FeatherNets: Convolutional Neural Networks as Light as Feather for Face Anti-spoofing

Peng Zhang[2,3], Fuhao Zou[1]*, Zhiwen Wu[2], Nengli Dai[3]
Skarpness Mark[2], Michael Fu[2], Juan Zhao[2], Kai Li[1]

[1]School of Computer Science and Technology, Huazhong University of Science and Technology

[2]Intel IAGS SSP

[3]Wuhan National Laboratory for Optoelectronics, Huazhong University of Science and Technology

{peng3zhang,fuhao_zou,dainl}@hust.edu.cn

{peng3.zhang,zhiwen.wu,juan.j.zhao,michael.fu,mark.skarpness}@intel.com

Abstract

Face Anti-spoofing gains increased attentions recently in both academic and industrial fields. With the emergence of various CNN based solutions, the multi-modal(RGB, depth and IR) methods based CNN showed better performance than single modal classifiers. However, there is a need for improving the performance and reducing the complexity. Therefore, an extreme light network architecture(FeatherNet A/B) is proposed with a streaming module which fixes the weakness of Global Average Pooling and uses less parameters. Our single FeatherNet trained by depth image only, provides a higher baseline with ***0.00168 ACER****,* ***0.35M parameters and 83M FLOPS****. Furthermore, a novel fusion procedure with "ensemble + cascade" structure is presented to satisfy the performance preferred use cases. Meanwhile, the MMFD dataset is collected to provide more attacks and diversity to gain better generalization. We use the fusion method in the Face Anti-spoofing Attack Detection Challenge@CVPR2019 and got the result of* ***0.0013(ACER), 0.999(TPR@FPR=10e-2), 0.998(TPR@FPR=10e-3) and 0.9814(TPR@FPR=10e-4)****.*

1. Introduction

Currently, face recognition is an important way for identity authentication systems. However, it confronts with the challenge caused by face spoofing attacks such as the 2D/3D Presentation Attack. Therefore, it is important to equip the system with robust anti-spoofing algorithms. Anti-spoofing is usually regarded as a problem of binary classification. Some works are texture-based using binary classifiers with handcrafted features[1, 2, 3, 4]. However, these methods suffer from poor generalization because the texture information varies with cameras/capture devices. Another problem of texture-based approaches is that the texture information is not as discriminative as the depth information on task of 2D presentation attack detection.

The depth information is more discriminative since the depth of the real face is uneven, and the depth images of the attacking face is plane. Atoum *et al.* [5] exploited the depth supervised procedure. Nevertheless, the depth information is estimated from RGB image and not as accurate as the depth image captured by depth camera such as RealSense 300[1].

Recently, deep learning techniques are widely used to extract deep features[6, 7, 8], which have richer semantical information compared to traditional handcrafted features. Hence utilizing the deep learning for face PAD has been widely used recently.

However, there is a new trend that face recognition is gradually moving to the mobile devices or embedded devices. This requires the face anti-spoofing algorithms to run with less computation and storage costs. From this perspective, the design of deep learning based anti-spoofing algorithms become more challenge in the mobile or embedded environments. Thus, it is necessary to develop a light-weight deep learning algorithm so that spoofing detection can be used.

To address the issues of computational and storage costs, we design a light-weight CNN architecture (named as FeatherNet) which gets a higher accuracy and computational complexity. Firstly, FeatherNets have a thin CNN stem, thus the computational cost is less. Secondly, a new architecture (named as Streaming Module) is proposed, which has better performance in terms of accuracy than the

*Corresponding author

[1] https://realsense.intel.com/

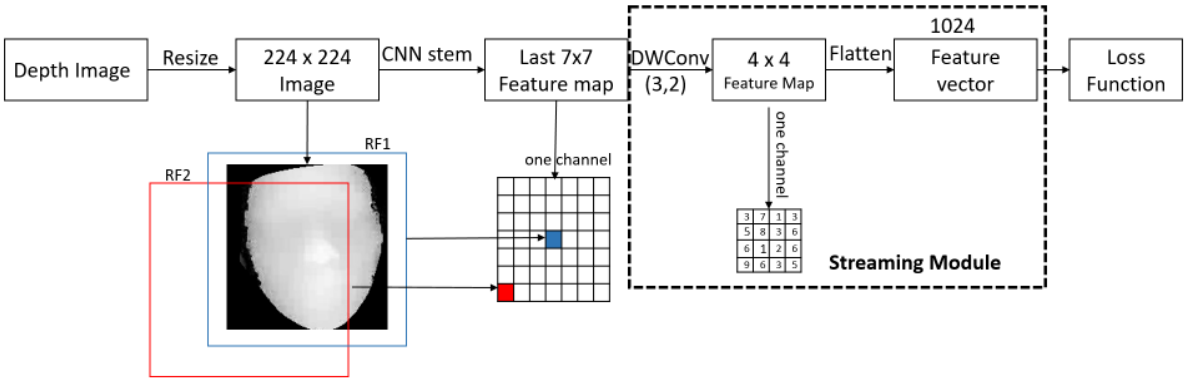

Figure 1. Our depth faces feature embedding CNN structure. In the last 7×7 feature map, the receptive field and the edge (RF2) portion of the middle part (RF1) is different, because their importance is different. DWConv is used instead of the GAP layer to better identify this different importance. At the same time, the fully connected layer is removed, which makes the network more portable.

Global Average Pooling (GAP) approach.

We also design a new fusion classifier architecture which assembles and cascades several models learned from multi-modal data, *i.e.*, the depth and IR data, to generate better prediction accuracy than single depth models. Although the depth image is discriminative on 2D presentation attack detection, multi-modal fusion can boost the performance further due to its complementary and generalization capability[9]. The new fusion procedure has been applied to face anti-spoofing competition@CVPR2019 and showed the result of 0.0013 (ACER), 0.999 (TPR@FPR=10e-2), 0.998 (TPR@FPR=10e-3) and 0.9814 (TPR@FPR=10e-4) in the test dataset.

The major contributions of this paper are summarized as follows: **a)**.An extremely light CNN architecture with a *Streaming Module* which has good performance; **b)**.A novel *fusion procedure* with "ensemble+cascade structure which outperforms the single model classifiers; **c)**.A new Multi-Modal Face Dataset (*MMFD*) is collected which will be released recently and a new data augmentation algorithm is applied on training.

2. Related work

The related work is reviewed in two categories in chronological order: traditional and CNN based methods.

Traditional: Face anti-spoofing is treated as a binary classification problem by traditional SVM (Support Vector Machine), through two steps as below:

1) Crafted features detection: Various filters were used to detect the points to present the feature. The widely adopted features include: Local Binary Patterns (LBP) [10, 3, 1], Scale Invariant Feature Transform (SIFT)[11], Speeded-Up Robust Features (SURF)[4], histogram of oriented gradients (HOG)[2, 12], Difference of Gaussian (DoG)[12].

2) Liveness or not classification through SVM or Random Forest[13].

However, Wang *et al.* [14] indicated that the feature detection is greatly influenced by the environment, for example the lighting condition. Furthermore, the feature detection shows limited features, and the feature points don't provide as many features' information as those CNN methods could bring with the huge data sets.

CNN based: There are mainly three types of CNN based PAD.

1) Using RGB single frame with binary supervision[7, 8]: Most approaches just adopt the final fully-connected layer to distinguish the real and fake faces. While Li *et al.* [7] proposed a way to link the deep partial features (from CNN) and Principle Component Analysis (PCA) to reduce the dimension, and lastly they used SVM to distinguish real and fake faces. Patel *et al.* [8] applied the action features (such as eye blinking) to enhance the state of the art. And the researchers found that this can still be improved through multiple supervisions.

2)Using RGB multi-frame with depth or rPPG (remote photoplethysmography) supervision[5, 15]: Two different types of supervision are applied: depth or rPPG. Different frames are also captured by the shift of camera and frames to anticipate the depth. Moreover, researchers analyzed the presentation attack and video-based pulse detection. Live Faces could show some blood signal through rPPG but not in fake face. Recently, Liu *et al.* [16] proposed a procedure which uses single frame to regression depth map and uses multi-frame to predict rPPG, which is a good way to distinguish living face. This network architecture combining CNN and RNN, could simultaneously estimate the depth map and rPPG signal of the face.

3)Recently, Zhang *et al.* [9] provided a large-scale multi-modal dataset, namely CASIA-SURF, which consists of 3 modalities data (RGB, depth and IR). It provides a strong baseline to make full use of these features by fusing multi-

modal data through a three-stream network.

There are two main aspects to enhance for the multi-modal method: (1) The baseline performance of CASIA-SURF still has a lot of room to improve; (2) The adoption of light-weight network architecture that can benefit more edge side applications. In next section, an extreme lite network architecture is proposed which uses depth and IR information as supervision respectively to learn complementary models, achieving a well trade-off between performance and computational burden. Furthermore, a novel fusion classifier with "ensemble + cascade" structure is proposed for the performance preferred use case.

3. Approach

In this section, we will introduce the details of the FeatherNets. Inspired by the equal importance gap of Global Average Pooling (GAP) in face tasks, a new Streaming module is adopted in the FeatherNets which can provide a strong baseline for Face Anti-spoofing. Furthermore, to achieve higher performance, the "ensemble + cascade" fusion procedure will be proposed.

3.1. FeatherNet Architecture Design

The existing anti-spoofing networks[8, 7, 15, 14] have the problems of large parameters and weak generalization ability. For this reason, FeatherNets architecture is proposed, targeting a network as lite as feather.

3.1.1 The Weakness of GAP for Face Task

Global Average Pooling (GAP) is employed by a lot of state-of-the-art networks for object recognition task, e.g.ResNets[17], DenseNet[18] and some light-weight networks, like MobilenetV2[19], Shufflenet_v2[20], IGCV3[21]. GAP has been proved on its ability of reducing dimensions and preventing over-fitting for the overall structure[22]. However, for the face related tasks, Wu[23] and Deng[24] have observed that CNNs with GAP layer are less accurate than those without GAP. Meanwhile, MobileFaceNet[25] replaces the GAP with Global Depthwise Convolution (GDConv) layer, and explains the reason why it is effective through the theory of receptive field[26]. The main point of GAP is "equal importance" which is not suitable for face tasks.

As shown in Figure 1, the last 7×7 feature map is denoted as FMap-end, each cell in FMap-end corresponds to a receptive field at different position. The center blue cell corresponds to RF1 and the edge red one corresponds to RF2. As described in[27], the distribution of impact in a receptive field distributes as a Gaussian, the center of a receptive field has more impact on the output than the edge. Therefore, RF1 has larger effective receptive field than RF2. For our face anti-spoofing task, the network input is 224×224 images which only contain the face region. As above analysis, the center unit of FMap-end is more important than the edge one. GAP is not applicable to this case. One choice is to use fully connected layer instead of GAP, this will introduce large number of parameters to the whole model and increase the risk of over-fitting.

3.1.2 Streaming Module

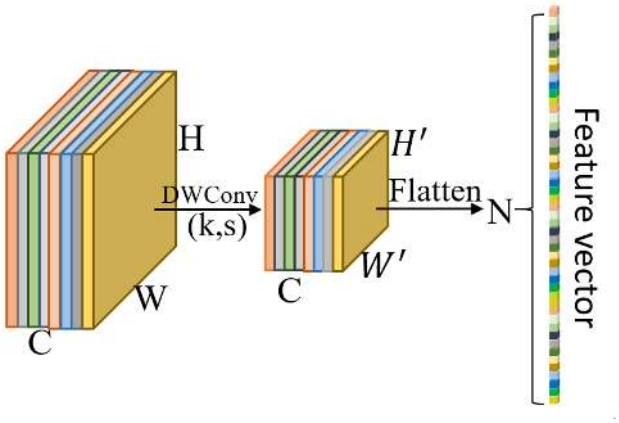

Figure 2. Streaming Module. The last blocks' output is down-sampled by a depthwise convolution[28, 29] with stride larger than 1 and flattened directly into an one-dimensional vector.

To treat different units of FMap-end with different importance, Streaming Module is designed, as shown in the Figure 2. In Streaming Module, a depthwise convolution (DWConv) layer with stride larger than 1 is used for down-sampling whose output, is then flattened directly into an one-dimensional feature vector. The compute process is represented by equation (1).

$$FV_{n(y,x,m)} = \sum_{i,j} K_{i,j,m} \cdot F_{IN_y(i),IN_x(j),m} \qquad (1)$$

In equation 1, FV is the flattened feature vector while $N = H^{'} \times W^{'} \times C$ elements ($H^{'}$, $W^{'}$ and C denote the height, width and channel of DWConv layer's output feature maps respectively). $n(y, x, m)$, computed as equation (2), denotes the n_{th} element of FV which corresponds to the (y, x) unit in the m_{th} channel of the DWConv layer's output feature maps.

$$n(y, x, m) = m \times H^{'} \times W^{'} + y \times H^{'} + x \qquad (2)$$

On the right side of the equation (1), K is the depthwise convolution kernel, F is the FMap-end of size H×W×C (H, W and C denote the height, width and channel of FMap-end respectively). m denotes the channel index. i,j denote the spatial position in kernel K, and $IN_y(i)$, $IN_x(j)$ denote the

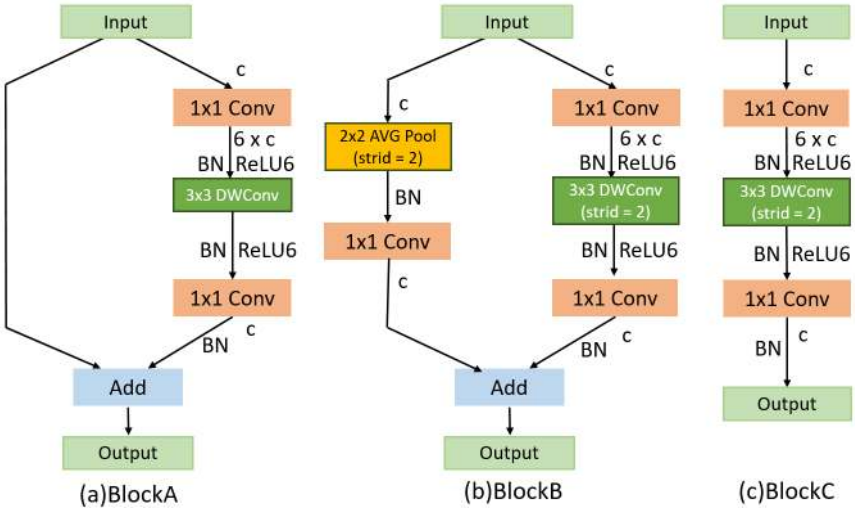

Figure 3. FeatherNets' main blocks. FeatherNetA includes BlockA & BlockC. FeatherNetB includes BlockA & BlockB. (BN: BatchNorm; DWConv: depth wise convolution; c:number of input channels.)

corresponding position in F. They are computed as equation (3), (4).

$$IN_y(i) = y \times S_0 + i \tag{3}$$

$$IN_x(j) = x \times S_1 + j \tag{4}$$

S_0 is the vertical stride and S_1 is the horizontal stride. A fully connected layer is not added after flattening feature map, because this will increase more parameters and the risk of overfitting. Meanwhile, related experiments are processed to verify the reason for removing the fully connected layer, as show in Table 4.

Streaming module can be used to replace global average pooling and fully connected layer in traditional networks.

3.1.3 Network Architecture Detail

Besides Streaming Module, there are BlockA/B/C as shown in Figure 3 to compose FeatherNetA/B. The detailed structure of the primary FeatherNet architecture is shown in Table 1. **BlockA** is the inverted residual blocks proposed in MobilenetV2[19]. BlockA is used as our main building block which is shown in the Figure 3(a). The expansion factors are the same as in MobilenetV2[19] for blocks in our architecture. **BlockB** is the down-sampling module of FeatherNetB. Average pooling (AP) has been proved in Inception[30] to benefit performance, because of its ability of embedding multi-scale information and aggregating features in different receptive fields. Therefore, average pooling (2×2 kernel with stride = 2) is introduced in BlockB (Figure 3(b)). Besides, in the network ShuffleNet[20], the down-sampling module joins 3×3 average pooling layer with stride=2 to obtain excellent performance. Li *et al.* [31] suggested that increasing average pooling layer works well and impacts the computational cost little. Based on the above analysis, adding pooling on the secondary branch can learn more diverse features and bring performance gains. The performance comparison between using the auxiliary branch (BlockB in Figure 3(b)) and not using the branch (BlockC in Figure3(c)) is showing in the Table 4. **BlockC** is the down-sampling Module of our network FeatherNetA. BlockC is faster and with less complexity than BlockB. According to our experiment in Table 2, FeatherNetA used less parameters.

Input	Operator	t	c
$224^2 \times 3$	Conv2d,/2	-	32
$112^2 \times 32$	BlockB	1	16
$56^2 \times 16$	BlockB	6	32
$28^2 \times 32$	BlockA	6	32
$28^2 \times 32$	BlockB	6	48
$14^2 \times 48$	5xBlockA	6	48
$14^2 \times 48$	BlockB	6	64
$7^2 \times 64$	2xBlockA	6	64
$7^2 \times 64$	Streaming	-	1024

Table 1. Network Architecture: FeatherNet B. All spatial convolutions use 3×3 kernels. The expansion factor t is always applied to the input size, while c means number of Channel. Meanwhile, every stage SE-module[32] is inserted with reduce = 8. And FeatherNetA replaces BlockB in the table with BlockC.

After each down-sampling stage, SE-module[32] is inserted with reduce = 8 in both FeatherNetA and FeatherNetB. In addition, when designing the model, a fast down-sampling strategy[33] is used at the beginning of our network which makes the feature map size decrease rapidly and without much parameters. Adopting this strategy can avoid the problem of weak feature embedding and high processing time caused by slow down-sampling due to limited

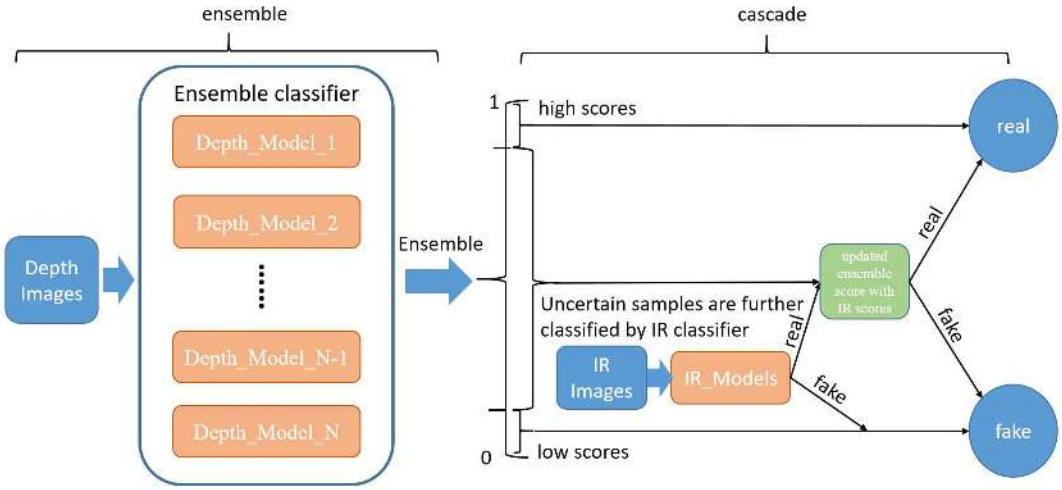

Figure 4. Multi-Modal Fusion Strategy: Two stages cascaded, stage 1 is an ensemble classifier consisting of several depth models. Stage 2 employs IR models to classify the uncertain samples from stage 1.

computing budget[34]. The primary FeatherNet only has 0.35M parameters.

The FeatherNets' structure is built on BlockA/B/C as mentioned above except for the first layer which is a fully connected. As shown in Table 1, the size of the input image is 224 $\times$ 224. A layer with regular convolutions, instead of depthwise convolutions, is used at the beginning to keep more features. Reuse channel compression to reduce 16 while using inverted residuals and linear bottleneck with expansion ratio = 6 to minimize the loss of information due to down-sampling. Finally, the Streaming module is used without adding a fully connected layer, directly flatten the $4 \times 4 \times 64$ feature map into an one-dimensional vector, reducing the risk of over-fitting caused by the fully connected layer. After flattening the feature map, focal loss is used directly for prediction. The related ablation experiments are shown in the Table 4. When we added the fully connected layer, the performance dropped.

3.2. Multi-Modal Fusion Method

The main idea for the fusion method is to use cascade inference on different modals: depth images and IR images. The model trained based on depth data could provide a high baseline (approximately 0.003 ACER in test set). According to our experiments, the IR data could provide a good performance in fake judgement for those samples that depth modal is not sure about. The cascade structure has two stages, as show in the Figure 4:

Stage 1: An ensemble classifier, consisting of multiple models , is employed to generate the predictions. These models are trained on depth data and from several checkpoints of different networks, including FeatherNets. If the weighted average of scores from these models is near 0 or 1, input sample will be classified as fake or real respectively. Otherwise, the uncertain samples will go through the second stage.

Stage 2: FeatherNetB learned from IR data will be used to classify the uncertain samples from stage 1. The fake judgement of IR model is respected as the final result. For the real judgement, the final scores are decided by both stage 1 and IR models.

4. Experiments

The preliminary work will be introduced firstly, such as the evaluation metrics, datasets used for training, the proposed data augmentation method, the training settings of the FeatherNets and the baseline models. Secondly, the performance of the trained models (including FeatherNets) will be showed. Thirdly, the comparative experiments are used to show the validity of the MMFD dataset. Finally, the effectiveness of the network design is verified by ablation experiments.

4.1. Preliminary Work

4.1.1 Evaluation Metrics

For the performance evaluation, the following commonly used metrics[2] will be introduced: Attack Presentation Classification Error Rate (APCER), Normal Presentation Classification Error Rate (NPCER) and Average Classification Error Rate (ACER). ACER is treated as the evaluation metric, in which APCER and NPCER are used to measure the error rate of fake or real samples, respectively. Besides, the other metrics[9] are also used, such as TPR@FPR=10E-2, 10E-3, 10E-4.

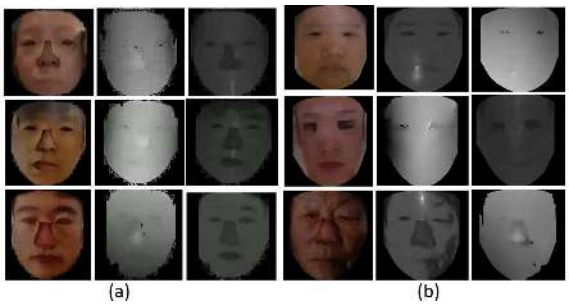

Figure 5. (a)Training set contains attacks 4,5,6 (b)Validation and test sets contains attacks 1,2,3

.

4.1.2 Datasets

Two datasets are used in the experiments: CASIA-SURF[9] and the proposed Multi-Modal Face Dataset (MMFD).

CASIA-SURF is the largest publicly available dataset for face Anti-spoofing, provided by Surfing Technology[9]. It consists of 1,000 subjects with 21,000 videos and each sample has 3 modalities (*i.e.*, RGB, Depth and IR), as shown in Figure 5. There are 6 attack ways of this dataset: *Attack 1*: One person holds his/her flat face photo where eye regions are cut from the printed face. *Attack 2*: One person holds his/her curved face photo where eye regions are cut from the printed face. *Attack 3*: One person holds his/her flat face photo where eyes and nose regions are cut from the printed face. *Attack 4*: One person holds his/her curved face photo where eyes and nose regions are cut from the printed face. *Attack 5*: One person holds his/her flat face photo where eyes, nose and mouth regions are cut from the printed face. *Attack 6*: One person holds his/her curved face photo where eyes, nose and mouth regions are cut from the printed face.

MMFD In order to make the model more robust, more attack ways of diverse faces are collected. Then we sort out a dataset which is consisted of 15 subjects with 15415 real samples and 28438 fake samples, namely Multi-Modal Face Dataset (MMFD).

And each sample also has 3 modalities (RGB, Depth, IR). They are treated by the similar way as CASIA-SURF with a little modification. Besides the 6 attack ways of CASIA-SURF, 2 new attack ways are added. *Attack A*: One person holds his/her flat face photo where eyes and mouth regions are cut from the printed face. *Attack B*: One person holds his/her curved face photo where eyes and mouth regions are cut from the printed face. The presenters turn their head left/right/up/down to get different samples. Other variations on the presenters include: wearing glasses or not; opening mouth or not; moving face close to and far away from the camera; showing different emotions, e.g. happy, angry, sad and so on.

Collecting and masking steps are proposed to obtain the final images. *Collecting*: Intel RealSense SR300[2] camera is used to generate RGB, Depth, IR and aligned-RGB frames simultaneously. RGB frame is 1280×720 resolution, Depth, IR and Aligned-Depth frames are 640×480 resolution. *Masking*: Dlib[35] is used to detect the bounding-box of face for RGB frame and Aligned-Depth frame. And the face region is passed into PRNet[36] to estimate the depth. To generate the mask image, the depth value of each pixel is checked in face box. If it is larger than 0.5, 1 will be sent otherwise 0 will be sent into mask image. At last, the RGB, Depth and IR images are multiplied with the mask, and only the face region is saved to files.

4.2. Implementation Detail

4.2.1 Data Augmentation

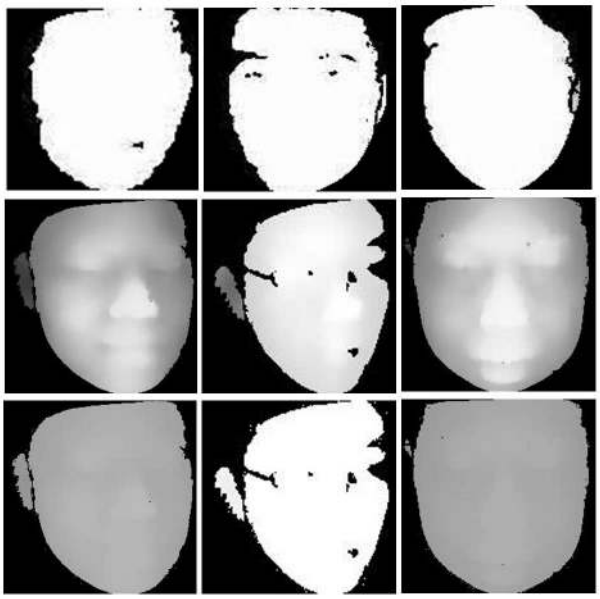

Figure 6. depth image augmentation.(line 1): CAISA-SURF real depth images; (line 2): MMFD real depth images; (line 3): our augmentation method on MMFD.

There are some differences in the images acquired by different devices, even if the same device model is used. As shown in the Figure 6. The upper line is the depth images of the CASIA-SURF data set. The depth difference of the face part is small. It is difficult for the eyes to distinguish whether the face has a contour depth. The second line is the depth images of the MMFD dataset whose outline of the faces are clearly showed. In order to reduce the data difference caused by the device, the depth of the real face images is scaled in MMFD which can be seen in the third line of Figure 6. The way of data augmentation is as Algorithm 1:

[2] https://realsense.intel.com/

Model	ACER	TPR@FPR=10E-2	TPR@FPR=10E-3	Params	FLOPS
ResNet18[9]	0.05	0.883	0.272	11.18M	1800M
Baseline[9]	0.0213	0.9796	0.9469	–	–
FishNet150(our impl)	0.00144	0.9996	0.998330	24.96M	6452.72M
MobilenetV2(1)(our impl)	0.00228	0.9996	0.9993	2.23M	306.17M
ShuffleNetV2(1)(our impl)	0.00451	1.0	0.98825	1.26M	148.05
FeatherNetA	0.00261	1.0	0.961590	0.35M	79.99M
FeatherNetB	**0.00168**	**1.0**	**0.997662**	**0.35M**	**83.05M**

Table 2. Performance in validation dataset. Baseline is a way of fusing three modalities data (IR, RGB, Depth) through a three-stream network. Only depth data is used for training in the other networks. FeatherNetA and FeatherNetB have achieved higher performance with less parameters. Finally, the models are assembled to reduce ACER to 0.0.

Algorithm 1 Data Augmentation Algorithm

$scaler \leftarrow$ a random value in range [1/8, 1/5]
$offset \leftarrow$ a random value in range [100, 200]
$OutImg \leftarrow 0$
for $y = 0 \rightarrow Height - 1$ **do**
  for $x = 0 \rightarrow Width - 1$ **do**
    if $InImg(y, x) > 20$ **then**
      $off \leftarrow offset$
    else
      $off \leftarrow 0$
    end if
    $OutImg(y, x) \leftarrow$ InImg(y,x) * scaler + off
  end for
end for
return $OutImg$

4.2.2 Training Strategy

Pytorch[37] is used to implement the proposed networks. It initializes all convolutions and fully-connected layers with normal weight distribution[38]. For optimization solver, Stochastic Gradient Descent(SGD) is adopted with both learning rate beginning at 0.001, and decaying 0.1 after every 60 epochs, and momentum setting to 0.9. The Focal Loss[39] is employed with $\alpha = 1$ and $\gamma = 3$.

4.3. Result Analysis

4.3.1 How useful is MMFD dataset?

A comparative experiment is executed to show the validity and generalization ability of our data. As shown in Table 3, the ACER of FeatherNetB with MMFD depth data is better than that with CASIA-SURF[9], though only 15 subjects are collected. Meanwhile, the experiment shows that the best option is to train the network with both data. The results of using our FeatherNetB are much better than the baselines that use multi-modal data fusion, indicating that our network has better adaptability than the third-stream ResNet18 for baseline.

Network	Training Dataset	ACER in Val
Baseline	CASIA-SURF	0.0213
FeatherNetB	CASIA-SURF depth	0.00971
FeatherNetB	MMFD depth	0.00677
FeatherNetB	CASIA-SURF+ MMFD depth	**0.00168**

Table 3. Performance of FeatherNetB training by different datasets. Column 3 means the ACER value in the validation dataset of CASIA-SURF[9]. It shows that our dataset MMFD generalization ability is stronger than baseline of CASIA-SURF. The performance is better than the baseline method using multi-modal fusion.

4.3.2 Compare with other network performance

As show in Table 2, experiments are executed to compare with other network's performance. All experimental results are based on depth of CASIA-SURF and MMFD depth images, and then the performance is verified on the CASIA-SURF verification set. It can be seen from the table 2 that our parameter size is much smaller, only 0.35M, while the performance on the verification set is the best.

4.3.3 Ablation Experiments

A number of ablations are executed to analyze different models with different layer combination, shown in Table 4. The models are trained with CASIA-SURF training set and MMFD dataset.

Why AP-down in BlockB: Comparing *Model1 and Model2*, Adding the Average Pooling branch to the secondary branch (called AP-down), as shown in block B of Figure 3(b), can effectively improve performance with a small number of parameters.

Why not use FC layer: Comparing *Model1 and Model3*, fully connected (FC) layer doesn't reduce the error when adding a fully connected layer to the last layer of the network. Meanwhile, a FC layer is computationally expensive.

Why not use GAP layer Comparing *Model3 and Model4*, it shows that adding global average pooling layer at the end of the network is not suitable for face anti-spoofing task. They will reduce performance. For more details, please refer to Section 3.

Model	FC	GAP	AP-down	ACER
Model1	×	×	×	0.00261
Model2	×	×	✓	**0.00168**
Model3	✓	×	×	0.00325
Model4	✓	✓	×	0.00525

Table 4. Ablation Experiments.

5. Competition details

Based on CASIA-SURF[9], the Face Anti-spoofing challenge@CVPR2019 has been organized, aiming at compiling the latest efforts and research advances from the computational intelligence community in creating fast and accurate face spoofing detection algorithms[3]. This dataset provides a multi-modal dataset (RGB, Depth, IR) which is captured by Intel RealSense SR300. And it contains data for training, verification and the final evaluation.

Our fusion procedure (described in section 3.2) is applied in this competition. Meanwhile, the proposed FeatherNets with depth data only can provide a higher baseline alone (around 0.003 ACER). During the fusion procedure, the selected models are with different statistic features, and can help each other. For example, one model's characteristics of low False Negative (FN) are utilized to further eliminate the fake samples. The detailed procedure is described as below:

Training: The depth data is used to train 7 models: FishNet150_1, FishNet150_2, MobilenetV2, FeatherNetA, FeatherNetB, FeatherNetBForIR, ResNet_GC. Meanwhile, FishNet150_1, FishNet150_2 are models from different epoch of FishNet. The IR data is used to train FeatherNetB as FeatherNetBforIR.

Inference: The inference scores will go through the "ensemble + cascade" process. The algorithm is shown as Algorithm 2.

Competition Result: The above procedure is used to get the result of 0.0013 (ACER), 0.999 (TPR@FPR=10e-2), 0.998 (TPR@FPR=10e-3) and 0.9814 (TPR@FPR=10e-4) in the test set and showed excellent performance in the Face Anti-spoofing challenge@CVPR2019.

6. Conclusion

We propose an extreme lite network architecture (FeatherNetA/B) with Streaming module, to achieve a well trade-off between performance and computational complexity for multi-modal face anti-spoofing. Furthermore, a novel fusion classifier with "ensemble + cascade" structure is proposed for the performance preferred use cases. Meanwhile, MMFD dataset is collected to provide more diverse samples and more attacks to gain better generalization ability. All these are used to join the Face Anti-spoofing Attack Detection Challenge@CVPR2019. The experiment and the competition results show that the proposed method can achieve excellent performance.

Algorithm 2 Ensemble Algorithm

$scores[] \leftarrow$
    score_FishNet150_1,
    score_FishNet150_2,
    score_ MobilenetV2,
    score_FeatherNetA,
    score_FeatherNetB,
    score_ResNet_GC
$mean\_score \leftarrow$ mean of scores[]
if $mean\_score > max\_threshold \;||\; mean\_score < min\_threshold$ **then**
    $final\_score \leftarrow mean\_score$
else if $score\_FishNet150\_1 < fish\_threshold$ **then**
    $final\_score \leftarrow score\_FishNet150\_1$
else if $score\_FeatherNetBForIR < IR\_threshold$ **then**
    $final\_score \leftarrow score\_FeatherNetBForIR$
else
    $mean\_score \leftarrow$ (6 * mean_score + score_FishNet150_1) / 7
    if $mean\_score > 0.5$ **then**
        $final\_score \leftarrow$ max of scores[]
    else
        $final\_score \leftarrow$ min of scores[]
    end if
end if

Acknowledgement

The authors would like to thank Intel IAGS SSP DSS Video and Audio team[4] members' great support and Web Platform Engineering team's help on Hardware devices support.

This work is supported in part by the National Natural Science Foundation of China under Grant No.61672254, 61672246, 61572221 and 61300222, Key project of National Natural Science Foundation of China Grant No U1536203, Natural Science Foundation of Hubei Province Grant No.2015CFB687, the Fundamental Research Funds for the Central Universities, HUST:2016YXMS088 and 2016YXMS018.

[3] http://chalearnlap.cvc.uab.es/workshop/32/description/

[4] https://01.org/linuxmedia

References

[1] Jukka Määttä, Abdenour Hadid, and Matti Pietikäinen. Face spoofing detection from single images using micro-texture analysis. In *2011 international joint conference on Biometrics (IJCB)*, pages 1–7. IEEE, 2011.

[2] Jukka Komulainen, Abdenour Hadid, and Matti Pietikäinen. Context based face anti-spoofing. In *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 1–8. IEEE, 2013.

[3] Tiago de Freitas Pereira, André Anjos, José Mario De Martino, and Sébastien Marcel. Can face anti-spoofing countermeasures work in a real world scenario? In *2013 international conference on biometrics (ICB)*, pages 1–8. IEEE, 2013.

[4] Zinelabidine Boulkenafet, Jukka Komulainen, and Abdenour Hadid. Face antispoofing using speeded-up robust features and fisher vector encoding. *IEEE Signal Processing Letters*, 24(2):141–145, 2017.

[5] Yousef Atoum, Yaojie Liu, Amin Jourabloo, and Xiaoming Liu. Face anti-spoofing using patch and depth-based cnns. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pages 319–328. IEEE, 2017.

[6] Litong Feng, Lai-Man Po, Yuming Li, Xuyuan Xu, Fang Yuan, Terence Chun-Ho Cheung, and Kwok-Wai Cheung. Integration of image quality and motion cues for face anti-spoofing: A neural network approach. *Journal of Visual Communication and Image Representation*, 38:451–460, 2016.

[7] Lei Li, Xiaoyi Feng, Zinelabidine Boulkenafet, Zhaoqiang Xia, Mingming Li, and Abdenour Hadid. An original face anti-spoofing approach using partial convolutional neural network. In *2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–6. IEEE, 2016.

[8] Keyurkumar Patel, Hu Han, and Anil K Jain. Cross-database face antispoofing with robust feature representation. In *Chinese Conference on Biometric Recognition*, pages 611–619. Springer, 2016.

[9] Shifeng Zhang, Xiaobo Wang, Ajian Liu, Chenxu Zhao, Jun Wan, Sergio Escalera, Hailin Shi, Zezheng Wang, and Stan Z Li. Casia-surf: A dataset and benchmark for large-scale multi-modal face anti-spoofing. *arXiv preprint arXiv:1812.00408*, 2018.

[10] Tiago de Freitas Pereira, André Anjos, José Mario De Martino, and Sébastien Marcel. Lbp- top based countermeasure against face spoofing attacks. In *Asian Conference on Computer Vision*, pages 121–132. Springer, 2012.

[11] Keyurkumar Patel, Hu Han, and Anil K Jain. Secure face unlock: Spoof detection on smartphones. *IEEE Transactions on Information Forensics and Security*, 11(10):2268–2283, 2016.

[12] Jianwei Yang, Zhen Lei, Shengcai Liao, and Stan Z Li. Face liveness detection with component dependent descriptor. In *2013 International Conference on Biometrics (ICB)*, pages 1–6. IEEE, 2013.

[13] Saptarshi Chakraborty and Dhrubajyoti Das. An overview of face liveness detection. *arXiv preprint arXiv:1405.2227*, 2014.

[14] Zezheng Wang, Chenxu Zhao, Yunxiao Qin, Qiusheng Zhou, and Zhen Lei. Exploiting temporal and depth information for multi-frame face anti-spoofing. *arXiv preprint arXiv:1811.05118*, 2018.

[15] Javier Hernandez-Ortega, Julian Fierrez, Aythami Morales, and Pedro Tome. Time analysis of pulse-based face anti-spoofing in visible and nir. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 544–552, 2018.

[16] Learning deep models for face anti-spoofing: Binary or auxiliary supervision. In *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 389–398, 2018.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[18] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[19] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[20] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 116–131, 2018.

[21] Ke Sun, Mingjie Li, Dong Liu, and Jingdong Wang. Igcv3: Interleaved low-rank group convolutions for efficient deep neural networks. *arXiv preprint arXiv:1806.00178*, 2018.

[22] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[23] Bichen Wu, Alvin Wan, Xiangyu Yue, Peter Jin, Sicheng Zhao, Noah Golmant, Amir Gholaminejad, Joseph Gonzalez, and Kurt Keutzer. Shift: A zero flop, zero parameter alternative to spatial convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9127–9135, 2018.

[24] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. *arXiv preprint arXiv:1801.07698*, 2018.

[25] Sheng Chen, Yang Liu, Xiang Gao, and Zhen Han. Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices. In *Chinese Conference on Biometric Recognition*, pages 428–438. Springer, 2018.

[26] Jonathan L Long, Ning Zhang, and Trevor Darrell. Do convnets learn correspondence? In *Advances in Neural Information Processing Systems*, pages 1601–1609, 2014.

[27] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *Advances in neural information processing systems*, pages 4898–4906, 2016.

[28] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[29] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[30] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[31] Junyuan Xie, Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, and Mu Li. Bag of tricks for image classification with convolutional neural networks. *arXiv preprint arXiv:1812.01187*, 2018.

[32] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[33] Zheng Qin, Zhaoning Zhang, Xiaotao Chen, Changjian Wang, and Yuxing Peng. Fd-mobilenet: Improved mobilenet with a fast downsampling strategy. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 1363–1367. IEEE, 2018.

[34] Chi Nhan Duong, Kha Gia Quach, Ngan Le, Nghia Nguyen, and Khoa Luu. Mobiface: A lightweight deep learning face recognition on mobile devices. *arXiv preprint arXiv:1811.11080*, 2018.

[35] Davis E King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10(Jul):1755–1758, 2009.

[36] Yao Feng, Fan Wu, Xiaohu Shao, Yanfeng Wang, and Xi Zhou. Joint 3d face reconstruction and dense alignment with position map regression network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 534–551, 2018.

[37] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[38] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[39] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.