Led3D: A Lightweight and Efficient Deep Approach to Recognizing Low-quality 3D Faces

Guodong Mu[1], Di Huang[1*], Guosheng Hu[2], Jia Sun[1], and Yunhong Wang[1]
[1]Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, China
[2]Anyvision, Queens Road, Belfast, UK
{muyouhang,dhuang,sunjia,yhwang}@buaa.edu.cn,huguosheng100@gmail.com

Abstract

Due to the intrinsic invariance to pose and illumination changes, 3D Face Recognition (FR) has a promising potential in the real world. 3D FR using high-quality faces, which are of high resolutions and with smooth surfaces, have been widely studied. However, research on that with low-quality input is limited, although it involves more applications. In this paper, we focus on 3D FR using low-quality data, targeting an efficient and accurate deep learning solution. To achieve this, we work on two aspects: (1) designing a lightweight yet powerful CNN; (2) generating finer and bigger training data. For (1), we propose a Multi-Scale Feature Fusion (MSFF) module and a Spatial Attention Vectorization (SAV) module to build a compact and discriminative CNN. For (2), we propose a data processing system including point-cloud recovery, surface refinement, and data augmentation (with newly proposed shape jittering and shape scaling). We conduct extensive experiments on Lock3DFace and achieve state-of-the-art results, outperforming many heavy CNNs such as VGG-16 and ResNet-34. In addition, our model can operate at a very high speed (136 fps) on Jetson TX2, and the promising accuracy and efficiency reached show its great applicability on edge/mobile devices.

1. Introduction

Face recognition (FR) is a very hot topic in the computer vision community. Recently, 2D FR has achieved great success with the development of deep learning techniques and the availability of big visual data. For example, the well known FaceNet [30], which is built based on the Inception architecture and the triplet loss, makes use of 200M faces of 8M identities for training and reports a 99.63% accuracy on the LFW [14] benchmark, surpassing human-level performance. Despite 2D FR has been widely applied to many specific situations in the real world, its robustness to more challenging cases, *e.g.* large head poses and extreme lighting conditions, remains problematic.

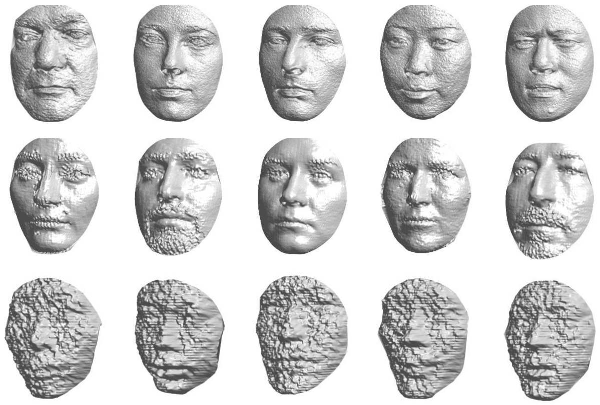


Figure 1. Visualization of 3D faces. The FRGC v2 (first row) and Bosphorus (second row) databases provide high-quality data while the Lock3DFace (last row) dataset offers low-quality ones, with averagely 53K, 27K and 9K points in each model respectively.

Unlike 2D face images, 3D face models deliver shape information, which is intrinsically invariant to pose and illumination[1] changes. During the last two decades, a large number of 3D solutions have been proposed, and the accuracies on public benchmarks, *e.g.* FRGC v2 [26], Bosphorus [29], and BU-3DFE [36] have been consistently promoted. More importantly, they have demonstrated the potential to handle the issues unsolved in the 2D domain. For instance, high scores are reached on the samples with serious data missing due to self-occlusions incurred by large poses (*i.e.* $\geq 45°$) [22, 6, 11], and a more recent work [40] presents a deep model based approach, which boosts the state of the art precisions on many major databases close to full marks. However, the overwhelming majority of current

* Corresponding author.

[1] Some 3D imaging devices partially depend on RGB cameras and the data are not really invariant to illumination changes.

3D FR studies focus on high-quality data acquired through very expensive 3D scanners, and their systems are generally sophisticated with relatively high computational cost, which limits 3D FR in practical applications.

The advent of consumer RGB-D cameras, such as Microsoft Kinect and Intel RealSense, makes it possible to obtain depth data at an affordable price. Although such data can be efficiently captured and processed, they are of really low-quality (see Figure 1), and early papers thus use them only for coarse-grained classification tasks, including gesture recognition [34] and gender recognition [16]. There indeed exist some attempts on FR [7, 1, 24, 12], but the subjects involved are quite limited. Zhang *et al.* [37] release the first comprehensive dataset that is suitable for evaluating methods on 3D FR using low-quality depth images, namely Lock3DFace, and they provide baseline results on it using Iterative Closet Points (ICP). Later, Cui *et al.* [2] present a deep model based baseline. They both illustrate the feasibility of identification on low-quality 3D face data. Moreover, 3D data are reputed to be more tolerant to photo and video based face spoofing than 2D images and low-quality 3D data also suggest competent at anti-spoofing such as facial mask attacking [4, 33], featuring another advantage in FR, as this reliability is crucial in some scenarios with strong security requirement, *e.g*. bank related applications. On the other side, FaceID, a software for user cooperative unlocking provided by iPhone, is generally acknowledged, which reveals a good commercial perspective of 3D FR with low-quality input. Unfortunately, very little research has investigated this issue. This work bridges this gap.

In this work, we propose a novel deep approach, namely Led3D, to 3D FR using low-quality depth images, targeting both higher accuracy and higher efficiency. To achieve this, we work on two ways, *i.e*. a new lightweight Convolutional Neural Network (CNN) architecture as well as bigger and finer training data.

New Lightweight Architecture. As we know, depth images by consumer 3D sensors are of low resolutions and with heavy noises as shown in Figure 1. In this case, a powerful model is needed to extract sufficiently discriminative features. To balance accuracy and efficiency, we focus on an enhanced lightweight network rather than stubbornly deepening the model. Our backbone network contains only 4 convolutional layers, and to make a high accuracy, we propose a Multi-Scale Feature Fusion (MSFF) module and a Spatial Attention Vectorization (SAV) module. The former combines features at different levels in an efficient way, improving the representation of low-quality face data, and the latter highlights important spatial facial clues when summarizing local features and outperforms the widely used Global Average Pooling (GAP) for FR.

Finer and Bigger Training Data. Deep models are data-hungry, and clean and well-organized training data are important to performance improvement. Due to the high cost of good-quality scanners for data collection, there is not any 3D face database as large-scale as the ones in 2D. To deal with the problem of inadequate data, FR3DNet [40] applies data augmentation to synthesize 3M faces of 100K identities from an ensemble of many public databases and a private one. In our case, we do not use extra data and propose a preprocessing pipeline and a data augmentation scheme for low-quality 3D face data, to generate a finer and bigger training set.

State-of-the-art Performance. With the new architecture and better data, we reach state-of-the-art performance on the Lock3DFace [37] and Bosphorus [29] databases. In addition, Led3D operates at a very high speed on an edge device, *i.e.* 136 fps on Jetson TX2, contributing a systematical solution for real-time 3D FR to the society.

2. Related Work

In this section, we briefly review the related work in the field of 2D FR, 3D FR, and lightweight CNNs.

2D FR. In the recent years, CNNs have been dominating this area by carefully designed network architectures and loss functions, with massive training data containing millions of faces and thousands of individuals. DeepFace [32], FaceNet [30], and VGGFace [25] are representatives. In spite of successive state of the art results, some of which are even better than that of human beings, recent investigations [38, 15, 13] point out that 2D FR is vulnerable to complex lighting changes and severe pose variations.

3D FR. In the last decade, 3D FR has been greatly developed along with the publicity of databases of high-quality 3D face models. It can be witnessed that the efforts made by hand-crafted methods are first on addressing expression changes [20, 3] and recently on dealing with poses and occlusions [3, 22], towards real-world scenarios.

In contrast to the case in the 2D domain, exploration on deep learning based 3D FR is not extensive. The reason mainly lies in the lack of big data, since public 3D databases are not so comprehensive. The largest one of high-quality data is ND-2006 [5], which only contains 13,450 scans of 888 subjects, much smaller than MS-Celeb-1M [8]. Data augmentation techniques are thus required. Kim *et al.* [21] integrate available benchmarks and increase samples by diversely generating expressions and poses and randomly cropping patches. With 10K augmented depth faces, they then fine-tune VGG-Face and reach the top accuracy on Bosphorus. Gilani *et al.* [6] further enhance data augmentation by adding a private dataset and synthesizing virtual IDs, and the deep model is trained from scratch and delivers very competitive scores on all the test sets.

Regarding 3D FR on low-quality data, research is limited. Preliminary attempts employ traditional methods, such as ICP, PCA, LBP, and HOG, and display some promising performance [1, 7, 24]. However, the databases used are small in terms of subject or image number and the variations involved are few. See [37] for a comparison of the low-quality 3D face databases.

To the best of our knowledge, Lock3DFace [37] is the first comprehensive low-quality 3D face benchmark, which contains 5,671 RGB-D videos of 509 individuals, collected by Kinect V2 in various conditions. The baseline results are given by ICP applied on high-quality scans reconstructed from depth videos. A recent work [2] exploits an existing deep model, namely Inception V2 [19], and provides another baseline on it. The two papers leave much room for improvement in both accuracy and efficiency.

Lightweight CNN. Many deep models show high accuracies in computer vision tasks. But the application to more scenarios, *e.g.* with Raspberry Pi, Jetson TX2, and mobile phones, is limited by the model size and the computation cost. Therefore, it is necessary to design lightweight architectures. SqueezeNet [18], ShuffleNet [39], and MobileNet [10] are famous examples. Although these solutions work well on image classification and object detection, they are not well investigated for FR, in particular for 3D FR using low-quality data.

3. An Efficient and Accurate Network

CNNs have been applied to 3D FR. However, they work on high-quality data. To our knowledge, very little research has investigated 3D FR with low-quality data, which actually has many applications. To bridge this gap, in this work, we propose a CNN based approach to improving the accuracy and efficiency.

For fast inference, the network has to be shallower, with a smaller number of parameters, leading to lower memory cost. Thus, our backbone network contains only 4 blocks which have 32, 64, 128, and 256 convolution filters respectively. Each block is composed of a convolution layer with a kernel size of 3×3, a batch normalization layer and a ReLU activation layer. As shown in Figure 2 (b), the blocks are very compact. To enhance the accuracy, we propose a Multi-Scale Feature Fusion (MSFF) module and a Spatial Attention Vectorization (SAV) module, detailed in Section 3.1 and 3.2 respectively. MSFF is used to fuse multi-scale features from each block for comprehensive representation and SAV emphasizes important spatial information, both of which improve the discriminative capacity of the resulting feature. We then apply a Dropout layer between SAV and the Fully-Connected (FC) layer, to overcome over-fitting. At the end of the network, we utilize a Softmax layer with the cross entropy loss to guide network training. The whole architecture is shown in Figure 2.

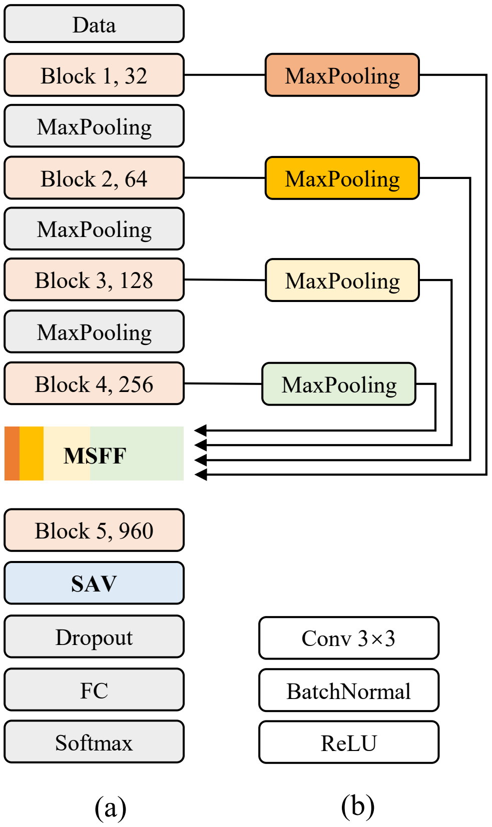


Figure 2. (a) The proposed architecture for 3D FR with low-quality data, including a Multi-Scale Feature Fusion (MSFF) module and a Spatial Attention Vectorization (SAV) module; and (b) details of the 'Block' used in (a).

3.1. Multi-Scale Feature Fusion

CNN has a hierarchical architecture which is a stack of multiple convolutional layers. Individual layers learn different information: the lower layers capture low-level elements such as basic colors, edges, while the higher ones encode abstract and semantic cues. It is natural to combine the features at different layers for better representation. For example, DenseNet [17] uses very dense connections to integrate the features. However, such connections are very heavy, leading to expensive computation costs. SSD [23] combines the feature maps of the last several convolution layers in their network, allowing predictions of detections at multiple scales. The fusion scheme in SSD is more efficient, since it adds convolutional feature layers to the end of the truncated base network. But, in our opinion, it is still a little bit heavy to a small model. Instead, we propose a lightweight feature fusion method, namely Multi-Scale Feature Fusion (MSFF).

Specifically, we extract the feature maps from each of the four convolutional blocks, corresponding to information captured by different Receptive Fields (RFs). All the feature maps are then down sampled to a fixed size by max

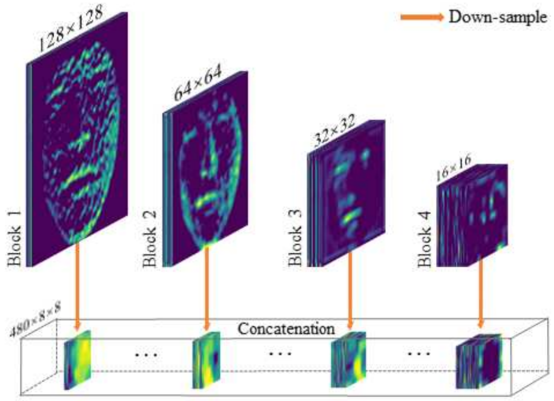


Figure 3. Concatenation of multi-scale feature maps. From left to right are the output feature maps from Block 1, Block 2, Block 3 and Block 4, respectively. Max pooling is used to down-sample these feature maps, with specific parameters (33, 16, 16), (17, 8, 8), (9, 4, 4) and (3, 2, 1), sorting by kernel size, stride and padding.

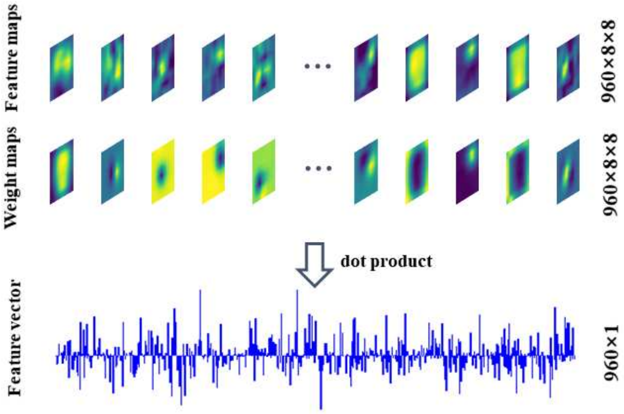


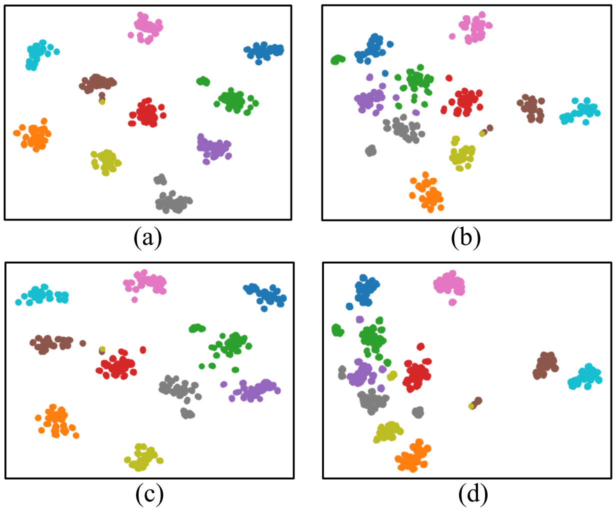
Figure 4. Implementation of spatial attention vectorization. Each feature map has a corresponding weight map, and the feature vector is calculated by dot product in the channel dimension.

pooling for fast processing. We further concatenate them in the channel dimension, as Figure 3 shows. Furthermore, we integrate the feature maps at different scales by another convolution layer consisting of 960 3×3 kernels (Block 5). In this way, we efficiently generate a more discriminative feature to represent the 3D face of a low-quality.

In addition, during model training, the convolution layers in the backbone are directed both by the successive layers as well as the neighboring ones, which can speed up the convergence of the network.

3.2. Spatial Attention Vectorization

Recently, many main-stream CNN architectures, such as Inception V2 [19] and ResNet [9], use the Global Average Pooling (GAP) layer to vectorize feature maps. Compared with the FC layer [31], GAP is much more efficient, and shows good performance in object recognition. However, it is not the best solution for FR. FR is fundamentally different from object recognition because all the faces are well aligned before recognition. For aligned faces, corresponding areas contain fixed facial components. In high-level feature maps, each pixel encodes a specific area of the input image, and the area size is dependent on the RF, thus including fixed semantic information. But GAP clearly ignores such correspondence. It motivates us to investigate another feature generation method which is as efficient as GAP and keeps the spatial cues.



(a) (b)

(c) (d)

Figure 5. Features generated by SAV and some other major pooling methods on the samples of 10 subjects randomly selected from Lock3DFace, visualized using t-SNE: (a) SAV; (b) global average pooling; (c) global max pooling; and (d) global sum pooling.

In this work, we propose a Spatial Attention Vectorization (SAV) module to replace GAP as shown in Figure 4. SAV is implemented by adding an attention weight map to each feature map. In this case, the contributions of pixels at different locations can be separately emphasized in training, and the weights are then fixed for inference. In our network, SAV is applied to the feature maps produced by MSFF, which previously integrates both the low-level and high-level features. In SAV, there are 960 convolution filters related to 960 feature maps, whose kernel size is 8×8, the same as that of feature maps. After training the model by massive faces, SAV sets corresponding weights for each feature map, taking both the strength of abstract representation and spatial information of the input face into account. Thus, the feature vector we calculate conveys more discriminative cues than GAP, benefiting FR.

It is obvious that SAV works as efficiently as GAP. Meanwhile, it is more powerful than the feature computed by GAP. In Figure 5, we visualize the features of the samples of 10 subjects randomly selected from Lock3DFace, achieved by SAV and three other major pooling schemes, involving GAP, global max pooling and global sum pooling.

Compared with the ones of the counterparts, our feature is more compact and separable, indicating the effectiveness of SAV.

4. Finer and Bigger Training Data

The structured light and Time of Flight (TOF) techniques are usually applied to generate low-quality 3D faces in the form of rough surfaces with strong noises, while they are very efficient. Meanwhile, in current databases, the samples for training a deep model are not so sufficient. In this section, we focus on ameliorating the data for training deep models. A preprocessing pipeline (Section 4.1, Section 4.2) as well as a data augmentation scheme (Section 4.3) are specifically proposed to refine the quality and improve the quantity respectively. In addition, we also consider a new scenario that probably appears in the real world, namely 3D FR across quality, where the gallery set includes high-quality data and the probe samples are of low-quality, and discuss how to handle the data for this case in Section 4.4.

4.1. Point-cloud Recovery

Depth frames collected by low-cost 3D sensors usually have heavy noises (*e.g*. spikes and holes) and compared with the entire image, face areas are very small. Therefore they cannot be directly used for FR. Here, we take the Lock3DFace database [37] as an example to discuss the way to recover point-clouds to better support the subsequent steps. The depth images of Lock3DFace are collected by Kinect V2 and five manual landmarks are provided in the first frame of each sequence.

Interpolation. Within the low-quality depth image, the face only occupies a small part, leading to a very low resolution. To solve that, we crop the face of 180 × 180 from the original depth frame (of 512 × 424) based on the x and y coordinates of the nose tip manually labeled and linearly interpolate it to 360 × 360.

Nose-tip Calibration. Nose-tip is usually regarded as the origin of a 3D face. An inaccurate location of the nose-tip greatly degrades final face representation. Although the nose-tip is manually annotated in our case, there may exist holes around it, leading to wrong values in the Z axis. Therefore, we locate a 10 × 10 patch around the given nose-tip based on the x and y coordinates and use its median value rather than the average value as the modified point. After interpolation and nose-tip calibration, we map the cropped face to the 3D domain and cut off the non-facial area based on the estimated nose-tip. The whole process is shown in Figure 6 (a). Compared with the method used in [2], our 3D face is finer and does not contain the non-facial area (background).

4.2. Surface Refinement

To refine the surface, [2] simply uses a bilateral filter to suppress the noises, but in this way, discriminative features can also be damaged. In this study, we follow the subsequent steps.

Outlier Removing. After we crop the face from the depth frame as in Sec. 4.1, there remain some noises inside the sampling sphere. They are actually the outliers in 3D face representation. In this work, we use the method in [27], which sets the threshold of nearest neighbors to remove these outliers.

Face Projecting. To adapt the 3D face to widely investigated 2D image-based CNN training, we project the 3D point-cloud back to the 2D space (depth face). We then pad the depth face to a fixed size (128 × 128 in this work). Finally, we normalize the depth face image to the range of [0, 255].

Hole Filling. The depth faces generated by projection have holes. To fill those holes, we first binarize the depth face to locate them (1 and 0 indicate valid areas and holes respectively), and morphological reconstruction is then applied to the pixels surrounding it.

Normal Estimating. In 3D FR, the depth image is the most widely used representation. However, some studies demonstrate that the face normals (normal maps) are more discriminative [20, 35]. Same as in [35], we compute three normal images, *i.e*. NCIx, NCIy, and NCIz, and then stack them to generate a normal face, as shown in Figure 6 (b).

4.3. Data Augmentation

Since previous public databases of low-quality data are small and CNNs are data hungry, we launch data augmentation techniques to generate more samples for training our Led3D model. Apart from the widely used pose augmentation (out-of-plane rotation), in this work, we propose two new schemes (shape jittering and shape scaling) to adapt to 3D FR on low-quality data. The generated samples are shown in Figure 6 (c).

Pose Generating. Given a point-cloud 3D face, we can synthesize faces with richer pose variations by adjusting the virtual camera parameters. In this work, we generate new facial point-clouds in the range of [-60°, 60°] on yaw and [-40°, 40°] on pitch, with the interval of 20°. For each generated face, we compute depth and normal images.

Shape Jittering. As shown in Figure 1, low-quality faces (in Lock3DFace) usually have very rough surfaces. Motivated by this, we add the Gaussian noise to augmented 3D faces to simulate such changes. By properly controlling the noise level, we do not change the identity information. In this work, the Gaussian noise we use has 0 mean and 2e-5 variance, on the normalized point-clouds. We find that such parameters lead to significant performance enhancement.

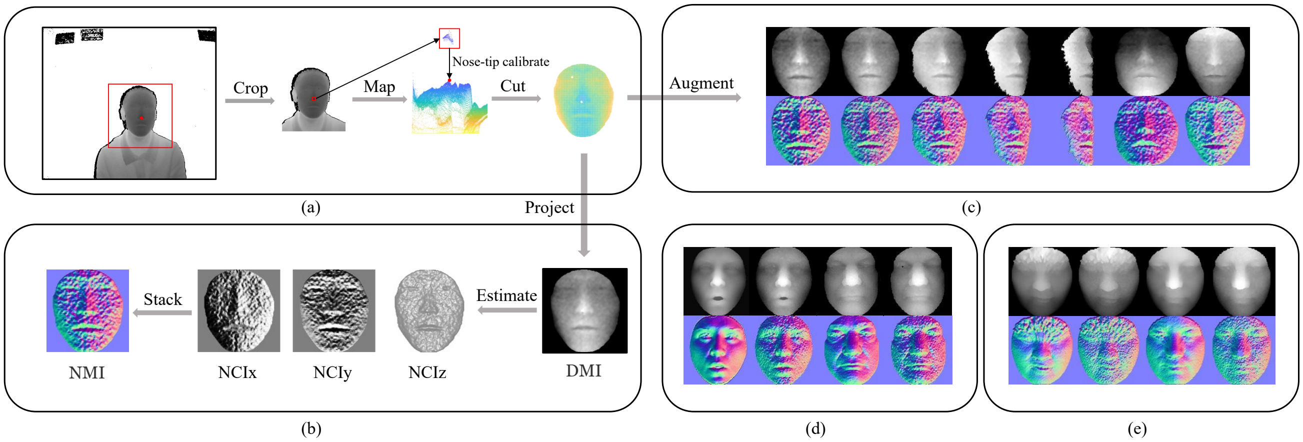


Figure 6. Pipeline of data improvement: (a) facial surface refinement; (b) generation of Depth Map Images (DMI) and Normal Map Images (NMI); (c) augmentation of 3D face samples; (d) generation of low-quality data from high-quality ones; and (e) generation of low-quality and high-quality data of virtual identities.

Shape Scaling. When the faces are collected by 3D cameras, the distance between the face and camera is not fixed. Actually there exist moderate changes on that distance, and the cropped faces thus have varying sizes. To simulate this change, firstly, we binarize the depth face to compute a mask image. Then, we zoom in the depth face image with 1.1 times. Finally, we render the new depth face, which is cropped from the enlarged one via the mask.

4.4. Cross-quality Data Generation

High-precision scanners capture high-quality 3D faces with smooth surfaces, leading to better FR performance. However, such scanners are in big volume and expensive, thus difficult to pervade for on-line scenarios. In comparison, low-quality sensors are more widely used. In the real world, a popular setting is: high-quality data work as gallery and low-quality data are used as probes. To simulate this setting, we convert the high-quality data (from FRGC v2 and Bospohrus in our case) with smooth surfaces to low-quality ones with rough surfaces. Random disturbance is added on high-quality face point-clouds to generate low-quality like depth maps.

Specifically, a 3D face and the disturbance can be represented as $F_i = [x_p, y_p, z_p]$ and $D_i = [d_p]$, respectively. Here $i = 1, ..., N, p = 1, ..., P$ and $D_i \sim N(0, 16)$; N is the number of 3D faces and P is the number of vertices of a 3D face. The generated low-quality like face $F_i^l = [x_p, y_p, z_p^l]$ can be obtained by $z_p^l = z_p + d_p$. Then, we use a maximum filter with a kernel size of 3×3 on every generated face to amplify the effect of the disturbance. Examples of generated low-quality faces from high-quality ones are shown in Figure 6 (d).

Furthermore, we use the virtual ID generation method in [40] to generate new individuals (identities) to increase the data size for cross-quality model training. The sample is shown in Figure 6 (e).

5. Experiments

We mainly evaluate our method on Lock3DFace [37]. To further validate its generalization ability, we synthesize low-quality data on the Bosphorus database [29] for additional analysis. FRGC v2 [26] is used to generate virtual identities as introduced in Section 4.4 for cross-quality model training. The databases, settings and protocols, and results are described in detail in the following.

5.1. Databases

Lock3DFace. It is the most comprehensive database public available, with low-quality 3D faces collected by Kinect V2. It includes 5,671 video sequences of 509 individuals, covering variations in expression, occlusion, pose and time.

Bosphorus. It contains 4,666 3D faces of 105 individuals, presenting variations in expression, occlusion, and pose. Face data are acquired using structured-light 3D system.

FRGC v2. It consists of 4,007 3D face models of 466 individuals, with expression variations. The dataset is collected by a Laser 3D scanner with high-precision and provides two modes of data: RGB and 3D.

5.2. Settings and Protocols

3D FR on Low-quality Data. The experiments are conducted on Lock3DFace. Firstly, we adopt the same settings as in [37]. Specifically, the first depth videos of the neutral expression of all the 509 individuals are used for training, and the remaining ones are divided to four test subsets (expression, occlusion, pose and time). For each video, we select six frames for data augmentation and model training. It should be noted that [37] reconstructs a high-quality model

Table 1. Performance comparison in terms of rank-one score in Lock3DFace using different training sets.

Model	Training data	Evaluation Type	Test subset				
			FE	OC	PS	TM	AVG
Baseline [37]	No augmentation	Video based	74.12	28.57	18.63	13.17	34.53
VGG-16 [31]	No augmentation	Video based	74.49	27.19	8.97	7.61	34.55
ResNet-34 [9]			63.06	21.81	12.92	5.82	30.2
Inception-V2 [19]			78.07	35.36	**14.4**	7.46	39.13
MobileNet-V2 [28]			73.72	27.49	10.75	7.01	34.73
Ours			**79.78**	**36.95**	12.33	**19.85**	**41.65**
VGG-16 [31]	With augmentation	Video based	79.63	36.95	21.7	12.84	42.8
ResNet-34 [9]			62.83	20.32	22.56	5.07	32.23
Inception-V2 [19]			80.48	32.17	33.23	12.54	44.77
MobileNet-V2 [28]			85.38	32.77	28.3	10.6	44.92
Ours			**86.94**	**48.01**	**37.67**	**26.12**	**54.28**

FE: expression. PS: pose. OC: occlusion. TM: time.

from each raw depth video and applies ICP to compute errors between reconstructed models for matching, which is a video to video scenario. In our case, to highlight the contribution of data augmentation, we build two training sets. The first is only with the original data, and in each video used for training, we select six frames at an equal interval, leading to a total of 3,054 (509×6) depth frames. The second is with synthesized data, where the samples in the first set are augmented as introduced in Section 4.3 and each original face renders 12 new ones, and thus of 39,702 (3054×12+3054) in total. For testing, all frames are extracted in each video for independent processing, and their results are combined by simply voting to predict the final label for the entire video. With the two training sets, we train the proposed model and four state-of-the-art CNNs [31, 9, 19, 28] and evaluate the rank-one recognition rate on the four subsets. In our model, all the depth face images (or normal face images) are resized to 128×128, and to adapt to other counterpart networks [31, 9, 19, 28], the input image is scaled to suitable solutions. These models are pre-trained on the combination of FRGC v2 and Bosphorus, and then fine-tuned on Lock3DFace. All CNNs use the same Adam optimizer for training but different batch sizes according to GPU memory. Furthermore, to validate the data augmentation scheme, we train the CNNs in the original training set and the augmented training set respectively.

Cui *et al.* [2] proposes another setting that divides training and test sets by subjects, which is more suitable for learning based methods. All the data of 340 subjects randomly selected are used for training, and those of the remaining 169 subjects are for testing. During the training phase, in each video of the neutral expression, we sample six frames at an equal interval, which are used for augmentation. Such data and the other original data are then adopted for model building. In the test phase, six frames are also extracted from each of the 1,628 videos, with a total of 9,768 frames. Among them, the first frame of the neutral expression of each subject is taken as the gallery sample (169), and the others are used as probes (9,599), including five subsets. Here, we train our network from scratch on the training set and evaluate the performance on the test set. For test, we firstly extract features from the SAV layer by our model. Then, we match the signature of a probe with those of all the identities in the gallery. Finally, the ID is assigned to the probe based on minimum cosine distance. We compare the results using depth faces and the combination of depth and normal.

Ablation Study. We evaluate the contributions of the MSFF and SAV modules. We follow the protocol in [2] with data augmentation. We train four networks: (A) the basic network with only five convolution layers; (B) the basic network with MSFF; (C) the basic network with SAV; and (D) the basic network with both the MSFF and SAV modules. We apply the Adam optimizer for model training, and set the batch size as 384.

Cross-quality 3D FR. To explore this new scenario, we carry out experiments on Bosphorus. The training set contains the augmented high-quality normal face data, the generated low-quality normal face data and the synthsized virtual face data on FRGC v2, with totally 122,150 faces of 1,000 identities. We train the proposed network and the counterpart Inception V2 used in [2] from scratch. For test, we use the first faces of the neutral expression in high-quality of all the 105 individuals as gallery and the remaining ones are processed into a low quality as probes. The identity is determined on minimum cosine distance as well.

5.3. Results

3D FR on Low-quality Data. Table 1 reports the rank-one accuracies of our model and four state-of-the-art CNNs [31, 9, 28, 19] on Lock3DFace, compared with the baseline method [37]. From this table, we can see that the proposed network achieves the best average scores in all the settings, showing its effectiveness. However, for the training data without augmentation, the scores of all the CNN methods on the subset (PS) are lower than Baseline [37] using ICP

Table 2. Performance in terms of rank-one recognition rate (%) of 3D FR using low-quality data on Lock3DFace using the protocol in [2].

Test subset	**Inception V2** [2] Depth	**Ours** Depth	 Depth&Normal
NU	99.55	**99.62**	**99.62**
FE	98.03	97.62	**98.17**
PS	65.26	64.81	**70.38**
OC	**81.62**	68.93	78.10
TM	55.79	64.97	**65.28**
Total	79.85	81.02	**84.22**

Table 3. Comparison in terms of rank-one recognition rate (%) on Lock3DFace with ablations in our proposed network.

Model	**MSFF**	**SAV**	**Lock3DFace** [37] NU	 FE	 PS	 OC	 TM	 Total
A			97.90	90.75	38.31	47.33	34.17	64.88
B	√		97.73	92.62	41.59	49.65	40.23	67.30
C		√	99.14	96.03	55.80	61.23	57.58	76.09
D	√	√	**99.62**	**97.62**	**64.81**	**68.93**	**64.97**	**81.02**

Table 4. Performance of cross-quality 3D FR (HL: high-quality in gallery and low-quality in probe; LL: low-quality in both gallery and probe).

Model	**Bosphorus** [29] HL	 LL
Inception-V2 [19]	78.56	77.23
Ours	**91.27**	**90.70**

Table 5. Comparison in terms of running speed (fps) with four CNNs on Jetson TX2. Low-Power Mode means the default setting of Max-Q, and High-Power Mode means the maximum clock frequency setting of Max-N.

Model	**Jetson TX2** Low-Power Mode GPU	 ARM	 High-Power Mode GPU	 ARM
VGG-16 [31]	7.09	0.43	11.13	0.88
ResNet-34 [9]	8.44	0.58	13.08	1.14
Inception-V2 [19]	24.33	2.90	39.02	5.16
MobileNet-V2 [28]	35.41	3.16	60.41	5.62
Ours	**46.26**	**9.77**	**135.93**	**15.66**

based registration. The reason lies in that the training data are not sufficient and do not contain faces with pose variations.

Once we apply augmentation techniques to training data, the accuracies of CNN models are significantly improved on the test subsets. It means that the proposed data augmentation methods are effective for performance improvement.

Table 2 shows that the proposed method outperforms the state-of-the-art methods using the same protocol, where the training and testing data are separated by subjects. The results for Inception V2 are reported by [2]. They pre-train Inception V2 on their private dataset, which contains 845K faces of 747 identities. Unlike [2], our model is trained from scratch and evaluated on depth faces and normal faces. Our model reports an accuracy of 81.02% on depth faces, around 1.17% higher than that in [2]. In addition, we achieve 84.22% by concatenating the feature of depth and normal, suggesting that these two features have complementary information.

Ablation Study. Table 3 shows the results of four networks. Compared with the baseline netwrok A, we can see MSFF and SAV do improve the performance. On the one hand, the MSFF module extracts more discriminative features by combing the information at different levels. On the other hand, the SAV module proves effective in capturing spatial clues that are crucial to FR. Not surprisingly, combining MSFF and SAV (model D) leads to the best performance.

Cross-quality 3D FR. We report the results of cross-quality 3D FR in Table 4. We can see that the proposed network achieves 91.27% accuracy for HL and 90.7% for LL, both of which are significantly superior to the ones reached by Inception V2, the major counterpart used in [2]. It illustrates that our network is also competent at recognizing 3D face across the change in data quality, where its generalization ability is highlighted.

5.4. Run-time

We evaluate the run-time of the four CNNs and the proposed network on Jetson TX2, which is one of the fastest, most power-efficient embedded AI edge device. The run-time is computed on a single inference using MXNet 1.2 and python 2.7. We set the device in different power modes and compute in different processors. As shown in Table 5, the network we propose runs at a speed of 136 fps in the high-power mode, which is much faster than MobileNet V2. If we use the ARM core process, it also achieves 15 fps, faster than MobileNet V2 as well. Furthermore, for the proposed data preprocessing method, the average run-time of each frame is 0.13s. It verifies that our solution is efficient and can be deployed on edge devices to achieve real time 3D FR using low-quality data.

6. Conclusion

In this paper, we propose a novel lightweight CNN for 3D FR using low-quality data. To achieve fast processing and accurate prediction, we propose the MSFF and SAV modules, both of which enhance representation of 3D faces in an efficient manner. In addition, we propose a systematical solution for data processing and augmentation. The proposed Led3D model achieves the state-of-the-art accuracy on Lock3DFace [37]. In terms of run-time, it operates at a high speed up to 136 fps on Jetson TX2.

Acknowledgment

This work is funded by the National Natural Science Foundation of China under Grant 61673033.

References

[1] S. Berretti, A. Del Bimbo, and P. Pala. Superfaces: A super-resolution model for 3d faces. In *ECCV*, 2012. 2, 3

[2] J. Cui, H. Zhang, H. Han, S. Shan, and X. Chen. Improving 2d face recognition via discriminative face depth estimation. In *ICB*, 2018. 2, 3, 5, 7, 8

[3] H. Drira, B. B. Amor, A. Srivastava, M. Daoudi, and R. Slama. 3d face recognition under expressions, occlusions, and pose variations. *TPAMI*, 35(9), 2013. 2

[4] N. Erdogmus and S. Marcel. Spoofing face recognition with 3d masks. *TIFS*, 9(7), 2014. 2

[5] T. C. Faltemier, K. W. Bowyer, and P. J. Flynn. Using a multi-instance enrollment representation to improve 3d face recognition. In *BTAS*, 2007. 2

[6] S. Z. Gilani, A. Mian, and P. Eastwood. Deep, dense and accurate 3d face correspondence for generating population specific deformable models. *PR*, 69, 2017. 1, 2

[7] G. Goswami, S. Bharadwaj, M. Vatsa, and R. Singh. On rgb-d face recognition using kinect. In *BTAS*, 2013. 2, 3

[8] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. MS-Celeb-1M: A dataset and benchmark for large scale face recognition. In *ECCV*, 2016. 2

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4, 7, 8

[10] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. In *CVPR*, 2017. 3

[11] G. Hu, Y. Hua, Y. Yuan, Z. Zhang, Z. Lu, S. S. Mukherjee, T. M. Hospedales, N. M. Robertson, and Y. Yang. Attribute-enhanced face recognition with neural tensor fusion networks. In *ICCV*, 2017. 1

[12] G. Hu, X. Peng, Y. Yang, T. M. Hospedales, and J. Verbeek. Frankenstein: Learning deep face representations using small data. *TIP*, 27(1), 2018. 2

[13] G. Hu, Y. Yang, D. Yi, J. Kittler, W. Christmas, S. Z. Li, and T. Hospedales. When face recognition meets with deep learning: an evaluation of convolutional neural networks for face recognition. In *ICCVW*, 2015. 2

[14] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, 2007. 1

[15] R. Huang, S. Zhang, T. Li, R. He, et al. Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis. In *ICCV*, 2017. 2

[16] T. Huynh, R. Min, and J.-L. Dugelay. An efficient lbp-based descriptor for facial depth images applied to gender recognition using rgb-d face data. In *ACCV*, 2012. 2

[17] F. Iandola, M. Moskewicz, S. Karayev, R. Girshick, T. Darrell, and K. Keutzer. Densenet: Implementing efficient convnet descriptor pyramids. In *CVPR*, 2014. 3

[18] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016. 3

[19] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 3, 4, 7, 8

[20] I. A. Kakadiaris, G. Passalis, G. Toderici, M. N. Murtuza, Y. Lu, N. Karampatziakis, and T. Theoharis. Three-dimensional face recognition in the presence of facial expressions: An annotated deformable model approach. *TPAMI*, 29(4), 2007. 2, 5

[21] D. Kim, M. Hernandez, J. Choi, and G. Medioni. Deep 3d face identification. In *IJCB*, 2017. 2

[22] H. Li, D. Huang, J.-M. Morvan, Y. Wang, and L. Chen. Towards 3d face recognition in the real: a registration-free approach using fine-grained matching of 3d keypoint descriptors. *IJCV*, 113(2), 2015. 1, 2

[23] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016. 3

[24] R. Min, N. Kose, and J.-L. Dugelay. Kinectfacedb: A kinect database for face recognition. *TSMC: Systems*, 44(11), 2014. 2, 3

[25] O. M. Parkhi, A. Vedaldi, A. Zisserman, et al. Deep face recognition. In *BMVC*, volume 1, 2015. 2

[26] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the face recognition grand challenge. In *CVPR*, 2005. 1, 6

[27] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz. Towards 3d point cloud based object maps for household environments. *RAS*, 56(11), 2008. 5

[28] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. In *CVPR*, 2018. 7, 8

[29] A. Savran, N. Alyüz, H. Dibeklioğlu, O. Çeliktutan, B. Gökberk, B. Sankur, and L. Akarun. Bosphorus database for 3d face analysis. In *BIOID*, 2008. 1, 2, 6, 8

[30] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015. 1, 2

[31] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 4, 7, 8

[32] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, 2014. 2

[33] Y. Wang, S. Chen, W. Li, D. Huang, and Y. Wang. Face anti-spoofing to 3d masks by combining texture and geometry features. In *CCBR*, 2018. 2

[34] D. Wu, F. Zhu, and L. Shao. One shot learning gesture recognition from rgbd images. In *CVPRW*, 2012. 2

[35] X. Yang, D. Huang, Y. Wang, and L. Chen. Automatic 3d facial expression recognition using geometric scattering representation. In *FG*, volume 1, 2015. 5

[36] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato. A 3d facial expression database for facial behavior research. In *FG*, 2006. 1

[37] J. Zhang, D. Huang, Y. Wang, and J. Sun. Lock3DFace: A large-scale database of low-cost kinect 3d faces. In *ICB*, 2016. 2, 3, 5, 6, 7, 8

[38] W. Zhang, Z. Xi, J.-M. Morvan, and L. Chen. Improving shadow suppression for illumination robust face recognition. *TPAMI*, 2018. 2

[39] X. Zhang, X. Zhou, M. Lin, and J. Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *CVPR*, 2018. 3

[40] S. Zulqarnain Gilani and A. Mian. Learning from millions of 3d scans for large-scale 3d face recognition. In *CVPR*, 2018. 1, 2, 6