Completely Contactless and Online Finger Knuckle Identification

Zhenyu ZHOU

June 18, 2022

1 Whole Image Translated and Rotated Triplet Loss Function

Because our neural networks were trained using the architecture of triplet network [4], we used translated and rotated triplet loss function (TRTLoss) as loss function to update convolutional kernel of our models. When we match two finger knuckle in the contactless scenario, they will not only translate but also will rotate with local deformable transformation. For solving it, we add rotation based on translation to compose a more robust transformation. As for the TRTLoss function, we rotate feature maps while translate feature maps based on the Soft-Shifted Triplet Loss Function [3].

In generally, the TRTLoss is still a variant of triple loss, so that the TRTLoss can be written as a format of triple loss function as the Equation 1. As for the N, it means the number of triplet during training iteration, and $F(I^a)$ is the output feature map of input anchor image I^a through neural network. The hard margin parameter m can determine the distance between different class cluster by pushing them away.

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

We add translation and rotation transformation to deal with local deformable of finger knuckle. In order to adapt to tasks with different degrees of deformation, and balance performance and speed, we set translation and rotation ranges as a hyperparameter. The $L(F_1, F_2)$ will get the minimal distance of two feature maps $D_{w,h,\theta}(F_1, F_2)$ after translation and rotation in the range $-W \le w \le W, -H \le h \le H, -\Theta \le \theta \le \Theta$. Meanwhile, the distance $D_{w,h,\theta}(F_1, F_2)$ calculates the pixel-wise MSE value when feature map is translated w pixel along x-axis and h pixel along y-axis and rotated θ angle in the Equation 3.

$$L(F_1, F_2) = \underset{-W \le w \le W, -H \le h \le H, -\Theta \le \theta \le \Theta}{min} D_{w,h,\theta}(F_1, F_2) \tag{2}$$

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

In terms of $C_{w,h,\theta}$, it represents the common region between two feature maps after one feature map shifted along x-axis with w, shifted along y-axis with h, and rotated with θ.As for the (F_a, F_p) pair, we can assume when the F_a is rotated angle of θ_{ap} and shifted with (w_{ap}, h_{ap}) pixels can get the minimal $D_{w_{ap},h_{ap},\theta_{ap}}(F_a, F_p)$, then $L(F_a, F_p) = D_{w_{ap},h_{ap},\theta_{ap}}(F_a, F_p)$. Meanwhile, with the $(w_{an}, h_{an}, \theta_{an})$, the (F_a, F_n) pair can get the minimal $D_{w_{an},h_{an},\theta_{an}}(F_a, F_m)$.

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

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

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

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.

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

2 Experiments and Results

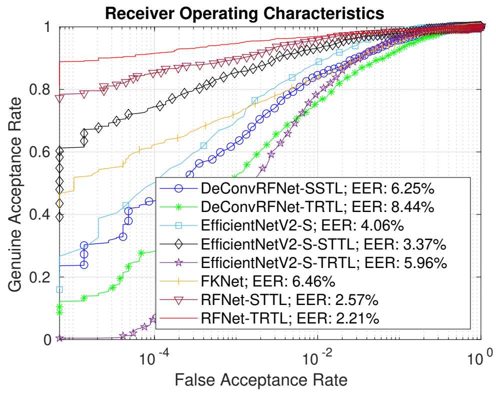
In this part, We will compare Translation and Rotation Triplet Loss (TRTL) and Soft-Translation Triplet Loss [3] performance on different finger knuckle dataset. Meanwhile, we will also compare with the state-of-the-art FKNet and EfficientNetV2. All the experiment will use the finger knuckle segmented by Yolov5 as training set and testing set, and in the Section 4, we will compare the impact of the finger knuckle segmented by YOLOV5 and the finger knuckle provided by these databases on the performance. As for the DeConvRFNet, we just change the RFNet convolution layer with deformable convolution layer. EfficientNetV2-S is the original classification model, we keep the same architecture and just change the FC layer of the head part to convolution layer for fitting TRTL and STTL. Because I want to follow the experiment protocol of FKNet, the FKNet just use one 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.

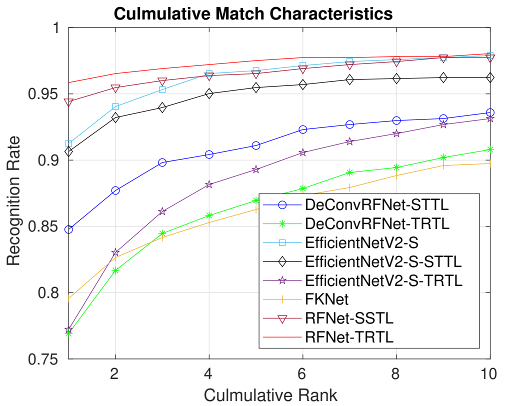
I also continue to train the DeConvRFN (change the RFN-128 convolution layer with deformable convolution) and EfficientNet, the loss converged to the local minimal point. As for the EfficientNet, I change the original EfficientNetV2-S to fit my application. It has 9 stages in totally, the 9th stage is 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.Meanwhile, for getting the original EfficientNetV2-S model performance, I also added corresponding experiments with EfficientNetV2-S. Because the EfficientNetV2-S model is a classification model, I calculate MSE of two feature vectors of last layer of model as matching score. And as for the input data, I follow the same method of FKNet, each image will rotate from $-10°$ to $10°$ to increase the amount of training data.

2.1 Within-Database Experiments

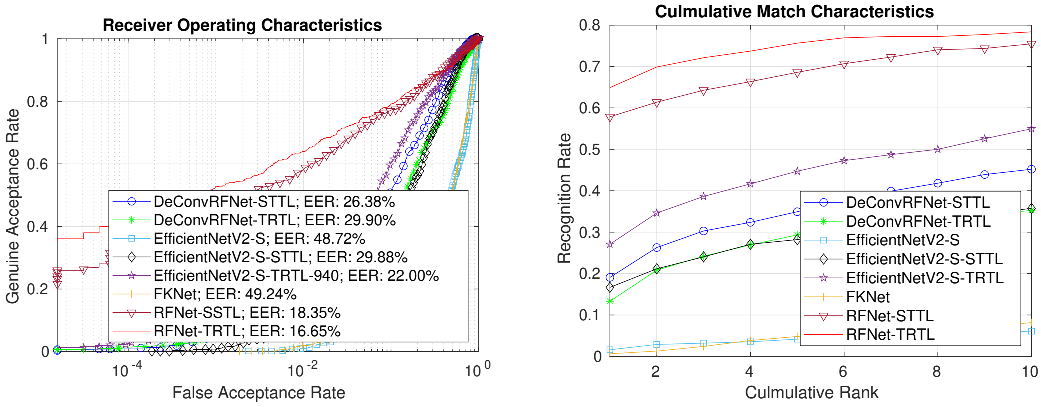
2.1.1 Finger Knuckle V3 Database with Deformation

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. From the CMC and ROC, each model with WRS is better than WS on this dataset. For the ROC curve, I add EfficientNetV2-S model performance.



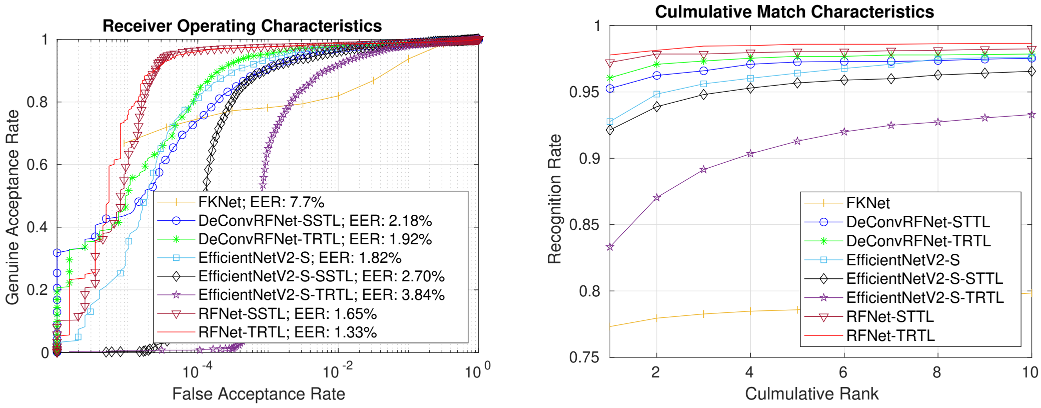


As for the two-session protocol on the database. I should fine-tune my model on the 105-221 subjects, and use two-session protocol to evaluate my model performance on the 1-104 subjects dataset. In totally, it will generate $104 * 6 = 624$ genuine scores, and $104 * 103 * 6$ imposter scores. However, 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

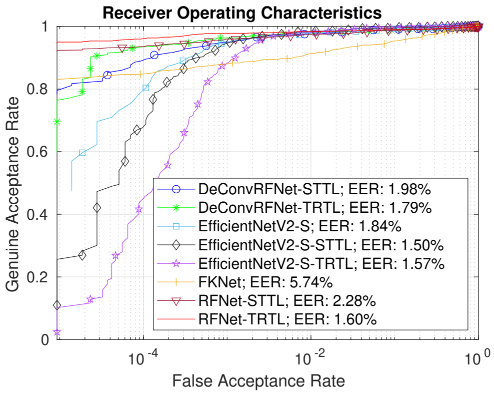


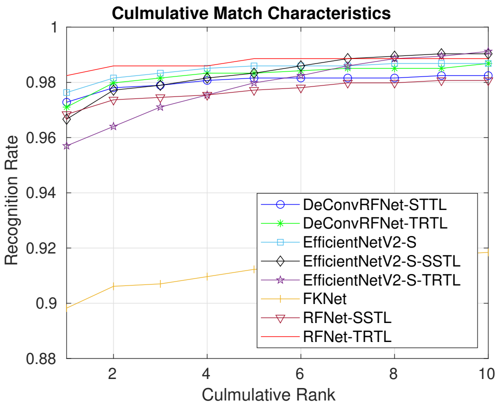
For the ROC curve, I add EfficientNetV2-S model performance. Update CMC Curve and ROC Curve with EfficientNet, DeConvRFNet and RFNet. As for the experiment, the dataset totally contains 712 subjects, and I use the segmented Index finger knuckle as my dataset. And I fine-tuned my model on the first sample 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, EfficientNetV2 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. Updated ROC Curve and CMC Curve with RFNet, EfficientNet and DeConvRFNet. And each model with WRS is better than WS. For the ROC curve, I add EfficientNetV2-S model performance.



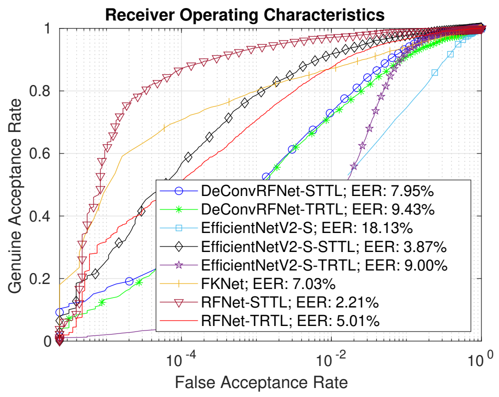


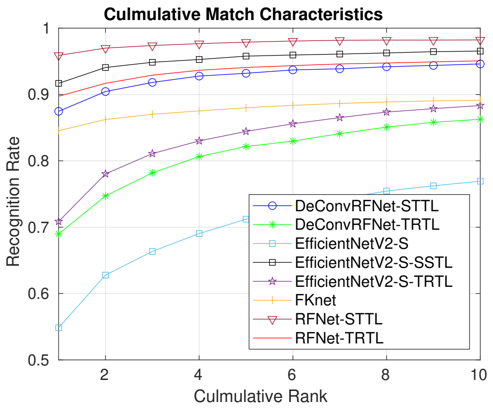
And then use the two session protocol. I use the rest samples of session1, and it has 191-228 subjects. In this kind of situation, the training dataset is too small. The two session protocol will test on the 190 subjects, these subjects can offer two session samples. Due to the training set is too small, 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. Updated ROC Curve and CMC Curve with RFNet, EfficientNet and DeConvRFNet. For the ROC curve, I add EfficientNetV2-S model performance.

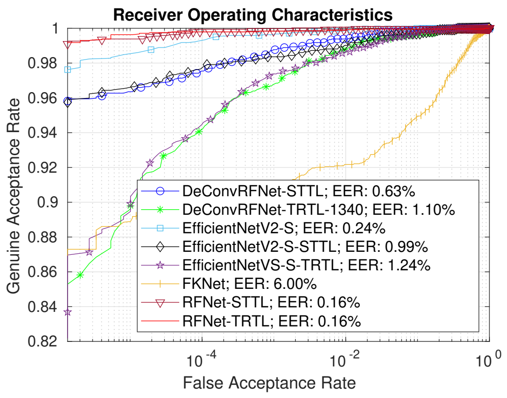
2.2.1 Index Finger Knuckle of Hand Dorsal Image

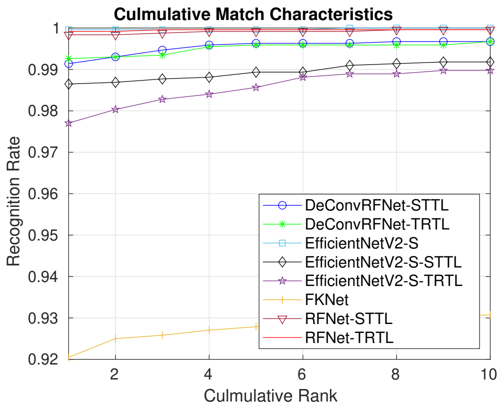




The database totally has 712 subjects, and each subject has 5 samples. Therefore, it will have $712 * 5$ genuine matching scores and $712 * 711 * 5$ imposter matching scores. From the cure, the performance of RFN-WS and RFN-WRS is similar, and the RFN-WS is slightly better than RFN-WRS while using the same training samples. Updated ROC Curve and CMC Curve with RFNet, EfficientNet and DeConvRFNet. For the ROC curve, I add EfficientNetV2-S model performance.

2.2.2 Tsinghua Finger Knuckle Database



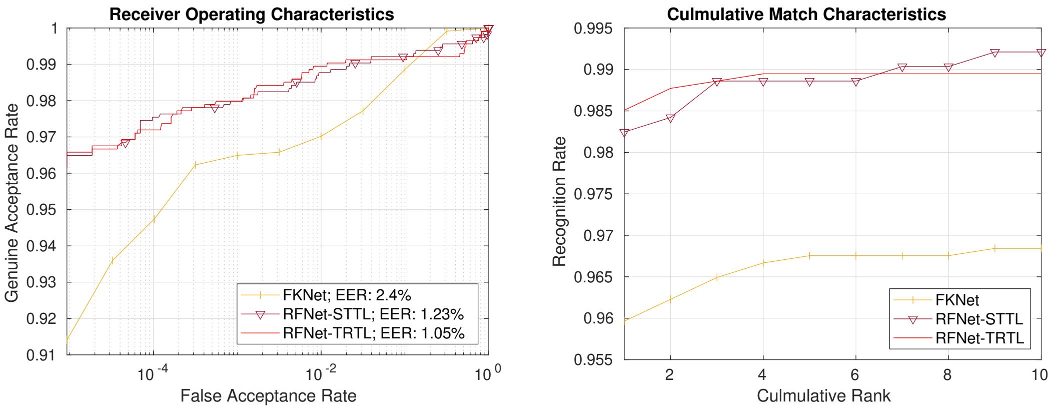


The database has 610 subjects, and each subject can offer 4 samples. Then as the cross database experiment, it will have $610 * 4$ genuine matching scores and $610 * 609 * 4$ imposter matching scores. In this database, all models can get very high matching performance from the table and figure.

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 Compare Performance on Yolov5 Segment and Dataset Offter Finger Knuckle

This section aim to compare quality of finger knuckle between Yolov5 segmented and dataset offered. Because the segmented finger knuckle on the 3D Finger Knuckle Dataset already have high quality, I mainly test on the Index Finger Knuckle of Hand Dorsal Dataset and the Finger Knuckle Dataset V3 (with deformable).

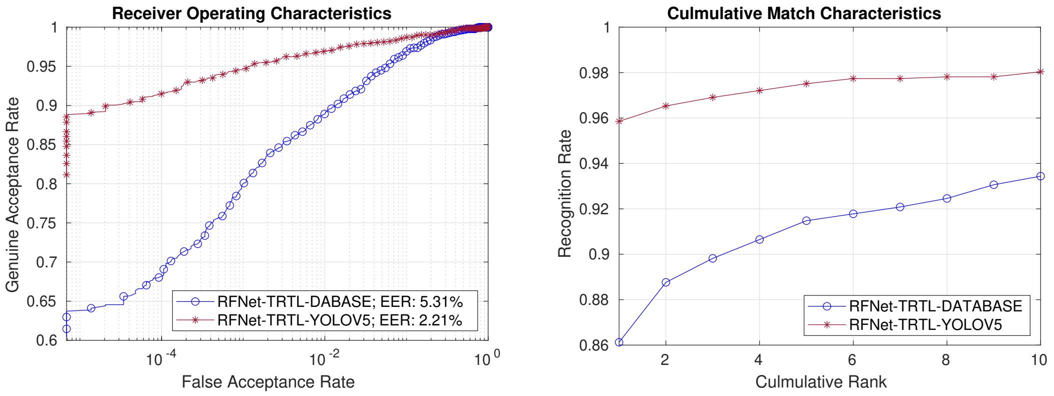


Figure 8: Compare performance on the Finger Knucle V3 Dataset (with deformable)

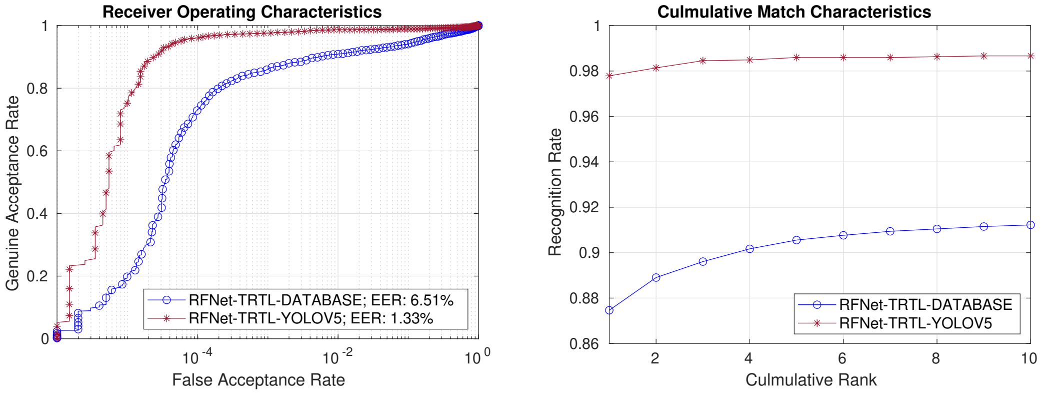


Figure 9: Compare performance on the Index Finger of the Hand Dorsal Image Database.

From the above figures, we can clearly get the conclusion that quality of segmented finger knuckle of YOLOV5 is better than the segmented finger knuckle of dataset based on the same model. Especially on the Finger Knuckle V3 Dataset, the EER value can drop from 5.31% to 2.21%. Because in the previous experiments, except for Finger Knuckle V3 and 3D Finger Knuckle Dataset, other datasets were segmented using the YOLOV5 model. But with Yolov5 segmented finger knuckle, the RFNet performance is slightly higer than the local feature descriptors based on key points matching [1], and performance higher than the paper [2]. If we want to compare different method performance, I think we should use same dataset. In this kind of situation, the method of [1] maybe will ger higer performance on the segmented finger knuckle by YOLOV5.

5 References

[1] Ajay Kumar. “Contactless finger knuckle authentication under severe pose deformations”. In: *2020 8th International Workshop on Biometrics and Forensics (IWBF)*. IEEE. 2020, pp. 1–6.

[2] Ajay Kumar. “Toward pose invariant and completely contactless finger knuckle recognition”. In: *IEEE Transactions on Biometrics, Behavior, and Identity Science* 1.3 (2019), pp. 201–209.

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

[4] Florian Schroff, Dmitry Kalenichenko, and James Philbin. “Facenet: A unified embedding for face recognition and clustering”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 815–823.