Completely Contactless and Online Finger Knuckle Identification

Zhenyu ZHOU

June 25, 2022

1 Whole Image Translated and Rotated Triplet Loss Function

Because our neural networks were trained using the architecture of triplet network [13], we used translated and rotated triplet loss function (TRTLoss) as loss function to update convolutional kernel of our models. When we match two finger knuckles 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 [8].

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 $-W \leq w \leq W, -H \leq h \leq H, -\Theta \leq \theta \leq \Theta$. 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.

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

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

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 Shift Translation Triplet Loss [8] performance on different finger knuckle dataset. Meanwhile, we will also compare with the state-of-the-art FKNet [2] and EfficientNetV2 [14]. All the experiment will use the finger knuckle segmented by YOLOv5-CSL as training set and testing set. In the Online Contactless Finger Knuckle Identification section, 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 [8] convolution layer with deformable convolution layer. Then as for DeConvRFNet and RFNet model, we pretrained on the Finger Knuckle V1 Database [15].EfficientNetV2-S is the original classification model, we keep the same architecture and just change the FC layer of the head part with convolution layer for fitting TRTL and STTL to compose EfficientNetV2-S-STTL and EfficientNetV2-S-TRTL model. When trained these EfficientNetV2-S model, we use the pretrained weight on the ImageNet21K. As for the FKNet, we use the pretrained ResNet-50 weights.

Because the EfficientNetV2-S and FKNet 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 Model Complexity Analysis

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 [14]	20.18M	300x300	5.40B		
EfficientNetV2-S-STTL	20M	300x300	5.38B		
EfficientNetV2-S-TRTL	20M	300x300	5.38B		
FKNet [2]	7.28M	96x64	0.28B		
RFNet-STTL [8]	0.46M	128x128	1.39B	0.0062s	0.049s
RFNet-TRTL	0.46M	128x128	1.39B	0.0062s	

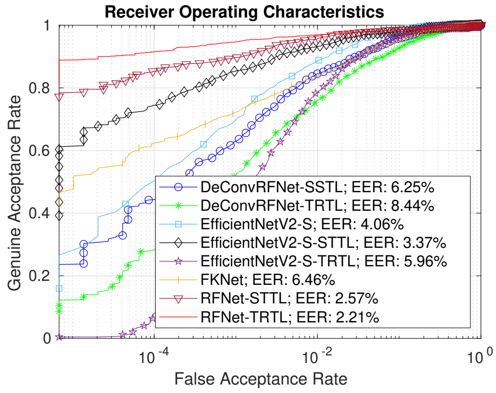
Table 1: Model complexity analysis

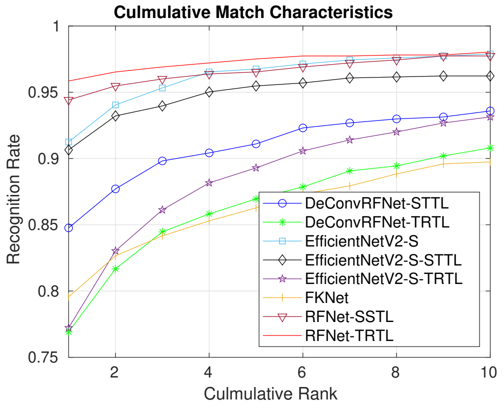
....

2.2 Within-Database Experiments

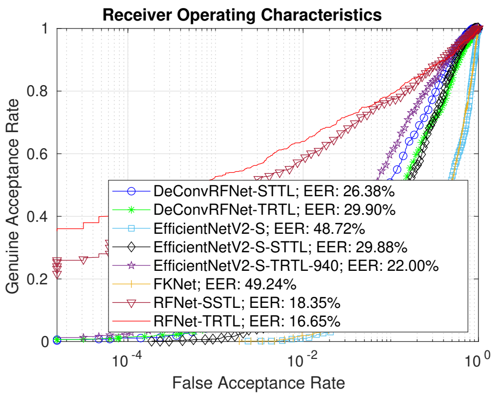
2.2.1 Finger Knuckle V3 Database with Deformation

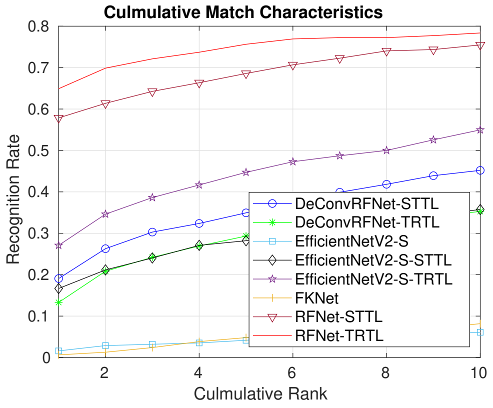
The Finger Knuckle V3 Database have 1-104 subjects that have two session samples, and the rest subjects of first session 105-221 just offer one session samples. So as the first experiment, I firstly fine-tuned my model on the second session of 1-104 subjects, and test on the first session 1-104 subjects. 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.

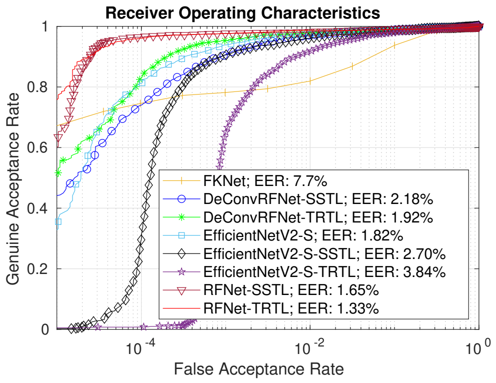


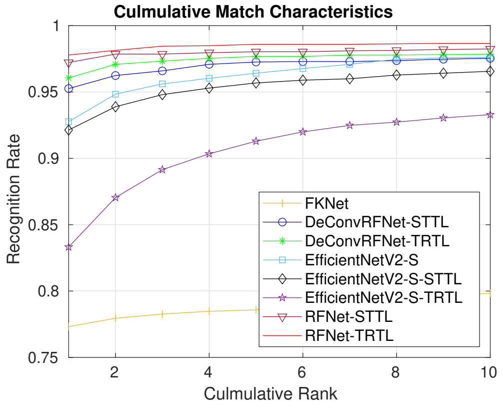


The two-session protocol will use the session1 as the probe and use the session2 as the enrollment. As for the genuine matching scores, each sample of a subject will choose the minimal matching score when compare to the rest samples. In this kind of situation, it will have $104x6$ genuine matching scores. 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.

With Yolov5-CSL segmented finger knuckle, the RFNet performance is slightly higer than the local feature descriptors based on key points matching [6], and performance higher than the paper [7]. If we want to compare different method performance, I think we should use same dataset. In this kind of situation, the method of [6] maybe will ger higer performance on the segmented finger knuckle by YOLOV5.

2.2.2 Index Finger Knuckle of Hand Dorsal Image Database



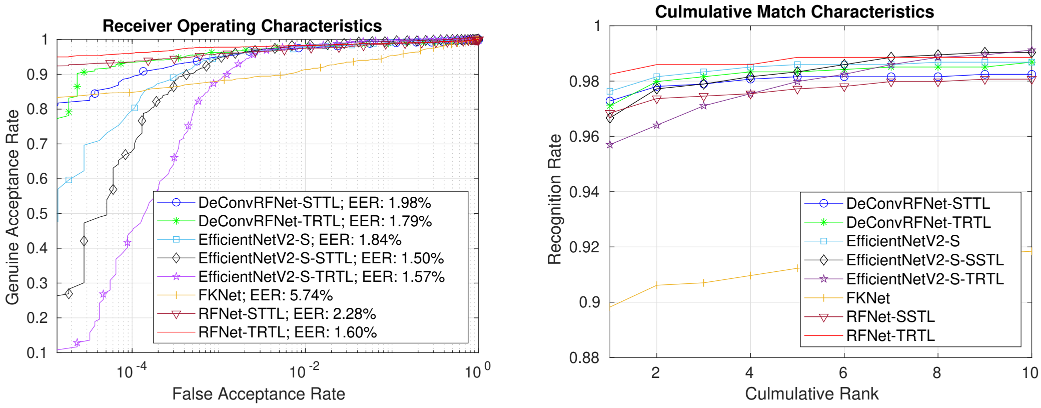


For the ROC curve, I add EfficientNetV2-S model performance. Update CMC Curve and ROC Curve with EfficientNet, DeConvRFNet and RFNet. As for the experiment, the dataset totally contains 712 subjects, and I use the segmented Index finger knuckle as my dataset. And I fine-tuned my model on the first sample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

2.2.3 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.

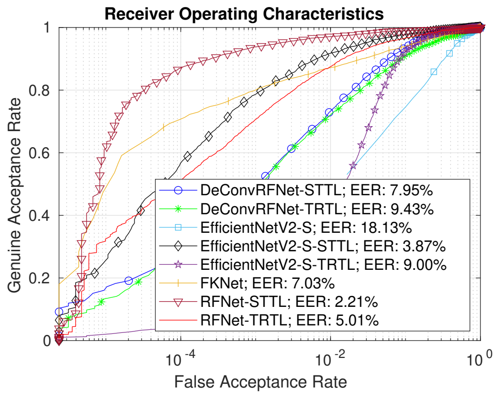


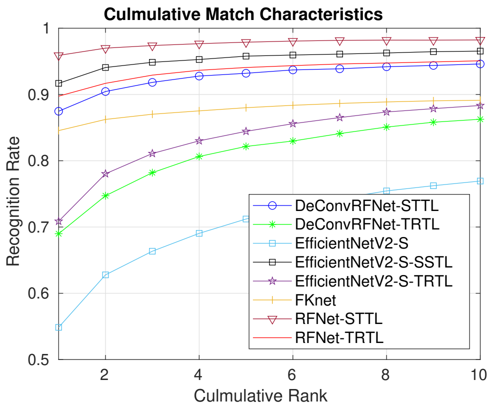
And then use the two session protocol. I use the rest samples of session1, and it has 191-228 subjects. In this kind of situation, the training dataset is too small. The two session protocol will test on the 190 subjects, these subjects can offer two session samples. Due to the training set is too small,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

2.3 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.

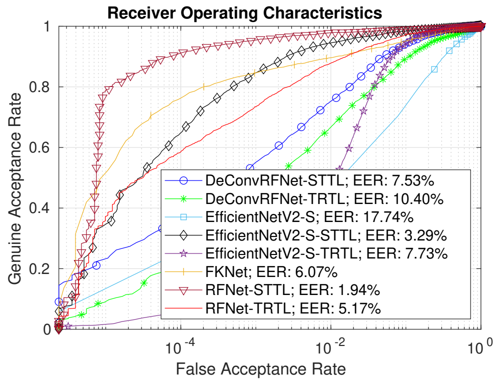
2.3.1 Index Finger Knuckle of Hand Dorsal Image

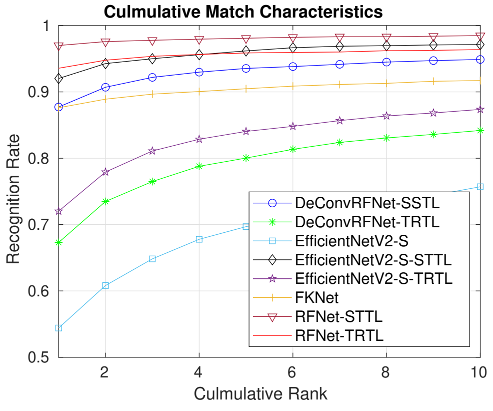




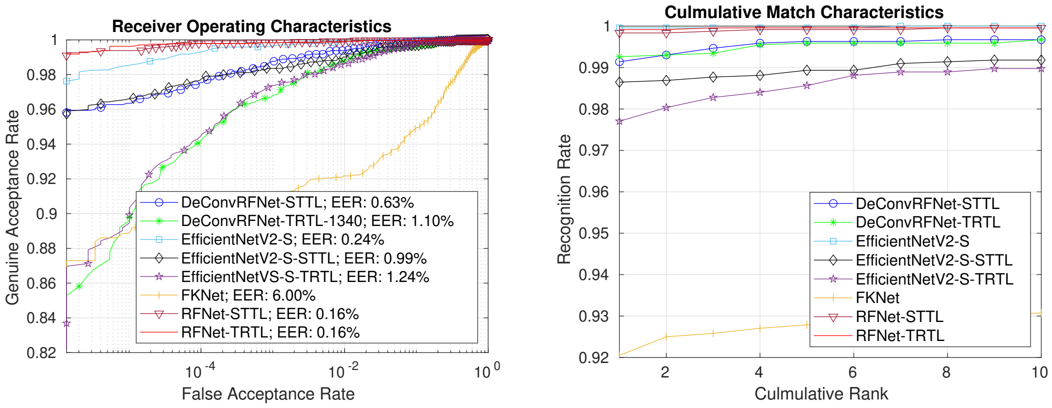
The database totally has 712 subjects, and each subject has 5 samples. Therefore, it will have $712 * 5$ genuine matching scores and $712 * 711 * 5$ imposter matching scores. From the cure, the performance of RFN-SSTL and RFN-TRTL is similar, and the RFN-WS is slightly better than RFN-WRS while using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

2.3.2 Middle Finger Knuckle of Hand Dorsal Image





2.3.3 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.

2.4 Discussion

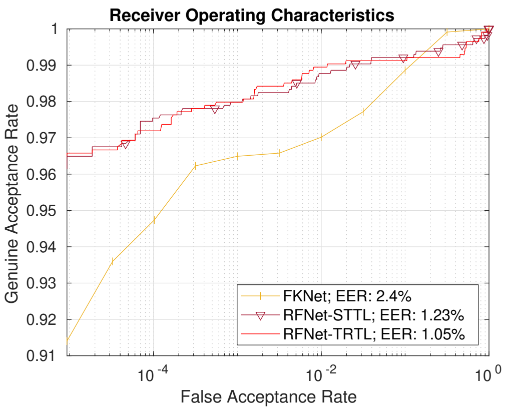
From these experiment results, we can see that EfficientV2-S model is better than FKNet in some dataset. Becuase EfficientNetV2 model use MBConv as a block unit for replacing residual block. As for MBConv block, it use depthwise convolutions to decrease training weights and use Squeeze-Excited block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

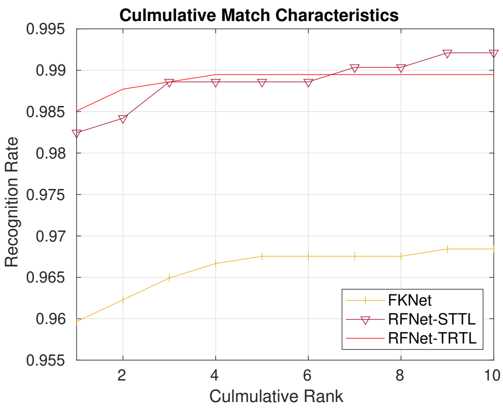
There is another conclusion is that TRTL generalization ablity is lower thant STTL loss from the cross database experiment. But in the within database experiment, these model with TRTL loss is better thant STTL loss.

....

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 Online Contactless Finger Knuckle Identification

With TRTL loss, the RFNet [8] 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 [11], [9], [10], [1], [17] and R-CNN [4], [3], [12], [5] 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.

4.1 Contactless Finger Knuckle Detection

4.1.1 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 [17] 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 [20], 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) [20] 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.

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

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

4.1.2 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 [18] 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 [19]. 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) [15], and 64 finger knuckles images from the HKPolyU Hand Dorsal Database [16] 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.

4.1.3 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).

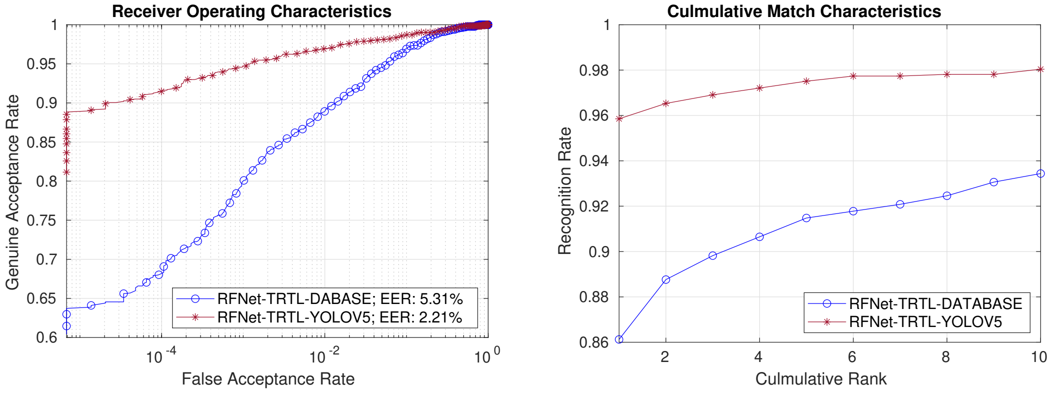


Figure 9: Compare performance on the Finger Knucle V3 Dataset (with deformable)

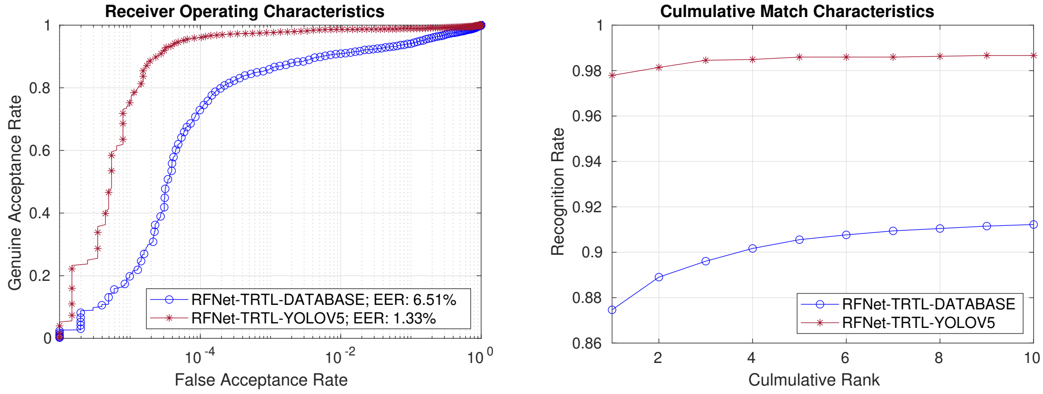


Figure 10: Compare performance on the Index Finger of the Hand Dorsal Image Database.

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%.

4.2 Online Contactless Finger Knuckle Identification Performance

....

5 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. “Deep feature collaboration for challenging 3D finger knuckle identification”. In: *IEEE Transactions on Information Forensics and Security* 16 (2020), pp. 1158–1173.

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

[4] 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.

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

[6] 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.

[7] 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.

[8] 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.

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

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

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

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

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

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

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

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