Completely Contactless and Online Finger Knuckle Identification

Zhenyu ZHOU

June 29, 2022

1 Abstract

2 Introduction

3 Matching Contactless Finger Knuckle

One of our contributions is the online finger knuckle identification. In this kind of situation, we choose the RFNet [13] as our feature extraction backbone, because the model not only is lightweight enough, but it achieves state-of-the-art performance on the palmprint dataset. Meanwhile, the paper [13] uses the soft-shifted triplet loss function, called SSTL to train the model and matching two features for dealing with translation problem. However, in generally, feature maps of the same class will not only just shift along two axes, but also will have local deformable transformation. For solving it, we propose a new loss and also a new matching method, called translation and rotation triplet loss function (TRTL). With the TRTL, the feature maps can be translated along the x-axis and y-axis, and can be rotated clockwise and counterclockwise. Then we will get the minimal value after translation and rotation as the similarity scores.

3.1 Translated and Rotated Triplet Loss Function

As for a new loss function, the most important point is whether it can be differentiable. With a differentiable loss, the back propagation process can proceed smoothly, and the learnable parameters can be updated to get the minimal loss. In this section, we will discuss the derivation of the TRTL loss function. Because our neural networks were trained using the architecture of triplet network [18], we used TRTL as loss function to update convolutional kernel of our models.

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 batch size during training iteration, and $T(I^a)$ is the output template 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 during training process.

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

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 $L(T_1, T_2)$ will get the minimal distance of two templates $D_{w,h,\theta}(T_1, T_2)$ after translation and rotation in the range $-W \leq w \leq W, -H \leq h \leq H, -\Theta \leq \theta \leq \Theta$. Meanwhile, the distance $D_{w,h,\theta}(T_1, T_2)$ calculates the pixel-wise MSE value when template T_1 is translated w pixel along x-axis and h pixel along y-axis and rotated θ angle in the Equation 3.

$$L(T_1, T_2) = \underset{-W \leq w \leq W, -H \leq h \leq H, -\Theta \leq \theta \leq \Theta}{min} D_{w,h,\theta}(T_1, T_2) \tag{2}$$

$$D_{w,h,\theta}(T_1, T_2) = \frac{1}{|C_{w,h,\theta}|}\sum_{(x,y)\in C_{w,h,\theta}}(T_1^{(w,h,\theta)}[x,y] - T_2[x,y])^2 \tag{3}$$

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

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

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

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

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

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

4 Experiments and Results

We choose the baseline model is the RFNet [13], its performance can outperform DenseNet-BC [9], CompCode [11], DoN [33], Ordinal Code [20], and RLOC [10] method on the palmprint verification. For proving TRTL loss function performance, we will compare its performance with Soft-Shift Triplet (STTL)[13] loss function on different public finger knuckle database based on the RFNet [13]. With TRTL loss function, the RFNet is represented by RFNet-TRTL, on the country, RFNet-STTL represents with STTL loss function. Compare to convolution layer or dilated convolution [31], the deformable convolution [34] can solve local deformable by sampling different location and different weight. We also replace the RFNet convolution layer with deformable convolution layer called DeConvRFNet. As for the RFNet and DeConvRFNet, we will firstly pretrain on the HKPolyU Finger Knuckle Images Database (V1.0) [23] as the pretrained weights.

Meanwhile, we will also compare with the FKNet [3] which get the state-of-the-art performance on 3D finger knuckle identification, and EfficientNetV2-S [21]. FKNet performance on the 3D finger knuckle database, 2D finger knuckle and even palmprint database can over SGD [2], CR_L1_DALM, CR_L2 [32], ResNet-50 [6], VGG-16 [19], AlexNet [12], DenseNet-121 [9], and SE-ResNet-50 [8]. Both of FKNet and EfficientNetV2-S are classification neural network. As a classification neural network, it commonly has a problem when the number of classes of testing dataset is not as same as the training set classes, result in fine-tuning on the testing set. Therefore, we use the vector before soft-max layer as the feature vector, and then calculate the MSE of two feature vectors as the similarity score during matching finger knuckle. We use the ResNet-50 pretrained weights as the FKNet initial weights, and use the pretrained weights on the ImageNet21K as the initial weights of EfficientNetV2-S.

We also want to show the performance of TRTL and SSTL on the EfficientNetV2-S model, therefore we keep the same architecture and just change the FC layer of the head part with convolution layer for fitting TRTL and STTL. The changed EfficientNetV2-S model with TRTL called EfficientNetV2-S-TRTL, and with STTL called EfficientNetV2-S-STTL. As same as the EfficientNetV2-S model, we also use the pretrained model weights on the ImageNet21K dataset. In generally, public finger knuckle database already offer segmented finger knuckle images, but we use our YOLOv5-CSL model to segment finger knuckle as our training and testing data during our experiment.

4.1 Model Complexity Analysis

As a completely contactless and online finger knuckle identification, we must choose a model that can meet the requirements of matching speed while ensuring matching accuracy, and even sacrifice matching accuracy for a certain matching speed. We have listed learnable weights of each model, and the corresponding feature extraction time and matching time.

Model	Prams (M)	Input Size	FLOPs (B)	Feature Extraction (s)	Matching (s)
DeConvRFNet-STTL	0.36M	128x128	1.29B		
DeConvRFNet-TRTL	0.36M	128x128	1.29B		
EfficientNetV2-S [21]	20.18M	300x300	5.40B		
EfficientNetV2-S-STTL	20.00M	300x300	5.38B		
EfficientNetV2-S-TRTL	20.00M	300x300	5.38B		
FKNet [3]	7.28M	96x64	0.28B		
RFNet-STTL [13]	0.46M	128x128	1.39B	0.0062s	0.049s
RFNet-TRTL	0.46M	128x128	1.39B	0.0062s	

Table 1: Model complexity analysis

......

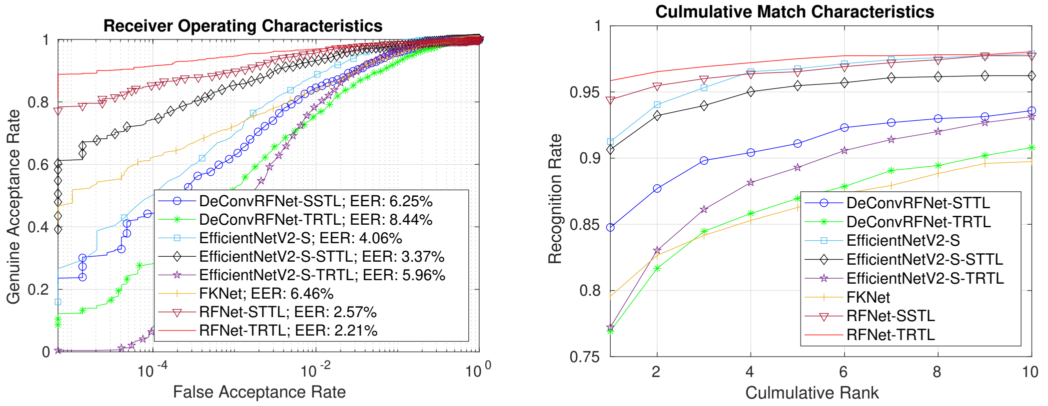
4.2 Within Database Performance Evaluation

4.2.1 Contactless Finger Knuckle Image Database (Version 3.0)

The finger knuckle database [24] can offer contactless finger knuckle of 221 subjects, but only 104 subjects have second session samples. For each session, each subject can offer 6 samples. It is worth mentioning that the finger knuckle sample provided by this database is more challenging and closer to real world scenarios, because the finger knuckle will bend from 0 to 90 degree result in crease deformation.

One-Session Protocol

As for the one-session protocol, I firstly fine-tuned models on the second session 104 subjects dataset, totally $104 * 6 = 624$ images as the testing set. Then use the first session 221 subjects as the testing set result in $221 * 6 = 1326$ genuine matching scores and $221 * 220 * 6 = 291720$ imposter matching scores. From the Figure 1, we can easily find the RFNet is the best model not only on the ROC but also on the CMC. In terms of the baseline model RFNet, our loss function TRTL can improve the matching accuracy when compare to the STTL loss function. Although the finger knuckle of the database with deformation while bend from 0 to 90 degree, the EER of the RFNet-TRTL can arrive at 2.21%. And as top-2 ranking, the RFNet-TRTL recognition rate is about 0.97 on the CMC. As for the rest model, EfficientNetV2-S model performance is better than FKNet and DeConvRFNet. From the performance result, if we just change the convolution layer with deformable convolution, it cannot overcome finger knuckle deformation, even the performance is dropped.

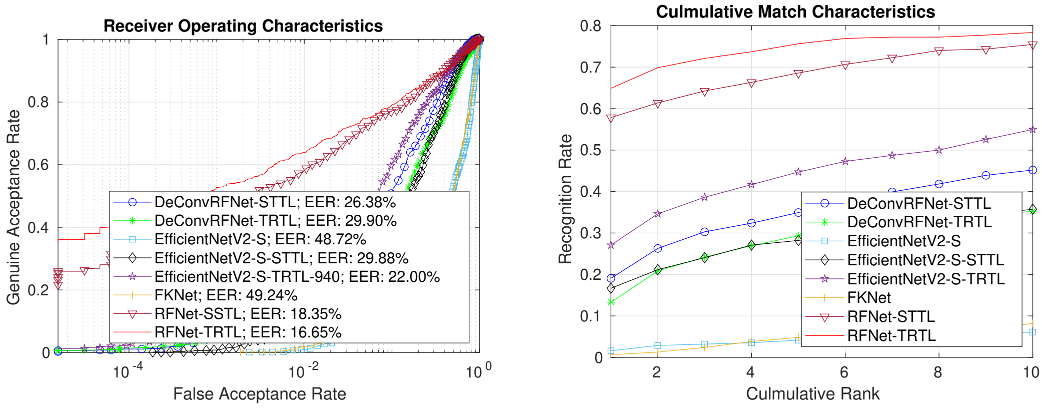


(a) (b)

Figure 1: Comparative ROC (a) and corresponding CMC (b) for one-session on the contactless finger knuckle image database [24].

Two-Session Protocol

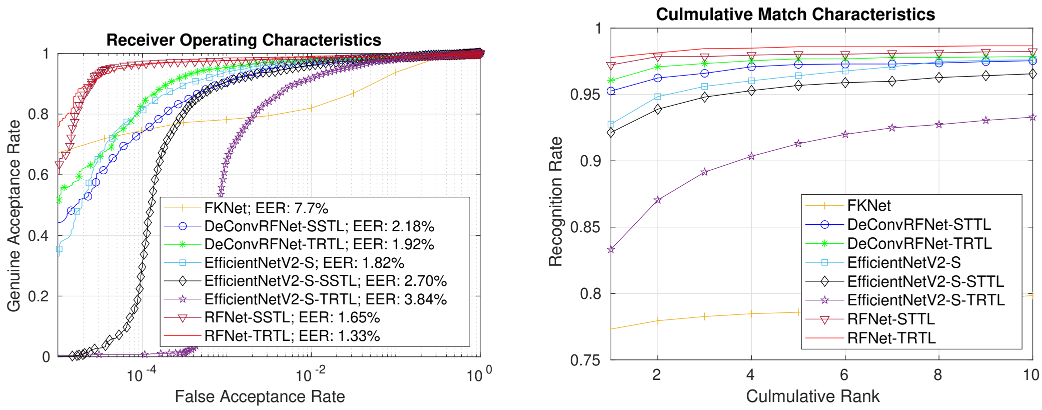
We fine-tune models on the first session subjects who don't provide second session samples, and use two-session protocol to evaluate my model performance on the first session subjects who can offer two-session data. In totally, it will generate $104 * 6 = 624$ genuine scores, and $104 * 103 * 6$ imposter scores. Just like said before, the FKNet and EfficientNetV2-S are classification networks, we use output feature vector to calculate MSE as the matching score. Because the degree of deformation vary on the two-session data, the verification and identification scenarios is more complexity than one-session protocol. Due to these factors, the accuracy on the two-session protocol is much lower than the one-session protocol. However, the RFNet is still the best model, even its EER is half of the EER of other models. Meanwhile, our TRTL loss function still work better than the STTL loss function, with 16.65% and 18.35% respectively on the ROC. As for the CMC, when the cumulative rank value is 2, recognition rate of RFNet-TRTL can arrive at 0.7. From the ROC and CMC Figure 2, we can also get that the STTL and TRTL triplet loss function are better than classification task, because the FKNet and EfficientNetV2-S have the lowest accuracy.



(a) (b)

Figure 2: Comparative ROC (a) and corresponding CMC (b) for two-session on the contactless finger knuckle image database [24].

4.2.2 Index Finger Knuckle of Contactless Hand Dorsal Image Database



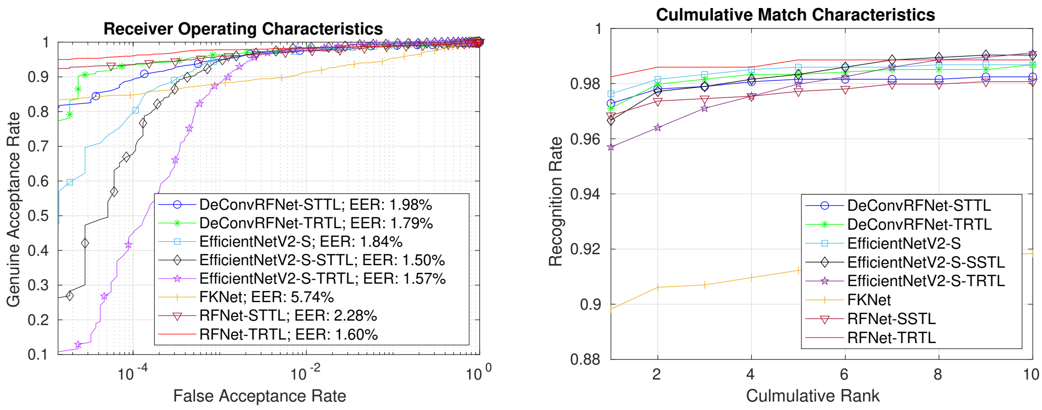
(a) (b)

Figure 3: Comparative ROC (a) and corresponding CMC (b) for one-session on the contactless hand dorsal image database [25].

As for the experiment, the dataset [25] totally contains 712 subjects, and each subject have 5 finger knuckle samples. And we fine-tuned our models on the first sample of each subject, and then use the rest four sample as the testing dataset. For protocol on the database, we use protocol as same as protocol of the FKNet [3]. At the evaluation process, it has $712 * 4 = 2848$ genuine matching scores, and has $712 * 711 * 4 = 2024928$ imposter matching scores. The performance of RFNet-TRTL and RFNet-STTL is similar, but the RFNet-TRTL is slightly better than RFNet-STTL depend on the EER value on ROC. And on the CMC, the RFNet-TRTL still get the best accuracy. We can notice that the FKNet get the worst result when compare to other models. The EfficientNetV2-S model is still better than the FKNet, because EfficientNetV2-S is deeper than FKNet with MBConv block. MBConv block is more robust than the original residual block.

4.2.3 2D Forefinger of 3D Finger Knuckle Database

The HKPolyU 3D Finger Knuckle Images Database [22] can offer reliable 3D finger knuckle pattern (surface normal vector, depth, or curvature) from 2D finger knuckle images, therefore we use its 2D images as our evaluation database. 190 subjects of the database have two-session finger knuckle samples, and 38 subjects offer one-session images. In this kind of situation, two-session protocol is not fit on the database, then we use one-session protocol to evaluate performance. We use the first session 190 subjects images to fine-tune models and then to test on the second session 190 subjects. It has $190 * 6 = 1140$ genuine matching scores and $190 * 189 * 6 = 215460$ imposter matching scores. From the ROC and CMC, we can get a conclusion that the performance of RFNet, DeConvRFNet, and EfficientNetV2-S are similar. However, the FKNet is still the worst one, which EER is 5.74% and the CMC is lower than others. The unchanged thing is that the RFNet with TRTL loss still get the best performance with 1.60% EER, even for the CMC.



(a) (b)

Figure 4: Comparative ROC (a) and corresponding CMC (b) for one-session on the 3D finger knuckle database[22].

4.3 Cross Database Performance Evaluation

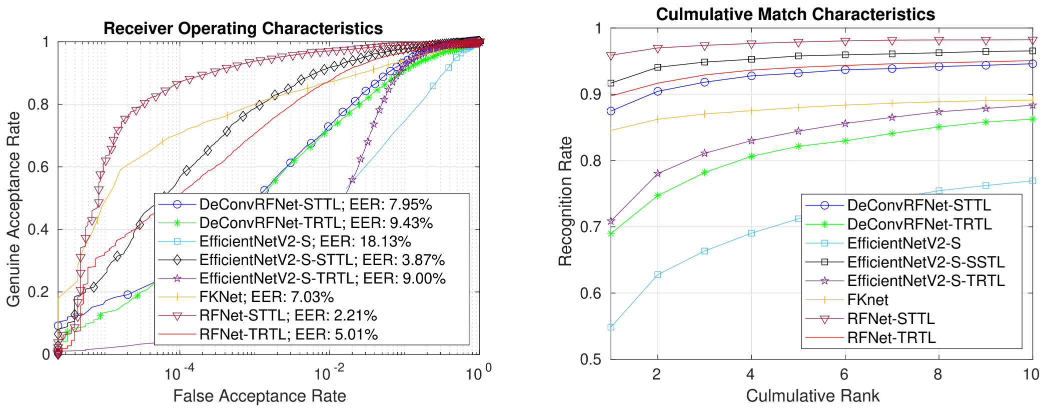
From the within database experiment, we can clearly get a conclusion that the TRTL loss function can increase the performance compare to the STTL loss function, and the RFNet is better than the DeConvRFNet, EfficientNetV2-S, and FKNet. Meanwhile, the FKNet performance is the worst one. In this section, we will compare these models' performance on the cross database experiment. For these cross database experiment, it can get the generalization ability of neural network, because these data can be regard as unseen data.

As for the cross database experiment, I firstly pre-trained our models on the Finger Knuckle Images Database V1, and then fine-tuned models on the Finger Knuckle Images Database V3 (with deformable). In the next step, we use these model to test all the finger knuckle of the

Hand Dorsal Images Database and the Tsinghua Finger Vein and Finger Dorsal Texture Database (THU-FVFDT3) [26]. Although the THU-FVFDT3 database can offer two-session samples with interval several seconds, but strictly speaking, it is not two-session database. Therefore, I just use the training set of the database as our testing dataset.

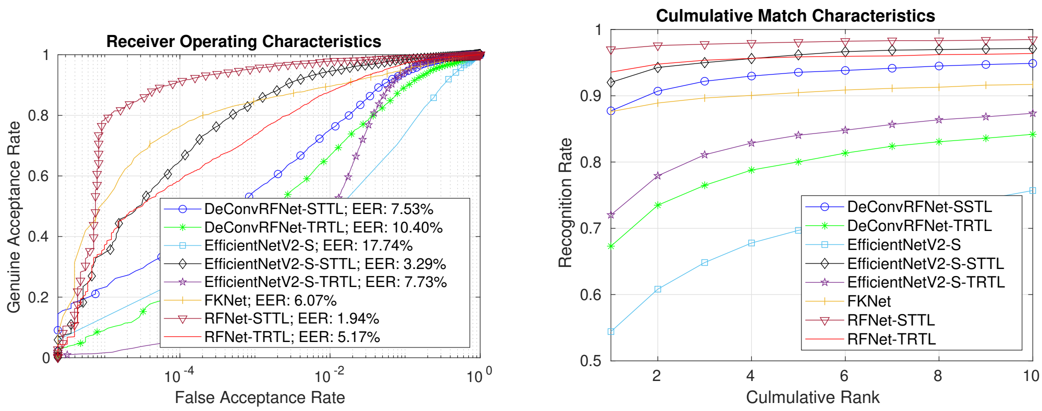
4.3.1 Hand Dorsal Images Database

Index Finger Knuckle and Middle Finger Knuckle



(a) (b)

Figure 5: Comparative ROC (a) and corresponding CMC (b) for one-session of the index finger knuckle on the contactless hand dorsal image database [25].

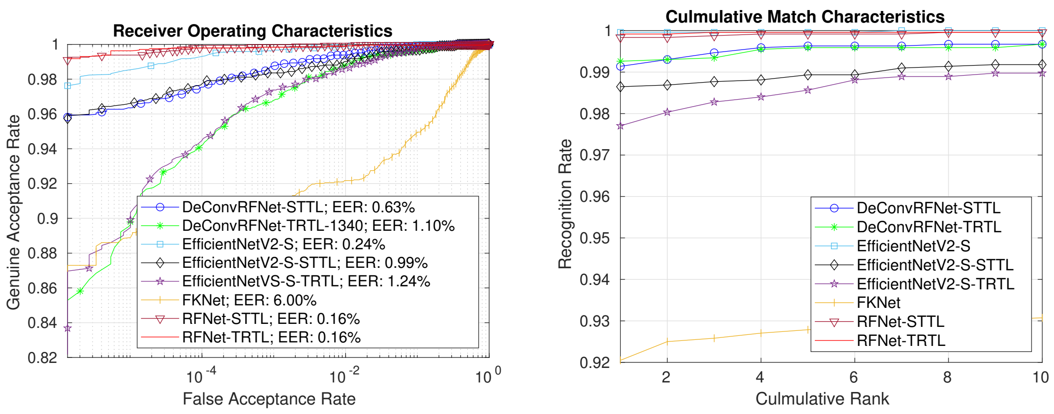


(a) (b)

Figure 6: Comparative ROC (a) and corresponding CMC (b) for one-session of the middle finger knuckle on the contactless hand dorsal image database [25].

The database totally has 712 subjects, and each subject has 5 samples. Therefore, it will have $712 * 5 = 3560$ genuine matching scores and $712 * 711 * 5 = 2531160$ imposter matching scores for index and middle finger knuckle. Figure 5 is the performance result on the index finger, and Figure 6 is the performance result on the middle finger knuckle. From Figure 5 and Figure 6, all models' cross database performance is similar on the database regardless which finger. We should also notice that STTL is better than TRTL on the cross database experiment, while within database, the TRTL is better than STTL. It shows that the generalization ability of TRTL is not better than STTL to some extent. However, the RFNet-STTL outperform the rest models depend on the ROC and CMC. Even better than FKNet and EfficientNetV2-S, both of them are classification models.

4.3.2 Tsinghua Finger Vein and Finger Dorsal Texture Database



(a) (b)

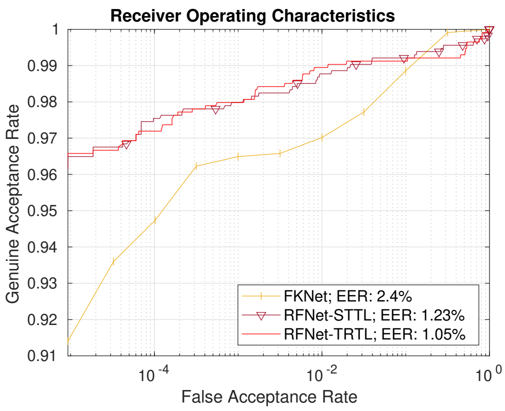
Figure 7: Comparative ROC (a) and corresponding CMC (b) for one-session of the finger dorsal texture images [26].

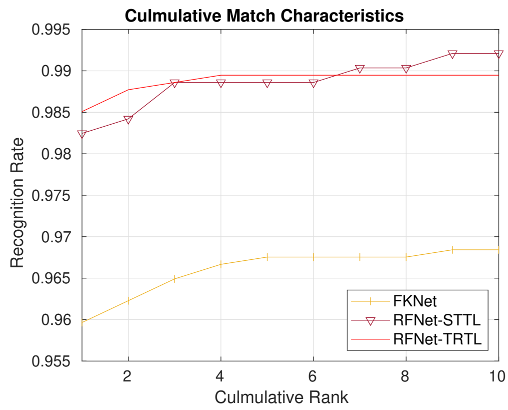
The database [26] has 610 subjects, and each subject can offer 4 samples. From the finger dorsal texture images, we can use our YOLOv5-CSL model to segment finger knuckle images as our testing set. Then as the cross database experiment, it will have $610*4 = 2440$ genuine matching scores and $610 * 609 * 4 = 1485960$ imposter matching scores. In this database, all models can get very high matching performance from the Figure 7, even the worst FKNet can arrive at 6.00% EER on the database. The RFNet with TRTL and STTL get the same accuracy, in terms of the CMC, the recognition rate almost arrive at 100%.

4.4 3D Finger Knuckle Images Database

Because the RFNet with TRTL and STTL can get the best performance on the within database experiment and cross database experiment,

I have used 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.





4.5 Discussion

From these experiment results, we can see that EfficientV2-S model is better than FKNet in some dataset. Because EfficientNetV2 model use MBConv as a block unit for replacing residual block. As for MBConv block, it uses depth-wise convolutions to decrease training weights and use Squeeze-Excited block as channel transformer. Meanwhile, the depth of EfficientNetV2-S is deeper than the FKNet.

There is another conclusion is that TRTL generalization ability is lower than STTL loss from the cross database experiment. But in the within database experiment, these model with TRTL loss is better than STTL loss.

....

5 Ablation Study

6 Online Contactless Finger Knuckle Identification

With TRTL loss, the RFNet [13] can outperform state-of-the-art methods. In the previous section, we have estimated its verification and identification performance on different public finger knuckle database, including within-db and cross-db experiments. As for a completely contactless and online finger knuckle identification, the finger knuckle detector is a very important module for automatically detect and segment finger knuckle region. However, as for traditional segmentation algorithm, they cannot correctly segment the finger knuckles in the presence of complex background interference, multiple finger knuckles in the same field of view, obscured finger knuckles or bent finger knuckles. Meanwhile, as for neural network, the current based on YOLO [16], [14], [15], [1], [27] and R-CNN [5], [4], [17], [7] series object detection and segmentation approaches cannot simultaneously obtain the angle of finger knuckle and the segmentation with high precision. Especially, the angle of the finger knuckle is a vital factor for identification. If we can get the angle of finger knuckle, we can use angle information to align two feature maps for increasing matching accuracy and efficiency. For solving above problems, we propose rotated bounding box detection based on YOLOv5 model for segmenting and getting angle information.

6.1 Contactless Finger Knuckle Detection

6.1.1 Rotated Bounding Box Based on YOLOv5

In order to solve the problem of finger knuckle detection in the real world, we choose to use YOLOv5 model because the YOLO series is famous for its fast detection speed and high accuracy. Especially, the YOLOv5's [27] speed can meet our online detection requirements.

Rotated Bounding Box

However, the YOLOv5 just detect horizontal bounding boxes which cannot offer angle information and will segment a lot of background information. In order to solve these above problem, a rotated bounding box will be predicted instead of horizontal bounding box. As analyzed in this paper [30], the rotated bounding boxes loss will mainly come from angular periodicity and the exchangeability of edges. When use the long side definition of rotated bounding box, it can deal with the exchangeability of edges problem. Meanwhile, using classification task to predict angle can make model easier to train. A periodic coding method called Circular Smooth Label (CSL) [30] soft coding can also solve the problem that One-Hot cannot distinguish class relationship. Formula 7 $g(x)$ is the window function to smooth One-Hot label, and r is a window function of the radius.

$$CSL(x) = \begin{cases} g(x), & \theta - r < x < r + \theta \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

Furthermore, in this paper, we used the Gaussian function for the Equation 7 window function, a commonly available function, and used a window radius of 6 to smooth the labels.

Loss function

The original YOLOv5 loss function can have three components. The formula can be simply written as $Loss = CIOU\_Loss + Loss_{conf} + Loss_{class}$. Since the rotated bounding box is based on the modification of YOLOv5, only the angle classification loss is added more. So the total loss function is as expressed in Equation 8, with the addition of $Loss_{angle}$ to YOLOv5 loss function.

$$Loss = CIOU\_Loss + Loss_{conf} + Loss_{class} + Loss_{angle} \tag{8}$$

$$\begin{aligned} Loss_{angle} = \sum_{i=0}^{S^2} I_{ij}^{obj} \sum_{a \in [0,180)} & [\hat{P_i(a)} log(P_i(a)) + \\ & (1 - \hat{P_i(a)}) log(1 - P_i(a))] \end{aligned} \tag{9}$$

6.1.2 Contactless Finger Knuckle Dataset

Our task is to detect finger knuckles in the contactless and online scenario, but by understanding current public finger knuckle database, their data are collected at specific conditions such as certain angle, certain light. In this kind of situation, this kind of data cannot represent real images of finger knuckle in real world. In order to address the shortcomings of current public finger knuckle dataset for contactless detection, we use a web crawler to get images from the

Unsplash [28] where the keywords are finger knuckles. The Unsplash is an image site that offers uploads and downloads, and uses a copyright license that allows users to download and use them for free or even for commercial use [29]. We have downloaded 2347 images, there are 738 images without knuckles, and these images can be used as background training, and the rest 1609 images that contain at least one finger knuckle are the positive samples for the network model. In the network training process, we use crawled images, 169 finger knuckle images from the HKPolyU Finger Knuckle Database (V1.0) [23], and 64 finger knuckles images from the HKPolyU Hand Dorsal Database [25] as for the training set. And we use the rest data as testing set to evaluate performance. The most important part is the data augmentaion which conatains flip, rotation, resize, translate and mosaic.

6.1.3 Contactless Finger Knuckle Detection

Detection Performance

The YOLOv4 model predict horizontal bounding box, while the remaining YOLOv5 model predict rotated bounding box with CSL classification, called YOLOv5-CSL. We can see the performance difference between these variations of the YOLOv5 model from the Table 2. Among the downloaded 2580 images, 100 images were randomly selected as the testing set.

Model	Inference Time/ms (1024x1024)	Number of Layers	mAP^{val} 0.5	AP of Major FK	AP of Minor FK
YOLOv5x-CSL	41.395	407	**89.9**	**89.6**	**90.1**
YOLOv5m-CSL	36.252	263	85.7	88.9	80.4
YOLOv4	25.992	161	70.7	83.6	57.7

Table 2: Comparison of the accuracy of the different models of the YOLO series for the detection of the finger knuckle.The calculated values of mAP were measured at a detection threshold of 0.4 as well as an IOU threshold of 0.5.

Segmentation Performance

This section aim to compare quality of finger knuckle between YOLOv5-CSL 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).

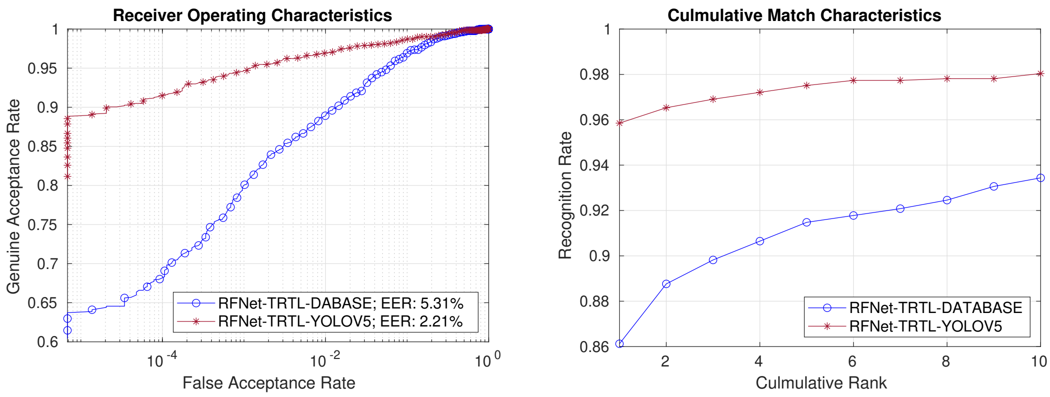


Figure 9: Compare performance on the Finger Knucle V3 Dataset (with deformable)

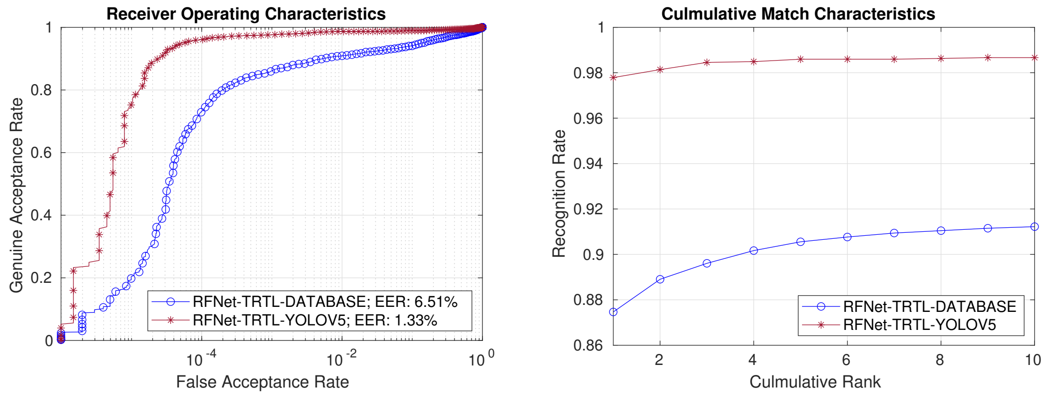


Figure 10: Compare performance on the Index Finger of the Hand Dorsal Image Database.

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 through the ROC curve and CMC curve. Especially on the Hand Dorsal Image Database, the EER value can drop from 6.51% to 1.33%.

6.2 Online Contactless Finger Knuckle Identification Performance

....

7 References

[1] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. "Yolov4: Optimal speed and accuracy of object detection". In: *arXiv preprint arXiv:2004.10934* (2020).

[2] Kevin HM Cheng and Ajay Kumar. "Contactless biometric identification using 3D finger knuckle patterns". In: *IEEE transactions on pattern analysis and machine intelligence* 42.8 (2019), pp. 1868–1883.

[3] Kevin HM Cheng and Ajay Kumar. "Deep feature collaboration for challenging 3D finger knuckle identification". In: *IEEE Transactions on Information Forensics and Security* 16 (2020), pp. 1158–1173.

[4] Ross Girshick. “Fast r-cnn”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1440–1448.

[5] Ross Girshick et al. “Rich feature hierarchies for accurate object detection and semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587.

[6] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[7] Kaiming He et al. “Mask r-cnn”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2961–2969.

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

[9] Gao Huang et al. “Densely connected convolutional networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708.

[10] Wei Jia, De-Shuang Huang, and David Zhang. “Palmprint verification based on robust line orientation code”. In: *Pattern Recognition* 41.5 (2008), pp. 1504–1513.

[11] AW-K Kong and David Zhang. “Competitive coding scheme for palmprint verification”. In: *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.* Vol. 1. IEEE. 2004, pp. 520–523.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems* 25 (2012).

[13] 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.

[14] Joseph Redmon and Ali Farhadi. “YOLO9000: better, faster, stronger”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 7263–7271.

[15] Joseph Redmon and Ali Farhadi. “Yolov3: An incremental improvement”. In: *arXiv preprint arXiv:1804.02767* (2018).

[16] Joseph Redmon et al. “You only look once: Unified, real-time object detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.

[17] Shaoqing Ren et al. “Faster r-cnn: Towards real-time object detection with region proposal networks”. In: *Advances in neural information processing systems* 28 (2015), pp. 91–99.

[18] 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.

[19] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[20] Zhenan Sun et al. “Ordinal palmprint represention for personal identification [represention read representation]”. In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*. Vol. 1. IEEE. 2005, pp. 279–284.

[21] Mingxing Tan and Quoc Le. “Efficientnetv2: Smaller models and faster training”. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 10096–10106.

[22] *The HKPolyU 3D Finger Knuckle Images Database:* https://www4.comp.polyu.edu.hk/~csajaykr/3DKnuckle.htm.

[23] *The HKPolyU Contactless Finger Knuckle Images Database (V-1.0):* http://www4.comp.polyu.edu.hk/~csajaykr/fn1.htm.

[24] *The HKPolyU Contactless Finger Knuckle Images Database (Version 3.0) :* https://www4.comp.polyu.edu.hk/~csajaykr/fn2.htm.

[25] *The HKPolyU Contactless Hand Dorsal Images Database:* http://www4.comp.polyu.edu.hk/~csajaykr/knuckleV2.htm.

[26] *Tsinghua University Finger Vein and Finger Dorsal Texture Database (THU-FVFDT3):* https://www.sigs.tsinghua.edu.cn/labs/vipl/thu-fvfdt.html.

[27] Ultralytics. *YOLOv5*. https://github.com/ultralytics/yolov5. 18 May 2020.

[28] Unsplash. *Unsplash*. https://unsplash.com/.

[29] Unsplash. *Unsplash.com*. ”Unsplash License”. Retrieved 11 January 2017.

[30] Xue Yang and Junchi Yan. “Arbitrary-oriented object detection with circular smooth label”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 677–694.

[31] Fisher Yu and Vladlen Koltun. “Multiscale context aggregation by dilated convolutions”. In: *arXiv preprint arXiv:1511.07122* (2015).

[32] Lin Zhang et al. “3D palmprint identification using block-wise features and collaborative representation”. In: *IEEE transactions on pattern analysis and machine intelligence* 37.8 (2014), pp. 1730–1736.

[33] Qian Zheng, Ajay Kumar, and Gang Pan. “A 3D feature descriptor recovered from a single 2D palmprint image”. In: *IEEE transactions on pattern analysis and machine intelligence* 38.6 (2016), pp. 1272–1279.

[34] Xizhou Zhu et al. “Deformable convnets v2: More deformable, better results”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 9308–9316.