Finger Knuckle Verification

Zhenyu, ZHOU

May 21, 2022

1 Back Propagation of Image Translation and Rotation

Because I just add rotation on the feature maps based on the soft shifted triplet loss function Liu and Kumar 2020. So my back propagation process is similar with it.

Because the loss is still use the triplet loss function, so the loss equation can be written as Equation 1.

$$Loss = \frac{1}{N}\sum_{i}^{N}[L(F(I_i^a), F(I_i^p)) - L(F(I_i^a), F(I_i^m)) + m]_+ \tag{1}$$

As for the $L()$ function, it is used to calculate the minimal value after shifting and rotating.

$$L(F_1, F_2) = \min_{-Ws \le w \le Ws, -Hs \le h \le Hs, -\Theta \le \theta \le \Theta} D_{w,h,\theta}(F_1, F_2) \tag{2}$$

And the $D()$ is to calculate the MSE of two input feature maps, just as show as below.

$$D_{w,h,\theta}(F_1, F_2) = \frac{1}{|C_{w,h,\theta}|}\sum_{(x,y)\in C_{w,h,\theta}}(F_1^{(w,h,\theta)}[x,y] - F_2[x,y])^2 \tag{3}$$

In terms of $C_{w,h,\theta}$, it represents the common region between two feature maps after shifting along x-axis with w, shifting along y-axis with h, and rotating with a. In this kind of situation, we assume that we shift and rotate anchor $w_{ap}, h_{ap}, \theta_{ap}$ with positive sample, and shift

and rotate anchor $w_{an}, h_{an}, \theta_{an}$ with negative sample can get minimal MSE. Then the partial differentiation of the loss function with respect to each variable can be written as follows:

$$\frac{\partial Loss}{\partial F_i^p} = \begin{cases} 0, if(x,y) \notin C_{w_{ap},h_{ap},\theta_{ap}} \ or \ Loss = 0 \\ \frac{-2(F_i^a[[x_{wap},y_{hap}]*M(\theta_{ap})]-F_i^p[x,y])}{N|C_{w_{ap},h_{ap},\theta_{ap}}|}, otherwise \end{cases} \tag{4}$$

The $M(\theta_{ap})$ is the rotation matrix.

$$\frac{\partial Loss}{\partial F_i^n} = \begin{cases} 0, if(x,y) \notin C_{w_{an},h_{an},\theta_{an}} \ or \ Loss = 0 \\ \frac{-2(F_i^a[[x_{wan},y_{han}]*M(\theta_{an})]-F_i^n[x,y])}{N|C_{w_{an},h_{an},a_{an}}|}, otherwise \end{cases} \tag{5}$$

As for the $F_i^a[x,y]$ derivation, because we shift and rotate the anchor in the above formula, we can inversely shift and rotate the positive and negative input feature.

$$\frac{\partial Loss}{\partial F_i^a[x,y]} = -\frac{\partial Loss}{\partial F_i^p[[x-w_{ap}, y-h_{ap}] * M(-\theta_{ap})]} + \frac{\partial Loss}{\partial F_i^n[[x-w_{an}, y-h_{an}] * M(-\theta_{an})]} \tag{6}$$

2 Experiments and Results

Because I want to follow the experiment protocol of FKNet, the FKNet just use on smaple of Index Finger Knuckle of Hand Dorsal Image Database to train due to classification model. However, my model uses triplet loss, so I need positive, negative and anchor samples. Firstly, all models are pre-trained on the Finger Knuckle V1 Database, and then fine-tuning on the corresponding finger knuckle database with bigger hard margin. On the Finger Knuckle V1 Database, it contains 512 subjects, and each of subjects offer 5 finger knuckle samples. In this kind of situation, I use three samples of these five samples to pre-train my models, meanwhile, as for the rest two samples as the validation dataset.

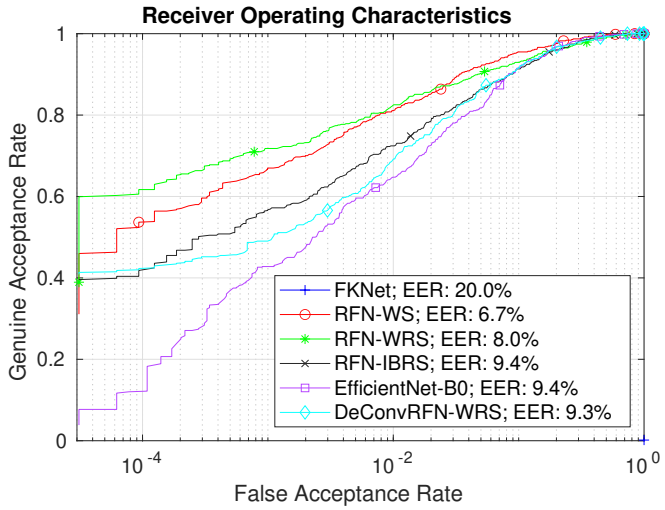
I also continue to train the DeConvRFN (change the RFN-128 convolution layer with deformable convolution) and EfficientNet, the loss converged to the local minimal point. As for the EfficientNet, I change

the original EfficientNet-B0 to fit my application. It has 9 stages in totally, the 9th stage is classification task with FC layer, so I replace it with convolution layer for output feature maps. Meanwhile, I delete the stage7 and stage8, and change stage3 and stage4 with stride 1. As for the RFNet, I use different loss to train the model, such as whole image rotation and shift, image blocks rotation and shift, and whole image shift. And I also compare the performance with the FKNet.

2.1 Within-Database Experiments

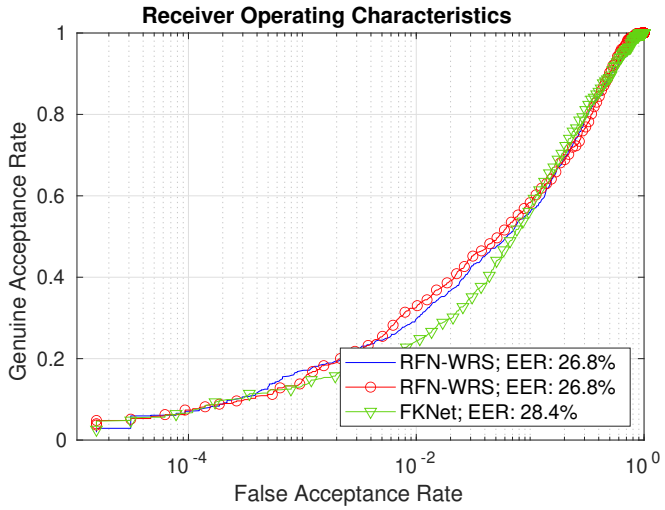
2.1.1 Finger Knuckle V3 Database with Deformation

The Finger Knuckle V3 Database have 1-104 subjects that have two session samples, and the rest subjects of first session 105-221 just offer one session samples. So as the first experiment, I firstly fine-tuned my model on the second session of 1-104 subjects, and test on the first session 1-104 subjects. So it will have $104 * 6 = 624$ genuine matching scores, and have $104 * 103 * 6 = 64272$ imposter matching scores. From the below figure, if the false accept rate is below 10^{-2}, the RFN-128-WRS is better than the RFN-128-WS. I also use the FKNet to train on this database, and the performance of FKNet is not better than the RFNet depend on the ROC figure.



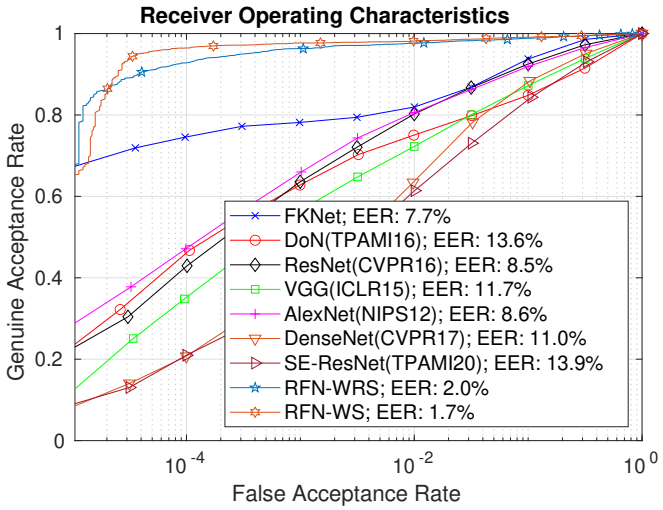
As for the two-session protocol on the database. I should fine-tune my model on the 105-221 subjects, and use two-session protocol to

evaluate my model performance on the 1-104 subjects dataset. In totally, it will generate $104 * 6 = 624$ genuine scores, and $104 * 103 * 6$ imposter scores. However, the FKNet is a classification task, and the output number classes should be same when training and testing. So the two session protocol experiment is not fit for FKNet. If the FKNet train on the 105-221 subjects and test on the 1-104 subjects with two sessions, the classes is different.



The two-session protocol will use the session1 as the probe and use the session2 as the enrollment. As for the genuine matching scores, each sample of a subject will choose the minimal matching score when compare to the rest samples. In this kind of situation, it will have $104x6$ genuine matching scores. Meanwhile, as for the imposter matching scores, it will also choose the minimal value result in $104 * 103 * 6$ imposter matching scores on the confusion matrix.

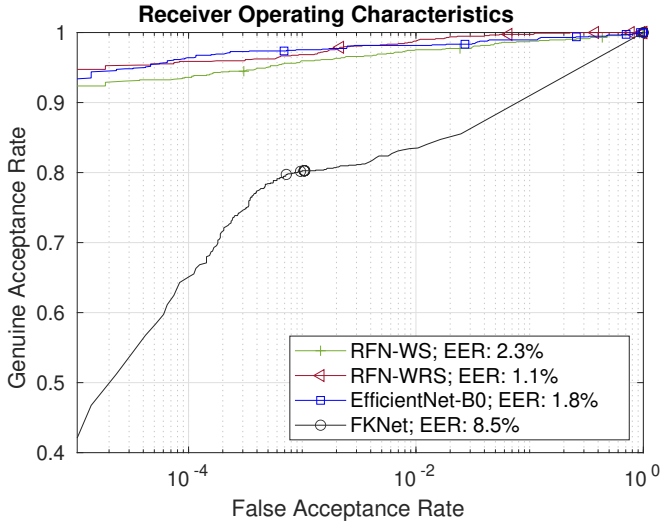
2.1.2 Index Finger Knuckle of Hand Dorsal Image Database



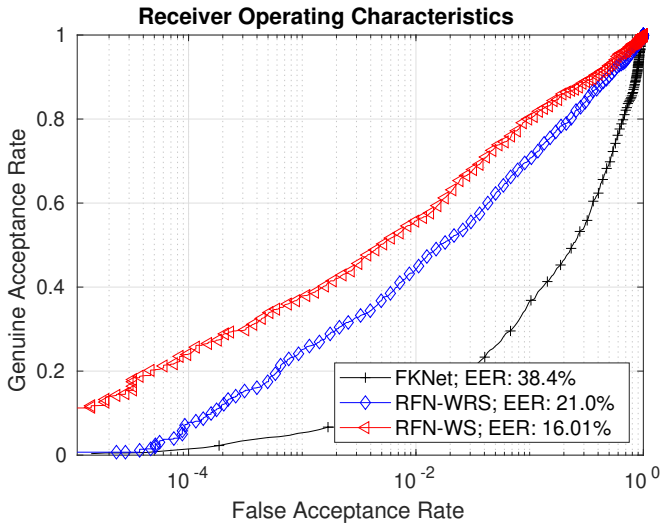
As for the experiment, the dataset totally contains 712 subjects, and I use the segmented Index finger knuckle as my dataset. And I fine-tuned my model on the first sample of each subject, and then use the rest four sample as the testing dataset. At the testing process, it has $712 * 4 = 2848$ genuine matching scores, and has $712 * 711 * 4 = 2024928$ imposter matching scores. The performance of RFN-128-WRS and RFN-128-WS is similar, but the RFN-128-WS is slightly better than RFN-128-WRS depend on the EER value. And we can get an information that the RFNet is better than the rest network in the ROC figure, including the FKNet.

2.1.3 2D Samples of 3D Finger Knuckle Database

First experiment on the database is to use the one session 190 subjects image to fine-tune models and then to test on the another session 190 subjects. It has $190 * 6$ genuine matching scores and $190 * 189 * 6$ imposter matching scores. From the result, we can see that these RFN-128-WRS, RFN-128-WS, EfficientNet can get very high matching accuracy. Meanwhile, the RFN-WRS has the minimal EER value among these models. As for the FKNet performance, it gets a very bad result on the 2D images of 3D finger knuckle. I think I have fully trained the FKNet. Maybe the model is overfitting on the training dataset.



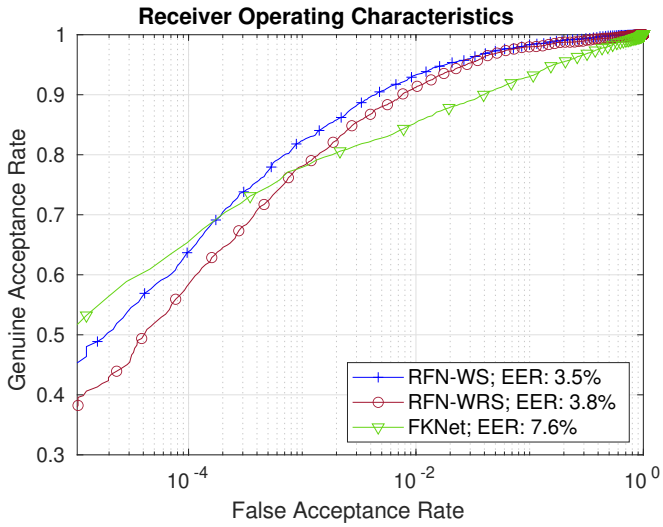
And then use the two session protocol. I use the rest samples of session1, and it has 191-228 subjects. In this kind of situation, the training dataset is too small. The two session protocol will test on the 190 subjects, these subjects can offer two session samples. Due to the training set is too small, so the matching performance is not very good. As for the FKNet, it cannot fit on two session protocol due to classification task.



2.2 Cross-Database Experiments

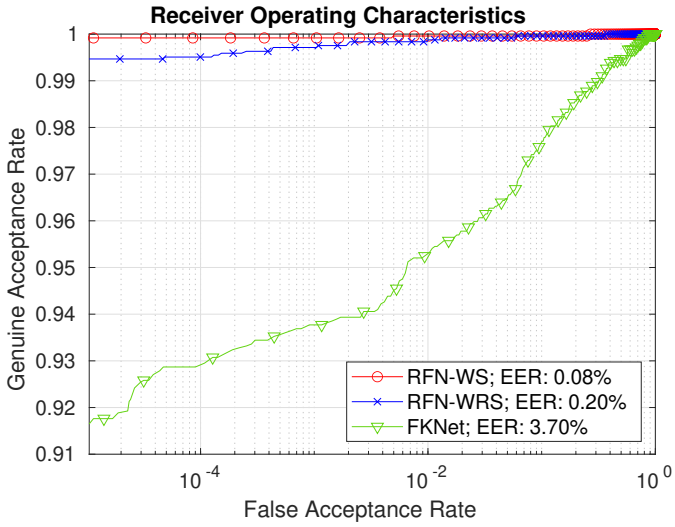
I firstly pre-trained my models on the Finger Knuckle V1 Database, and then fine-tuned models on the Finger Knuckle V3 Database (with deformable). I use these kind training method, and use these models to test performance on the Index Finger Knuckle of Hand Dorsal Image and Tsinghua Finger Knuckle Database as a cross database experiment. The label in the finger curve, the content in parentheses indicates the training samples. Such as RFN-WS(1-104), it uses 1-104 subjects of Finger Knuckle V3 Database to train models.

2.2.1 Index Finger Knuckle of Hand Dorsal Image



The database totally has 712 subjects, and each subject has 5 samples. Therefore, it will have 712∗5 genuine matching scores and 712∗711∗5 imposter matching scores. From the cure, the performance of RFN-WS and RFN-WRS is similar, and the RFN-WS is slightly better than RFN-WRS while using the same training samples.

2.2.2 Tsinghua Finger Knuckle Database



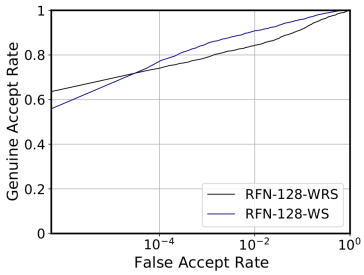
The database has 610 subjects, and each subject can offer 4 samples. Then as the cross database experiment, it will have $610 * 4$ genuine matching scores and $610 * 609 * 4$ imposter matching scores. In this database, all models can get very high matching performance from the table and figure.

2.3 Challenging Protocol Experiments

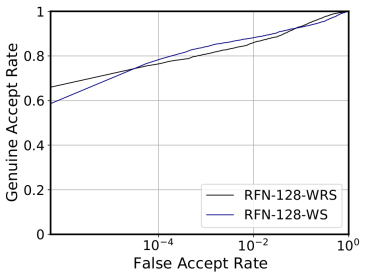
I have a challenging protocol experiment on the 2d image of 3d finger knuckle database. The experiment protocol is similar with the "Experiments Results using Challenging Protocol".

2.3.1 One-session

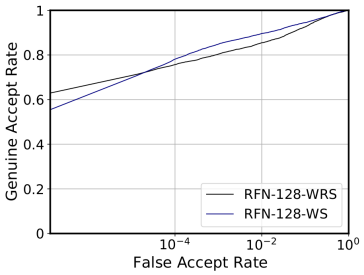
I pre-trained my model on Finger Knuckle V1 Database. As for the one-session experiment, I was firstly fine-tuning my model on the second session forefinger dataset which has 190 subjects, and then use the all-to-all protocol to evaluate matching accuracy on the first session of forefinger and middle finger, then combine both of them.



(a) Index Finger



(b) Middle Finger

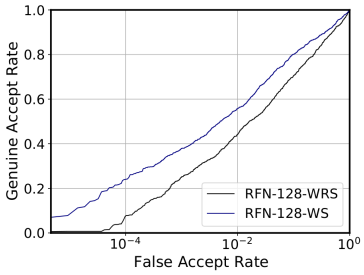


(c) Combine

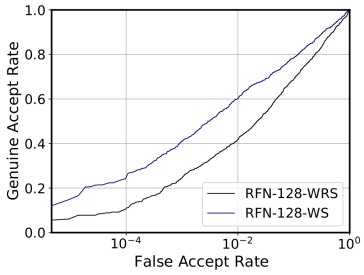
Finger	Model	Genuine	Imposter	EER
Combination	RFN-WRS	13680	7469280	0.08100
	RFN-WS	13680	7469280	0.06403
Forefinger	RFN-WRS	6840	1863226	0.08861
	RFN-WS	6840	1863226	0.05351
Middle	RFN-WRS	6840	1863226	0.07664
	RFN-WS	6840	1863226	0.07558

2.3.2 Two-session

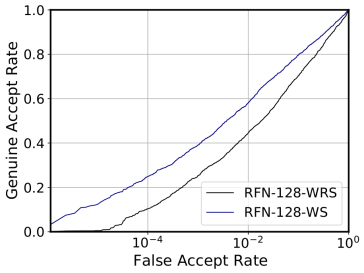
I also pre-trained model on Finger Knuckle V1 Database, and then trained model on the 191-228 subjects of first session of forefinger dataset. Using the rest 1-190 subjects samples as the testing dataset, I use the two session protocol to evaluate matching performance.



(a) Index Finger



(b) Middle Finger



(c) Combine

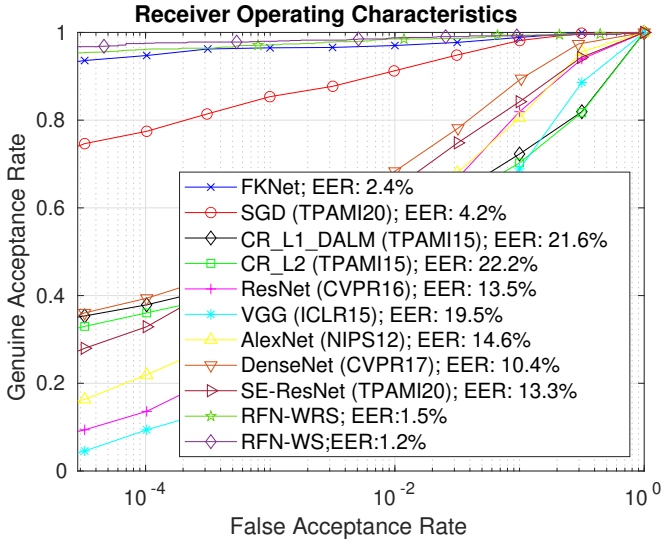
Finger	Model	Genuine	Imposter	EER
Combination	RFN-WRS	2280	864120	0.21138
	RFN-WS	2280	864120	0.16184
Forefinger	RFN-WRS	1140	215460	0.21053
	RFN-WS	1140	215460	0.15978
Middle	RFN-WRS	1140	215460	0.21603
	RFN-WS	1140	215460	0.16561

From the above result, the matching accuracy is very low result from the training dataset is too small. The training dataset just contains 191-228 subjects, 38 subjects. Meanwhile, the performance of forefinger, middle or combination is similar.

3 3D Finger Knuckle of 3D Finger Knuckle Database

I have use the matlab code that offered by the FKNet to generate the 3D finger knuckle images for getting the depth information. But it is

different that the input image size. The FKNet will resize the original image size $148 * 212$ to $70 * 100$ as the testing dataset, and crop from the $70 * 100$ to $48 * 80$ as the training dataset. As for RFNet, I just use the original image as the input data. Then the experiment protocol will generate $190 * 6$ genuine matching scores, and $190 * 189 * 6$ imposter matching scores. From the experiment result, we can get that the RFNet is the best model for the 3D Finger Knuckle Database.



4 References

Liu, Yang and Ajay Kumar (2020). "Contactless palmprint identification using deeply learned residual features". In: *IEEE Transactions on Biometrics, Behavior, and Identity Science* 2.2, pp. 172–181.