

RDCFace: Radial Distortion Correction for Face Recognition

He Zhao, Xianghua Ying*, Yongjie Shi, Xin Tong, Jingsi Wen, and Hongbin Zha
Key Laboratory of Machine Perception (MOE)
School of EECS, Peking University

{zhaoh97, shiyongjie, xintong, wenjingsi}@pku.edu.cn {xhying, zha}@cis.pku.edu.cn



Figure 1: Examples of Face images with and without radial lenses distortion. The faces are cropped from images under different levels of radial distortion (**Top**). These distorted face images are rectified by our method (**Bottom**), which greatly reduce the geometric deformation and calibrate the faces into standard view.

Abstract

The effects of radial lens distortion often appear in wide-angle cameras of surveillance and safeguard systems, which may severely degrade performances of previous face recognition algorithms. Traditional methods for radial lens distortion correction usually employ line features in scenarios that are not suitable for face images. In this paper, we propose a distortion-invariant face recognition system called RDCFace, which only utilize the distorted images of faces, to directly alleviate the effects of radial lens distortion. RDCFace is an end-to-end trainable cascade network, which can learn rectification and alignment parameters to achieve a better face recognition performance without requiring supervision of facial landmarks and distortion parameters. We design sequential spatial transformer layers to optimize the correction, alignment, and recognition modules jointly. The feasibility of our method comes from implicitly using the statistics of the layout of face features learned from the large-scale face data. Extensive experiments indicate that our method is robust to distortion and gains significant improvements on several benchmarks including LFW, YTF, CFP, and RadialFace, a real distorted face dataset compared with state-of-the-art methods.

1. Introduction

With the help of Deep Learning and Convolutional Neural Networks (CNNs), the performance of face recogni-

tion algorithms increased rapidly in the past few years [28, 21, 7]. Furthermore, the data-driven algorithm has successfully enhanced the robustness of learned facial features. In other words, a well designed CNN is capable of handling pose, occlusion and illumination variations of face images to a considerably high degree [42, 41, 19].

However, almost all existing face recognition algorithms are based on the assumption of the ideal pinhole camera model, whereas the ATM, monitoring system, and safeguard system usually use wide-angle cameras because of their large field of view (FOV). As illustrated in the Fig. 2, while the wide-angle lens provides broad scenes, face in the image is significantly distorted by radial lens distortion. The distorted face not only influences the way humans perceive one another, but also greatly degrade the performance of existing face recognition methods.

Traditional radial distortion correction algorithms tended to estimate the distortion parameters with calibration pattern [20, 39] or projections of lines detected in distorted images [29, 1]. However, the correction performance is limited by the requirement of a specific pattern and the accuracy of lines detecting results. In addition, face images are lack of line information which are not suitable for line-based methods. As far as we know, face recognition under radial distortion is required by practical application environments but is still a not well solved challenging problem.

In this paper, we propose RDCFace, a cascade network which maps a face image with radial distortion to its distortion-free counterpart to learn distortion-invariant

* Corresponding Author

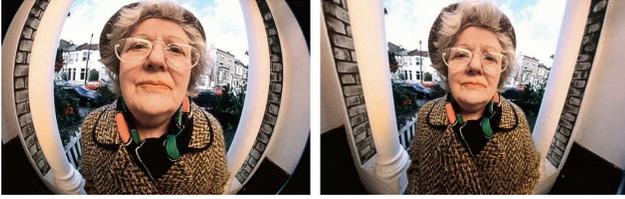


Figure 2. **Left:** An image of a woman standing in front of the access control system. It suffers severe radial distortion [source: Google Images]. The distortion will degrade the performance of previous face recognition systems. **Right:** Image corrected by the proposed method.

face representation. As shown in Fig. 3, RDCFace contains three modules: the distortion correction, face alignment and face recognition. The distortion correction module first estimates the distortion coefficient and rectifies the input face image. After that, the alignment module transforms the input face into a canonical view. Finally, recognition network takes the aligned image as input and learns discriminative representation. All the three networks are connected with several spatial transformer layers which make us be able to train the system in an end-to-end manner. To achieve a better rectification performance, an inverted foveal layer (IF), an edge enhancement layer and a re-correction loss are added to the correction module.

Training our proposed networks requires a large corpus of distorted face images with well-annotations of identity and distortion parameters. However, to the best of our knowledge, existing datasets are not suitable for this task. Therefore, we construct the first radial distorted face dataset by converting the IMDB-Face Dataset [30] to a distorted version. To numerically evaluate the effectiveness of our method on real distorted face images, we manually construct a real-world face dataset called RadialFace with fish-eye cameras.

The main contributions of our work can be summarized as follows:

1. For face images with radial lenses distortion, we observe that they can be directly applied to distortion correction, which is particularly beneficial for face recognition systems. The idea arises from the fact that human faces have strong statistics consistency in the layout of face features. To the best of our knowledge, this is the first attempt to formally study the feasibility of radial distortion correction from distorted face images.

2. We propose RDCFace, an end-to-end system which directly learns from distorted face images to improve face recognition performance under radial distortion by rectifying distorted images. We construct a synthetic distorted face dataset D-IMDB-Face and a real-world benchmark RadialFace to train and evaluate our network.

3. Experimental results show that the recognition performance under radial distortion of RDCFace outperforms

other state-of-the-arts on several public benchmarks including LFW, YTF, and CFP. Experiments on RadialFace and other collected real-world distorted images prove the generalization ability of our methods.

The rest of the paper is organized as follows: In Section 2, we briefly present several closely related existing works. After that, we describe the details of our network architecture and loss functions in Section 3. In Section 4, we introduce the acquisition of our training and test data. Experimental results are presented in Section 5. At the end, we conclude this paper in Section 6.

2. Related Work

Radial Distortion. Radial distortion correction has been widely studied in the past decades. Most of previous researchers took image un-distortion as the first step. Traditional methods required images taken with specific calibration pattern [20, 39] thus are lack of flexibility. Besides, since the distortion coefficient of zoom lenses is changing, the distortion correction should be dynamically performed. Swarninathan *et al.* [29] and Aleman-Flores *et al.* [1] calibrated images with the principle that a straight line should be projected into a straight line. However, their correction capacity are limited by the accuracy of lines detecting results. In addition, face images are lack of line information which are not suitable for these methods.

To achieve more robust and efficient performance, Rong *et al.* [25] first employed deep learning method for radial distortion correction. Recently, the FishEyeRecNet introduced scene parsing semantics into the rectification process and achieved better correction performance [37]. Xue *et al.* [35] proposed a line-guided parameters estimation module and a distorted line segments perception module to better utilize the line information. Unfortunately, all these methods mainly focus on specific scenes such as the building, street and indoor room, which are different from face images.

Meanwhile, some researchers tried to directly solve specific problems under radial distortion. Melo *et al.* [22] focus on radial distortion correction in medical endoscopy images. Fu *et al.* established FDDDB-360, a 360-degree fish-eye face detection dataset [12]. For pedestrian detection, Qian *et al.* created a warping pedestrian benchmark [24]. Deng *et al.* augmented fish-eye semantic segmentation dataset with three different focal lengths [8]. All these works can be summarized as a kind of data augmentation, which are not able to completely eliminate the effect of distortion. The comparison between data augmentation and our method are shown in Section 5.3.

Face Normalization and Recognition. Face recognition via deep learning has achieved a breakthrough in the past few years. The improvement can be generally attributed to the exploding size of dataset [31, 13, 23] and the evolution

of loss function [21, 32, 7, 33]. However, unconstrained photographs often include occlusions, non-frontal views, radial distortion, and even extreme poses, which introduce a myriad of challenges for face recognition. Most of prior works only focused on normalizing head pose and expression [42, 15, 6, 5]. Bas *et al.* [2] reconstruct 3D faces from 2D images under perspective distortion. Fried *et al.* [11] and Zhao *et al.* [40] rectify perspective distortion of near portraits which is complementary to our work. Different from them, our system is end-to-end trainable and could be optimized without paired images.

3. Approach

Unlike previous works, we combine the distortion correction and face recognition together and design an end-to-end distortion-invariant face recognition system. In this section, we first describe the general architecture of our system, and then discuss the details in each module.

Notation. Let I^d, I^c, I^a denote the distorted, corrected and aligned image, $f_{correct}, f_{align}, f_{rec}$, denote the correction, alignment, and recognition networks. The L_R, L_C, L_A represent the rectification, crop, and alignment layers. We define $I_i^* = [x_i^*, y_i^*]^T$ as the position of the i th point in the image coordinates. The coordinates are normalized to $[-1, 1]$.

As shown in Fig. 3, our RDCFace consists of three parts: distortion correction, alignment, and recognition modules. During the training process, we first randomly initialize the distortion coefficient and generate distorted image I^d . The face bounding box is detected by S3FD (Single shot scale-invariant face detector [38]) and sent to $f_{correct}$ with I^d . Then, the rectification layer L_R takes $\theta_{k_{pred}}$, the output of $f_{correct}$, to transform I^d into corrected image I^c based on the division distortion model [10]. After that, L_C crops the face into a 128×128 patch. Taking the cropped face as input, f_{align} generates the projective transformation parameters to help L_A align the patch into a frontal face I^a . Finally, the recognition network f_{rec} takes 112×112 aligned face I^a as input and produces a 512 dimensional feature vector. The ArcFace [7] loss is used for learning more discriminative features at the end. The whole cascaded network could be optimized with only identity information.

3.1. Correction and Alignment Network

We design two networks, correction and alignment networks to predict the distortion coefficient and projective transformation parameters separately. The correction network generates the single parameter k to rectify the radial distortion based on the inverse transformation of the division model [10]

$$r_d = \frac{1 - \sqrt{1 - 4kr_u^2}}{2kr_u}, \quad (1)$$

Table 1. Network details. ResBlock[N] denotes a Residual Block which consists of two groups of 3×3 Convolution, BatchNormalization and ReLU. N is the dimension of output feature maps. FullyConnected[N] is a fully-connected layer with N output neurons.

Network	Correction Network	Alignment Network
Input	$256 \times 256 \times 3$	$128 \times 128 \times 3$
Stage-1	Edge Enhancement IF Layer ResBlock[32]	ResBlock[16]
Stage-2	ResBlock[64] \times 2	ResBlock[32]
Stage-3	ResBlock[128] \times 2	ResBlock[32]
Stage-4	ResBlock[256] \times 2	ResBlock[64]
Stage-5	ResBlock[256] \times 2 Average Pool	ResBlock[64] Average Pool
Stage-6	FullyConnected[128] FullyConnected[1]	FullyConnected[8]

where r_u and r_d represents the Euclidean distance from an arbitrary pixel to the image center of original and distorted images. k represents the distortion coefficient. And the alignment network produces an eight parameters matrix P_θ which represents the projective transformation

$$I^a = P_\theta I^c = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \\ \theta_{31} & \theta_{32} & 1 \end{bmatrix} I^c. \quad (2)$$

We use an alignment network rather than face landmarks to align the input face images since face alignment algorithms is also strongly compromised by the radial distortion of facial features. Besides, as the position of landmarks is varying under different distortion degrees, it is hard to choose a proper target template as previous works [7, 32, 21].

The network architectures are shown in the Tab. 1. Our alignment network f_{align} is a simplified ResNet [16] which takes less additional parameters and computational cost compared with the recognition model. On the other side, learning the distortion parameter from a face image is much more difficult because of the lack of useful geometric information. Meanwhile, the network should not be too complex either since it is easy to cause over-fitting. To overcome those problems, we use an inverted foveal (IF) layer to re-weight the input image which lead to a faster and easier convergence. Moreover, we design an edge enhancement layer to provide more low-level information to the network. Besides the recognition loss function, we further introduce a re-correction loss to encourage the corrected face be in a standard view and help the network generalize robustly. The details of our distortion correction module are shown in Fig. 4 and we introduced them as follow.

Inverted Foveal Layer. One of the essential characteristics of radial distortion is that the degree of distortion increases rapidly with the radius r_d (see Fig. 5). It is natural to assign

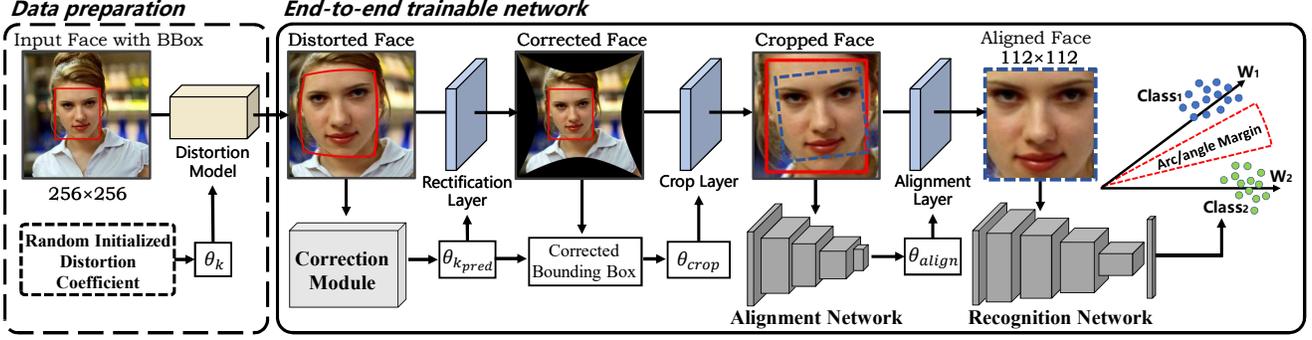


Figure 3. The overview of our RDCFace. The cascaded network sequentially rectifies distortion, aligns faces and extracts discriminative features. The three parts are connected with spatial transformer layers and can be jointly optimized by recognition loss.

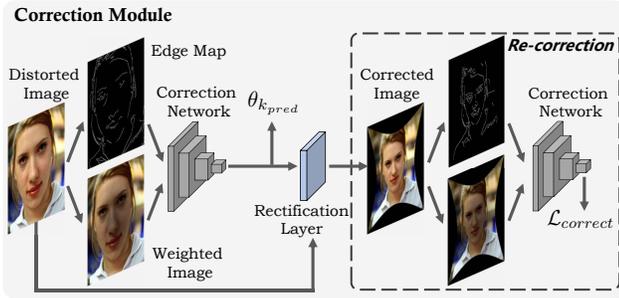


Figure 4. The details of our radial distortion correction module. Note that the edge map and weighted image generated by canny edge detector and inverted foveal layer are concatenated. The re-correction part could further improve the correction accuracy.

a higher weight to pixels far from the center. Besides, the radial distortion is strictly centrosymmetric. Therefore, we add a weight layer which is initialized by the inverted Gaussian kernel to re-weight the input image, which will help the correction network converge easier and faster. The inverted Gaussian kernel is defined as:

$$f(r) = \frac{1}{\sigma\sqrt{2\pi}} \left(1 - e^{-\frac{r^2}{2\sigma^2}}\right), \quad (3)$$

where r is the distance from an image point to the image center. The variance σ is empirically set to 64. Different from Shi *et al.* [27], we add this layer on the input image rather than the feature map of last convolutional layer.

Edge Enhancement Layer. As mentioned above, the low-level geometric information in the image plays a vital role in distortion correction. In order to provide more useful geometric information, we use Canny edge detector [4] to remove colour and increase local contrast. The extracted edge map is concatenated with the weighted image. The correction network takes this four-channel tensor as input to predict the distortion coefficient.

Re-correction Loss. Taking a distorted face image as input, a well-trained correction network should predict the distort-

ion parameter accurately to correct the image into a standard view. Moreover, if the input image is distortion-free, the correction module should not introduce unnecessary deformation. Therefore, we define the re-correction loss as

$$\begin{aligned} \mathcal{L}_{correct} &= E\|f_{correct}(I^c)\|_2^2 \\ &= E\|f_{correct}(L_R(f_{correct}(I^d), I^d))\|_2^2. \end{aligned} \quad (4)$$

The above equation means that, given a distortion image I^d , we first use the correction network $f_{correct}$ to predict its distortion parameter k . The rectification layer L_R then use parameter k to eliminate the distortion and generates corrected image I^c . After that, we send I^c into the correction network $f_{correct}$ again. This time, the output parameter is expected to be zero since the input I^c is expected to be corrected perfectly. The re-correction loss not only encourages the correction network to better eliminate the distortion but also suppress the excessive deformation on the distortion-free image. It increases the generalization ability of our system and leads to better recognition performance on both distorted and distortion-free images.

3.2. Face Recognition Network

Given the rectified and aligned face I^a , the recognition network f_{rec} extract the face representation $f_{rec}(I^a)$ as a 512 dimensional vector. We use the state-of-the-art network architecture, IR-SE50 [7] in all experiments. It obtains a balance between computational cost and accuracy. We use the same recognition loss as ArcFace:

$$\mathcal{L}_{rec} = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{s \cdot \cos(\theta_{y_i+m})}}{e^{s \cdot \cos(\theta_{y_i+m})} + \sum_{j=1, j \neq y_i}^n e^{s \cdot \cos(\theta_j)}} \quad (5)$$

where θ_{y_i} is the angle between the i -th feature $f_{rec}(I^a)$ and the classification weight of target class y_i . The θ_j denote the angle between the $f_{rec}(I^a)$ and the weight of other class. Hyper-parameter m and s control the margin penalty and the convergence difficulty. Here we set $m = 0.5$ and $s = 64$, the same setting reported in ArcFace [7].

In summary, we jointly optimize the three networks by minimizing recognition loss and re-correction loss:

$$\underset{\phi}{\operatorname{argmin}} \quad \mathcal{L}_{rec} + \lambda_c \mathcal{L}_{correct}, \quad (6)$$

where ϕ denotes the network parameters and λ_c is used for balancing the two loss functions.

3.3. Spatial Transformer Layers

To jointly optimize the correction, alignment and recognition modules with classification loss, our system should be end-to-end trainable. Inspired by STN [18], we design several spatial transformer layers to compute the coordinates of the corresponding pixels and generate the rectified image. The original DeepMind paper [18] only mentioned the linear transformation and merely elaborated the detailed implementation of the affine transformation. Our alignment and crop layers are similar to previous work, which indeed does projective and similarity transformations. Their detailed explanation and implementation can be found in recent works [18, 41]. However, the rectification layer, which performs a more complex nonlinear radial distortion correction is different from them. To fill this gap, we describe the details in the implementation of the rectification layer and analyze its forward and backward computations.

The rectification layer consists of a grid generator $\mathcal{T}_{correct}$ and a bilinear sampler. Taking distorted image as input, the correction network $f_{correct}$ outputs the distortion coefficient k :

$$k = f_{correct}(I^d). \quad (7)$$

To establish a reflection between the coordinates of distorted image I^d and corrected image I^c , we generate a sampling grid $G^c = \{G_i^c\}$ of pixel $G_i^c = (x_i^c, y_i^c)$. We assume that $\mathcal{T}_{correct}$ is a transformation from I^d to I^c . Based on the Eq. 1, we have

$$\begin{pmatrix} x_i^d \\ y_i^d \end{pmatrix} = \mathcal{T}_{correct}(G_i) = \frac{1 - \sqrt{1 - 4k(r_i^c)^2}}{2kr_i^c} \begin{pmatrix} x_i^c \\ y_i^c \end{pmatrix}. \quad (8)$$

We use a bilinear sampler to take the sampling grid $\mathcal{T}_{correct}(G_i)$ to extract the information from I^d and produce the sampled output image I^c . During backpropagation, the gradient should flow back to the distortion coefficient k and therefore back to the correction network $f_{correct}$. That can be written as :

$$\frac{\partial I_i^c}{\partial k} = \frac{\partial I_i^c}{\partial x_i^d} \cdot \frac{\partial x_i^d}{\partial k} + \frac{\partial I_i^c}{\partial y_i^d} \cdot \frac{\partial y_i^d}{\partial k}. \quad (9)$$

Based on Eq. 8, the partial derivative can be calculated as:

$$\begin{aligned} \frac{\partial x_i^d}{\partial k} &= \frac{\partial x_i^d}{\partial r_i^d} \cdot \frac{\partial r_i^d}{\partial k} \\ &= \frac{\partial x_i^d}{\partial r_i^d} \cdot \frac{1 - 2k(r_i^c)^2 - \sqrt{1 - 4k(r_i^c)^2}}{2k(r_i^c)^2 \sqrt{1 - 4k(r_i^c)^2}}, \end{aligned} \quad (10)$$

and the calculation of $\partial y_i^d / \partial k$ is similar.

With the help of the rectification layer, the correction network f_{pred} could be optimized by recognition loss to achieve a better rectification performance.

4. Data Preparation

4.1. Training Data Acquisition

Training the proposed system requires millions of images with well-annotated distortion parameters and identity information. However, to the best of our knowledge, there is no such large scale dataset that satisfy all requirements. Collecting millions of distorted face images will consume immense human resources. Meanwhile, it is impractical to take images with different kinds of cameras to cover the various degree of distortion. We therefore synthesize training data based on existing large scale face dataset with a proper radial distortion model.

There are many radial distortion models such as the polynomial model [14], the division model [10], and the field of view model [9]. Among all of them, the single parameter division model [10] strikes a balance between simplicity and generalization ability. It has been widely used in previous radial distortion methods [25, 1, 27]. Therefore, we use it to synthesize our training data. In the division model, the distortion degree is depend on the absolute value of distortion coefficient k . Fig. 5 shows the visual effect of different coefficient values. We choose coefficient from -3 to 0 to synthesis distorted data. This range is usually where common radial lens distortion coefficients vary in.

We choose IMDB-Face [30] as our training data due to its high signal-to-noise ratio. IMDB-Face consists of 1.7M images of 59K celebrities. We randomly choose 2K individuals as the validation set, 2K as the test set. The remaining data are used to train the model. During the training process, the images are distorted dynamically with randomly selected distortion parameters. While for the validation and test sets, we fix the distortion coefficient to ensure the fairness of evaluation between different methods.

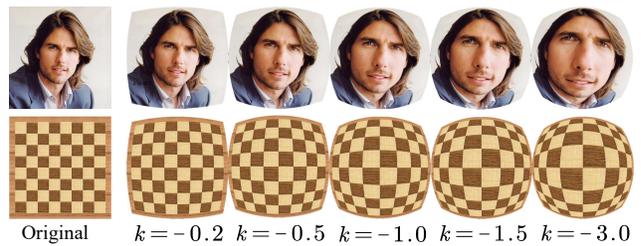


Figure 5. Distorted images generated by different coefficients k . The higher the absolute value of k is, the stronger radial distortion in the image will present.

4.2. Test Data Acquisition

To demonstrate that our system scales well to real-world distorted face images, we construct RadialFace, a dataset which contains 1564 images of 57 individuals captured by Canon EF 8-15mm f/4L fish-eye lens. The camera was set at a variable distance of 20 cm to 50 cm. Therefore the distortion degree is dynamically changing. Images in RadialFace cover a wide range of head poses, expressions and distortion degree. Different from Zhao [40], images in RadialFace are labeled with identity information but are not paired up with a distortion-free counterpart. We generate 600 positive (same identity) and 600 negative (different identity) pair and compare the verification accuracy of RDCFace with state-of-the-art distortion correction and face recognition methods.

5. Experiments

Our experiments are designed as follows: Firstly, we conduct ablation studies on the correction network. Secondly, we conduct quantitative analysis on the test set of IMDB-Face [30] to demonstrate the effectiveness of RDCFace. Thirdly, we compare the performance of our method with other state-of-the-arts on three commonly used public benchmarks. Finally, we evaluate our method on RadialFace and other collected real-world distorted images to show the generalization abilities of our system both quantitatively and qualitatively.

5.1. Experiment Settings

We train our network on IMDB-Face for 20 epochs, and the parameters are optimized by using stochastic gradient descent (SGD). The learning rates of correction, alignment and recognition networks are set to $1e-3$, $5e-4$, and $1e-1$, respectively. Moreover, they are shrunk by a factor of 10 once the validation error stops decreasing. The λ_c is set to 0.1. Those hyper-parameters are determined by the validation set. Also, we set momentum to 0.9 and weight decay to $5e-4$, which are same as ArcFace [7].

5.2. Ablation Study on Correction Network

To demonstrate the effectiveness of the proposed correction module, we compare the fitting error of the distortion coefficient on the test set of IMDB-Face with different settings. Here we use the Mean Absolute Error (MAE) between the prediction coefficient and the ground truth to measure the fitting error. In Tab. 2, the baseline \mathcal{L}_{rec} represent that the correction network is a purity ResNet which is supervised with only recognition loss. We observe that all of the three modules introduced in Section 3.1 can improve the correction performance. Moreover, by jointly using these modules, we further reduce the fitting error from 0.283 to 0.215 with no parameters increasing.

Table 2. Ablation study of correction network architectures on fitting error (MAE) of IMDB-Face test set.

Architecture	Fitting Error
\mathcal{L}_{rec}	0.283
\mathcal{L}_{rec} with IF	0.241
\mathcal{L}_{rec} with Edge	0.265
\mathcal{L}_{rec} with $\mathcal{L}_{correct}$	0.227
\mathcal{L}_{rec} with IF, Edge, $\mathcal{L}_{correct}$	0.215

5.3. Quantitative Analysis on IMDB-Face Test Set

Experiment Setting. To demonstrate the contribution of each component of our system, we conduct several experiments with different settings as follows:

(1) **Baseline:** The baseline model is trained on original IMDB-Face, which aligned by 2D-FAN [3] with similarity transform. (2) **Align:** Radial distortion deforms the spatial structure of facial features and degrades the performance of landmark detection algorithms. Therefore, we use an alignment network to accomplish face alignment. The training data here is original IMDB-Face with bounding box detected by S3FD [38]. (3) **Aug:** To narrow the domain gap between original and distorted images, it is a naive and intuitive way to directly use generated distorted data as a kind of data augmentation. We train the Baseline and Align model on distorted data without correction module and compare the performance with RDCFace. (4) **RDCFace:** RDCFace is our proposed method, which sequentially applies the correction, alignment, and recognition modules.

Results. Tab. 3 shows the quantitative results under 1:N identification protocol on the test set of IMDB-Face. To measure the effects of different degrees of distortion, we distorted the test set with various coefficients and compare the True Acceptance Rate (TAR) with False Acceptance Rate (FAR) at 10^{-2} of RDCFace with other methods.

As shown in the Tab. 3, the Baseline model performs well on the original data. However, the TAR@FAR=1e-2 significantly drops from 83.44% to 18.42% while the degree of distortion is increasing. The sharp decline indicates that radial distortion would severely degrade the performance of standard FR system. Align model slightly improves the performance which proves that the alignment network is more suitable for distorted data compared with landmark-based method. The Aug operation narrows the gap of the performance between original and distorted data from 37.76% to 2.55%. However, it also increases the difficulty of training and reduces the convergence rate, which leads to a worse performance on the original data. Different from them, RDCFace obtains good results on both distorted and original data. It outperforms the baseline by a large margin, improve the TAR from 45.68% to 83.03% on randomly distorted data. And the gap between randomly distorted and original data is reduced to only 0.35%. The

Table 3. Experiment results on the IMDB-Face test set. We compare our method RDCFace against other approaches and report 1:N identification accuracy (TAR@FAR=1e-2, %) on the IMDB-Face test set. RDCFace steadily improves the performance compared with Baseline and Align while the degree of distortion is increasing. Aug operation bridges the gap between original and distorted data to some extent. However, it also increases the difficulty which lead to a worse performance than our RDCFace.

Method	Original	$k = -0.5$	$k = -1.0$	$k = -1.5$	$k = -2.0$	$k = -3.0$	Random k
Baseline	83.44	68.79	47.98	42.56	28.05	18.42	45.68 (-37.76)
Align	83.62	69.82	51.43	49.10	35.91	20.63	49.15 (-34.47)
Baseline+Aug	80.70	77.75	78.21	76.54	74.61	73.82	77.27 (-3.43)
Align+Aug	81.37	79.27	78.52	78.64	77.38	76.77	78.82 (-2.55)
RDCFace	83.38	83.12	83.25	83.08	82.87	82.68	83.03 (-0.35)

Table 4. Verification accuracy (%) of different methods on LFW, CFP-FP and YTF and their distorted version. RDCFace gets the highest accuracy on distorted data.

Method	#Imgs	LFW	D-LFW	CFP-FP	D-CFP-FP	YTF	D-YTF
<i>Our Approach:</i>							
Baseline	1.7M	99.78	98.27 (-1.51)	97.86	63.30 (-34.56)	97.18	84.68 (-12.5)
Align	1.7M	99.80	97.67 (-2.13)	96.53	68.28 (-28.25)	97.25	85.03 (-12.22)
Baseline+Aug	1.7M	99.23	98.55 (-0.68)	95.43	93.20 (-2.23)	95.53	93.26 (-2.27)
Align+Aug	1.7M	99.43	98.83 (-0.60)	95.32	93.64 (-1.68)	95.46	93.84 (-1.62)
RDCFace	1.7M	99.80	99.78 (-0.02)	96.62	95.30 (-1.32)	97.10	96.98 (-0.12)
<i>Previous FR Methods:</i>							
ArcFace, R50 [7]	5.8M	99.82	98.37 (-1.45)	98.03	67.53 (-30.50)	97.40	84.70 (-12.70)
SphereFace [21]	0.5M	99.40	96.24 (-3.16)	94.28	58.92 (-35.36)	94.96	78.22 (-16.74)
<i>Previous RDC Methods:</i>							
Rong [25]	1.7M	99.72	99.20 (-0.52)	95.76	88.16 (-7.60)	96.70	94.36 (-2.34)
Alemán-Flore [1]	1.7M	99.76	98.31 (-1.45)	96.68	64.92 (-32.76)	96.98	85.20 (-11.78)

results indicate that the proposed RDCFace is a practical end-to-end framework for face recognition under various degrees of radial distortion.

5.4. Evaluation on Public Benchmarks

Experiment Setting. To ensure that our methods have cross-data generalization ability, we compare the performance of RDCFace with other state-of-the-arts of face recognition (FR) and radial distortion correction (RDC) on distorted public benchmarks.

(1) **Previous FR methods:** We re-implement the ArcFace [7] and SphereFace [21] on MS1MV2 [7] and CASIA-WebFace [36] dataset. The result of our re-implementation is comparable to the reported on original benchmarks, and we evaluate their performance under radial distortion. (2) **Previous RDC Methods:** We first use previous RDC methods to rectify the distorted benchmarks and then use our Baseline model to extract deep features. We assume that the recognition accuracy is positively correlated with the correction performance. Since Rong [25] focused on distortion correction of human-made scenes images, we retrain their model on face images. On the other hand, the methods proposed by Alemán-Flores [1] is not training-based. Therefore, we directly apply their algorithm on our data.

Test Dataset. We conduct experiments on three commonly used benchmarks LFW [17], CFP [26], and YTF [34]. LFW and YTF datasets are the most widely used benchmark for unconstrained face verification on images and videos. Here we follow the *unrestricted with labelled outside data protocol* to report the performance. CFP is a challenging dataset contains 500 celebrities, each of which has ten frontal and four profile face images. We focus on the Frontal-Profile (CFP-FP) setting and follow the standard 10-fold protocol.

Based on the division distortion model[10], we generate distorted LFW, CFP, and YTF (D-LFW, D-CFP, D-YTF). For all of those datasets, we employ the S3FD [38] to detect the face bounding box. For those methods that need pre-alignment, we use the 2D-FAN [3] to detect face landmarks and align the face images by similarity transformation.

Results. Tab. 4 shows the verification performance. On the original data, the performance of our Baseline is comparable to other state-of-the-arts. Both the Baseline and the state-of-the-art FR methods achieve high performance on the distortion-free data. However, public methods are severely degraded by the radial distortion. Our RDCFace consistently improves the performance up from 98.27% to 99.78% (D-LFW), 63.30% to 95.30% (D-CFPFP) and 84.68% to 96.98% (D-YTF) on distorted data compared

with Baseline. The performance gap between distorted and original images is scaled down more than 10 times. It also outperforms the ArcFace and SphereFace with a considerable gap. Although the previous RDC methods could alleviate the distortion to some extent, they still face two problems. Firstly, since the face images are lack of useful geometric information, previous RDC methods cannot eliminate the distortion without the supervision of recognition accuracy. Secondly, they cannot avoid influencing the distortion-free data, so the performance on the original benchmarks drops a little. On the contrary, with the help of the spatial transformer layers, RDCFace could directly use the recognition loss to jointly optimize the correction and recognition modules. In addition, the re-correction loss ensures the distortion-free data remain unchanged. In conclusion, the distortion correction and face recognition module are complementary which help RDCFace outperforms other RDC methods by a large margin on both standard and distorted datasets.

5.5. Evaluation on RadialFace

To verify the robustness of RDCFace on real-world distorted data. We evaluate our method on RadialFace dataset. The results illustrated in Fig. 6 indicate two points. First, as column (a) and (d) show, our synthesized data are similar to the real-world distorted images which ensure the generalization ability of our system. Second, our method could moves a wide range of radial distortion artifacts such as the increased nose size. That explain why RDCFace could outperform other state-of-the-arts on several benchmarks.

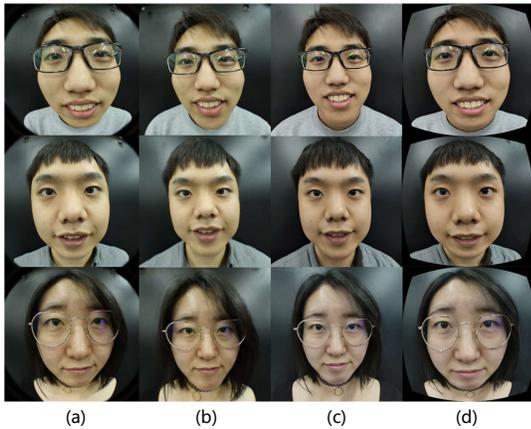


Figure 6. (a). Input images of RadialFace; (b). Images corrected by RDCFace; (c). Images taken by distortion-free lens; (d). Distorted images synthesized by our method.

We generate 600 positive and 600 negative pairs from RadialFace dataset and compare the mean verification accuracy under 10-folders cross-validation. The receiver operating characteristic (ROC) curves plotted in Fig. 7 show that RDCFace reaches the highest accuracy as expected.

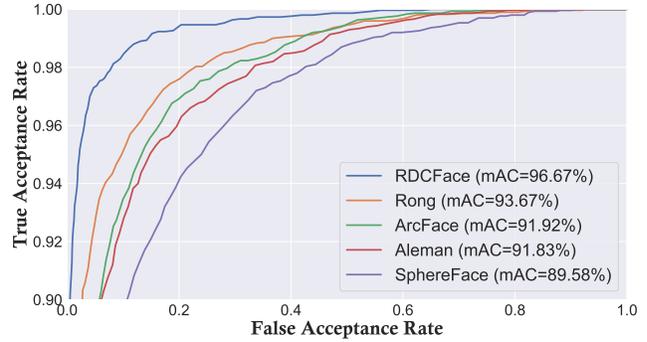


Figure 7. ROC curves of different methods on RadialFace dataset.

5.6. Qualitative Analysis on Real Images

To further evaluate the generalization ability of RDCFace, we conduct experiment on real-world distorted images collected from the Internet. These images contain a large variety of lightness, face pose, and expressions.

As shown in Fig. 8, traditional method [1] which depends on the line information cannot satisfy the needs of correcting face images. Besides, without the supervision of recognition loss, learning-based method [25] also has the problems of over-rectified and under-rectified. On the other hand, faces corrected by our method look natural and real, which potentially lead to higher recognition accuracy.



Figure 8. Qualitative analysis on real-world images. From **Left to Right**: real image, image corrected by Aleman [1], Rong [25], RDCFace, and repeated this order.

6. Conclusion

In contrast to recent face recognition systems which usually require calibrating radial lens distortion in advance, we have presented RDCFace, the first method that can make such calibration online, directly using the distorted face images. RDCFace alleviates the distortion to improve the recognition performance and allows entire end-to-end training. With the help of spatial transformer layers, we optimize the cascaded network by only recognition loss. We construct RadialFace, the first real-world face dataset under radial distortion to evaluate our method. Empirical results show that RDCFace outperforms state-of-the-art face recognition and radial distortion correction methods.

Acknowledgement. This work was supported in part by State Key Development Program Grand No. 2016YFB1001001, and NNSFC Grant No. 61971008.

References

- [1] Miguel Alemán-Flores, Luis Alvarez, Luis Gomez, and Daniel Santana-Cedrés. Automatic lens distortion correction using one-parameter division models. *IPOLE*, 4:327–343, 2014.
- [2] Anil Bas and William AP Smith. What does 2d geometric information really tell us about 3d face shape? *IJCV*, 127(10):1455–1473, 2019.
- [3] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks). In *ICCV*, pages 1021–1030, 2017.
- [4] John Canny. A computational approach to edge detection. *TPAMI*, (6):679–698, 1986.
- [5] Feng-Ju Chang, Anh Tuan Tran, Tal Hassner, Iacopo Masi, Ram Nevatia, and Gerard Medioni. Faceposenet: Making a case for landmark-free face alignment. In *ICCVW*, pages 1599–1608, 2017.
- [6] Forrester Cole, David Belanger, Dilip Krishnan, Aaron Sarna, Inbar Mosseri, and William T Freeman. Synthesizing normalized faces from facial identity features. In *CVPR*, pages 3703–3712, 2017.
- [7] Jiansheng Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, pages 4690–4699, 2019.
- [8] Liuyuan Deng, Ming Yang, Yeqiang Qian, Chunxiang Wang, and Bing Wang. Cnn based semantic segmentation for urban traffic scenes using fisheye camera. In *IV*, pages 231–236. IEEE, 2017.
- [9] Frédéric Devernay and Olivier Faugeras. Straight lines have to be straight. *Machine Vision & Applications*, 13(1):14–24, 2001.
- [10] Andrew W Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. In *CVPR*, volume 1, pages 1–1. IEEE, 2001.
- [11] Ohad Fried, Eli Shechtman, Dan B Goldman, and Adam Finkelstein. Perspective-aware manipulation of portrait photos. *ACM TOG*, 35(4):1–10, 2016.
- [12] Jianglin Fu, Saeed Ranjbar Alvar, Ivan Bajic, and Rodney Vaughan. Fddb-360: Face detection in 360-degree fisheye images. In *MIPR*, pages 15–19. IEEE, 2019.
- [13] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *ECCV*, pages 87–102. Springer, 2016.
- [14] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [15] Tal Hassner, Shai Harel, Eran Paz, and Roei Enbar. Effective face frontalization in unconstrained images. In *CVPR*, pages 4295–4304, 2015.
- [16] Kaifeng He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CVPR*, pages 770–778, 2016.
- [17] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.
- [18] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *NIPS*, pages 2017–2025, 2015.
- [19] Hongjun Jia and Aleix M Martinez. Face recognition with occlusions in the training and testing sets. In *FG*, pages 1–6. IEEE, 2008.
- [20] Seok-Han Lee, Sang-Keun Lee, and Jong-Soo Choi. Correction of radial distortion using a planar checkerboard pattern and its image. *IEEE Transactions on Consumer Electronics*, 55(1):27–33, 2009.
- [21] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphreface: Deep hypersphere embedding for face recognition. In *CVPR*, pages 212–220, 2017.
- [22] Rui Melo, Joao P Barreto, and Gabriel Falcao. A new solution for camera calibration and real-time image distortion correction in medical endoscopy—initial technical evaluation. *ITBE*, 59(3):634–644, 2011.
- [23] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, et al. Deep face recognition. In *BMVC*, volume 1, page 6, 2015.
- [24] Yeqiang Qian, Ming Yang, Chunxiang Wang, and Bing Wang. Self-adapting part-based pedestrian detection using a fish-eye camera. In *IV*, pages 33–38. IEEE, 2017.
- [25] Jiangpeng Rong, Shiyao Huang, Zeyu Shang, and Xianghua Ying. Radial lens distortion correction using convolutional neural networks trained with synthesized images. In *ACCV*, pages 35–49. Springer, 2016.
- [26] Soumyadip Sengupta, Jun-Cheng Chen, Carlos Castillo, Vishal M Patel, Rama Chellappa, and David W Jacobs. Frontal to profile face verification in the wild. In *WACV*, pages 1–9. IEEE, 2016.
- [27] Yongjie Shi, Danfeng Zhang, Jingsi Wen, Xin Tong, Xianguhua Ying, and Hongbin Zha. Radial lens distortion correction by adding a weight layer with inverted foveal models to convolutional neural networks. In *ICPR*, pages 1–6. IEEE, 2018.
- [28] Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation by joint identification-verification. In *NIPS*, pages 1988–1996, 2014.
- [29] R Swarninathan and Shree K Nayar. Non-metric calibration of wide-angle lenses and polycameras. In *CVPR*, volume 2, pages 413–419. IEEE, 1999.
- [30] Fei Wang, Liren Chen, Cheng Li, Shiyao Huang, Yanjie Chen, Chen Qian, and Chen Change Loy. The devil of face recognition is in the noise. In *ECCV*, pages 765–780, 2018.
- [31] Fei Wang, Liren Chen, Cheng Li, Shiyao Huang, Yanjie Chen, Chen Qian, and Chen Change Loy. The devil of face recognition is in the noise. In *ECCV*, pages 765–780, 2018.
- [32] Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. Normface: 12 hypersphere embedding for face verification. In *ACM MM*, pages 1041–1049. ACM, 2017.
- [33] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, pages 499–515. Springer, 2016.

- [34] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *ECCV*, 2011.
- [35] Zhucun Xue, Nan Xue, Gui-Song Xia, and Weiming Shen. Learning to calibrate straight lines for fisheye image rectification. In *CVPR*, June 2019.
- [36] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014.
- [37] Xiaoqing Yin, Xinchao Wang, Jun Yu, Maojun Zhang, Pascal Fua, and Dacheng Tao. Fisheyerecnet: A multi-context collaborative deep network for fisheye image rectification. In *ECCV*, pages 469–484, 2018.
- [38] Shifeng Zhang, Xiangyu Zhu, Zhen Lei, Hailin Shi, Xiaobo Wang, and Stan Z Li. S3fd: Single shot scale-invariant face detector. In *ICCV*, pages 192–201, 2017.
- [39] Zhengyou Zhang. A flexible new technique for camera calibration. *PAMI*, 22, 2000.
- [40] Yajie Zhao, Zeng Huang, Tianye Li, Weikai Chen, Chloe LeGendre, Xinglei Ren, Ari Shapiro, and Hao Li. Learning perspective undistortion of portraits. In *ICCV*, October 2019.
- [41] Yuanyi Zhong, Jiansheng Chen, and Bo Huang. Toward end-to-end face recognition through alignment learning. *IEEE signal processing letters*, 24(8):1213–1217, 2017.
- [42] Erjin Zhou, Zhimin Cao, and Jian Sun. Gridface: Face rectification via learning local homography transformations. In *ECCV*, pages 3–19, 2018.