of the proposed dynamic parametric loss function.

Table III: Five-fold cross-validation of facial beauty prediction using L1, MSE dynamic smoothl1 loss

Architecture	Fold	PC ↑	MAE ↓	RMSE ↓	ϵ-error ↓
REX-INCEP with L1 loss function	Fold 1	91.61	0.2115	0.2775	0.07888
	Fold 2	91.24	0.2122	0.2829	0.0813
	Fold 3	92.07	0.2095	0.2767	0.0789
	Fold 4	92.47	0.2052	0.2631	0.0750
	Fold 5	92.06	0.2051	0.2686	0.0761
	Mean	**91.89**	**0.2087**	**0.2737**	**0.0780**
REX-INCEP with MSE loss function	Fold 1	91.90	0.2081	0.2722	0.0772
	Fold 2	91.72	0.2068	0.2755	0.0783
	Fold 3	92.12	0.2085	0.2748	0.0783
	Fold 4	92.52	0.2045	0.2654	0.0767
	Fold 5	92.13	0.2049	0.2691	0.0756
	Mean	**92.08**	**0.2066**	**0.2714**	**0.0772**
REX-INCEP with dynamic ParamSmoothL1 loss function	Fold 1	92.02	0.2070	0.2718	0.0763
	Fold 2	91.67	0.2056	0.2766	0.0774
	Fold 3	92.20	0.2094	0.2733	0.0779
	Fold 4	92.65	0.2033	0.2634	0.0759
	Fold 5	92.38	0.2011	0.2639	0.0742
	Mean	**92.18**	**0.2052**	**0.2698**	**0.0763**

The above observations prove the effectiveness of the proposed fusion scheme. This also shows the efficiency of using two branch networks with different loss functions.

Table IV: Comparison with the Sate-of-the-Arts methods using the five-fold cross-validation scenario. $^+$ the authors of [9] used ResNeXt-50 as a backbone network to re-implement the [34] and [35] methods on the newly created SCUT - FBP5500 dataset. Dynamic ParamSmoothL* is our REX-INCEP network trained with the dynamic ParamSmoothL1 loss function.

PC ↑	1	2	3	4	5	Mean
Alexnet [11]	86.67	86.45	86.15	86.78	85.66	86.34
Resnet-18 [11]	88.47	87.92	89.29	89.32	90.04	89.00
ResneXt-50 [11]	89.85	89.32	90.16	89.90	90.64	89.97
CNN with SCA [18]	89.90	89.39	90.20	89.99	90.67	90.03
PI-CNN [35]$^+$	-	-	-	-	-	89.78
CNN + LDL [34]$^+$	-	-	-	-	-	90.31
ResNet-18 based AaNet [8]	-	-	-	-	-	90.55
ResneXt-50-R^{3}CNN [9]	91.43	90.66	91.36	91.46	9217	91.42
Dynamic ParamSmoothL1* (Ours)	**92.02**	**91.67**	**92.20**	**92.65**	**92.38**	**92.18**

MAE ↓	1	2	3	4	5	Mean
Alexnet [11]	0.2633	0.2605	0.2681	0.2609	0.2728	0.2651
Resnet-18 [11]	0.2480	0.2459	0.243	0.2383	0.2383	0.2419
ResneXt-50 [11]	0.2306	0.2285	0.226	0.2349	0.2258	0.2291
CNN with SCA [18]	0.2300	0.2284	0.2257	0.2345	0.2251	0.2287
PI-CNN [35]$^+$	-	-	-	-	-	0.2267
CNN + LDL [34]$^+$	-	-	-	-	-	0.2201
ResNet-18 based AaNet [8]	-	-	-	-	-	0.2236
ResneXt-50-R^{3}CNN [9]	0.2109	0.2152	0.2126	0.2130	0.2085	0.2120
Dynamic ParamSmoothL1* (Ours)	**0.2070**	**0.2056**	**0.2094**	**0.2033**	**0.2011**	**0.2052**

RMSE ↓	1	2	3	4	5	Mean
Alexnet [11]	0.3408	0.3449	0.3538	0.3438	0.3576	0.3481
Resnet-18 [11]	0.3258	0.3286	0.3184	0.3107	0.2994	0.3166
ResneXt-50 [11]	0.3025	0.3084	0.3016	0.3044	0.2918	0.3017
CNN with SCA [18]	0.3020	0.3081	0.3013	0.3039	0.2916	0.3014
PI-CNN [35]$^+$	-	-	-	-	-	0.3016
CNN + LDL [34]$^+$	-	-	-	-	-	0.2940
ResNet-18 based AaNet [8]	-	-	-	-	-	0.2954
ResneXt-50-R^{3}CNN [9]	0.2767	0.2895	0.2837	0.2804	0.2701	0.2800
Dynamic ParamSmoothL1* (Ours)	**0.2718**	**0.2766**	**0.2733**	**0.2634**	**0.2639**	**0.2698**

F. Comparison with State-of-the-Art methods

In this section, we compare our proposed methods with the state-of-the-art methods using five-fold cross-validation results. Table IV shows a comparison between our method and state-of-the-art methods using the five-fold cross-validation experiments and their average. Three evaluation metrics (PC, MAE and RMSE) are used for this comparison. The comparison shows that our approach performs better than the state-of-the-art methods, both in terms of average performance and performance of individual folds for all the evaluation metrics used. The proposed REX-INCEP with the dynamic loss function ParamSmoothL1 is shown to perform better than the state-of-the-art methods in all three evaluations metrics (PC, MAE and RMSE). This confirms that both the proposed two branches network and the dynamic loss functions play a crucial role in outperforming the state-of-the-art methods.

V. Conclusion

In this paper, we address the evaluation of the face beauty in facial images using Deep Learning. First, we propose a two-branch architecture (REX-INCEP) based on merging the architecture of two already trained networks. Second, we introduce Parametric SmoothL1 (ParamSmoothL1) loss function with a dynamic law to control the behavior of the robust regression during training and make he robust losses adaptive and dynamic.

Our proposed REX-INCEP solution is a two-branch CNN architecture that combines the ResneXt-50 and Inception-v3 architectures through FC layers. The main advantage of our REX-INCEP architecture is the ability to learn high-level FBP features simultaneously with ResneXt and Inception blocks.

In addition to using CNN architectures, several loss functions are used, namely L1, MSE, and the proposed dynamic ParamSmoothL1. For the dynamic loss functions (ParamSmoothL1), a cosine law is proposed to reduce the robust loss parameter during training. The dynamic schemes have been shown to be very efficient, both in terms of performance and in terms of avoiding the grid search for the best value, which incurs high computational costs. Our proposed approach outperformed many CNN baselines as well as many published state-of-the-art solutions. This superior performance was achieved in the evaluation protocol for the SCUT-FBP5500 dataset with five cross-validation using the three evaluation metrics (PC, MAE and RMSE).

References

[1] K. Dion, E. Berscheid, and E. Walster, "What is beautiful is good." *Journal of personality and social psychology*, vol. 24, no. 3, p. 285, 1972, publisher: American Psychological Association.

[2] T. Alashkar, S. Jiang, and Y. Fu, "Rule-based facial makeup recommendation system," in *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*. IEEE, 2017, pp. 325–330.

[3] A. Laurentini and A. Bottino, "Computer analysis of face beauty: A survey," *Computer Vision and Image Understanding*, vol. 125, pp. 184–199, 2014, publisher: Elsevier.

[4] L. Liang, L. Jin, and X. Li, "Facial skin beautification using adaptive region-aware masks," *IEEE transactions on cybernetics*, vol. 44, no. 12, pp. 2600–2612, 2014, publisher: IEEE.

[5] L. Xu, H. Fan, and J. Xiang, "Hierarchical Multi-Task Network For Race, Gender and Facial Attractiveness Recognition," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 3861–3865.

[6] F. Dornaika, K. Wang, I. Arganda-Carreras, A. Elorza, and A. Moujahid, "Toward graph-based semi-supervised face beauty prediction," *Expert Systems with Applications*, vol. 142, p. 112990, 2020, publisher: Elsevier.

[7] F. Dornaika, A. Moujahid, K. Wang, and X. Feng, "Efficient deep discriminant embedding: Application to face beauty prediction and classification," *Engineering Applications of Artificial Intelligence*, vol. 95, p. 103831, Oct. 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0952197620302013

[8] L. Lin, L. Liang, L. Jin, and W. Chen, "Attribute-Aware Convolutional Neural Networks for Facial Beauty Prediction." in *IJCAI*, 2019, pp. 847–853.

[9] L. Lin, L. Liang, and L. Jin, "Regression guided by relative ranking using convolutional neural network (R3CNN) for facial beauty prediction," *IEEE Transactions on Affective Computing*, 2019.

[10] L. Xu, J. Xiang, and X. Yuan, "Transferring rich deep features for facial beauty prediction," *arXiv preprint arXiv:1803.07253*, 2018.

[11] L. Liang, L. Lin, L. Jin, D. Xie, and M. Li, "SCUT-FBP5500: a diverse benchmark dataset for multi-paradigm facial beauty prediction," in *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 1598–1603.

[12] D. Gray, K. Yu, W. Xu, and Y. Gong, "Predicting facial beauty without landmarks," in *European Conference on Computer Vision*. Springer, 2010, pp. 434–447.

[13] D. Zhang, Q. Zhao, and F. Chen, "Quantitative analysis of human facial beauty using geometric features," *Pattern Recognition*, vol. 44, no. 4, pp. 940–950, 2011, publisher: Elsevier.

[14] P. Aarabi, D. Hughes, K. Mohajer, and M. Emami, "The automatic measurement of facial beauty," in *2001 IEEE International Conference on Systems, Man and Cybernetics. e-Systems and e-Man for Cybernetics in Cyberspace (Cat. No. 01CH37236)*, vol. 4. IEEE, 2001, pp. 2644–2647.

[15] H. Yan, "Cost-sensitive ordinal regression for fully automatic facial beauty assessment," *Neurocomputing*, vol. 129, pp. 334–342, 2014, publisher: Elsevier.

[16] W.-C. Chiang, H.-H. Lin, C.-S. Huang, L.-J. Lo, and S.-Y. Wan, "The cluster assessment of facial attractiveness using fuzzy neural network classifier based on 3D Moiré features," *Pattern Recognition*, vol. 47, no. 3, pp. 1249–1260, Mar. 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320313003798

[17] J. Fan, K. P. Chau, X. Wan, L. Zhai, and E. Lau, "Prediction of facial attractiveness from facial proportions," *Pattern Recognition*, vol. 45, no. 6, pp. 2326–2334, Jun. 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S003132031100478X

[18] K. Cao, K.-n. Choi, H. Jung, and L. Duan, "Deep Learning for Facial Beauty Prediction," *Information*, vol. 11, no. 8, p. 391, 2020, publisher: Multidisciplinary Digital Publishing Institute.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[21] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.

[22] D. Xie, L. Liang, L. Jin, J. Xu, and M. Li, "Scut-fbp: A benchmark dataset for facial beauty perception," in *2015 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, 2015, pp. 1821–1826.

[23] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[24] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, and M. Bernstein, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015, publisher: Springer.

[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[26] F. Bougourzi, K. Mokrani, Y. Ruichek, F. Dornaika, A. Ouafi, and A. Taleb-Ahmed, "Fusion of transformed shallow features for facial expression recognition," *IET Image Processing*, vol. 13, no. 9, pp. 1479–1489, Apr. 2019, publisher: IET Digital Library. [Online]. Available: https://digital-library.theiet.org/content/journals/10.1049/iet-ipr.2018.6235

[27] F. Bougourzi, F. Dornaika, K. Mokrani, A. Taleb-Ahmed, and Y. Ruichek, "Fusing Transformed Deep and Shallow features (FTDS) for image-based facial expression recognition," *Expert Systems with Applications*, vol. 156, p. 113459, Oct. 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417420302839

[28] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[29] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in *International Conference on Learning Representation*, 2017.

[30] K. Pearson, "VII. Note on regression and inheritance in the case of two parents," *proceedings of the royal society of London*, vol. 58, no. 347-352, pp. 240–242, 1895, publisher: The Royal Society London.

[31] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, and L. Antiga, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in neural information processing systems*, 2019, pp. 8026–8037.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[34] Y.-Y. Fan, S. Liu, B. Li, Z. Guo, A. Samal, J. Wan, and S. Z. Li, "Label distribution-based facial attractiveness computation by deep residual learning," *IEEE Transactions on Multimedia*, vol. 20, no. 8, pp. 2196–2208, 2017.

[35] J. Xu, L. Jin, L. Liang, Z. Feng, D. Xie, and H. Mao, "Facial attractiveness prediction using psychologically inspired convolutional neural network (PI-CNN)," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 1657–1661.