Weekly Report

Zhenyu Zhou

November 19, 2022

1 Finger Knuckle Project

1.1 Using histogram equalization to normalize image

I have tried linear, log, and histogram to normalize image for getting better bounding box detection. From experiments, I found histogram equalization algorithm can get the best performance for detecting finger knuckle on the dark light.

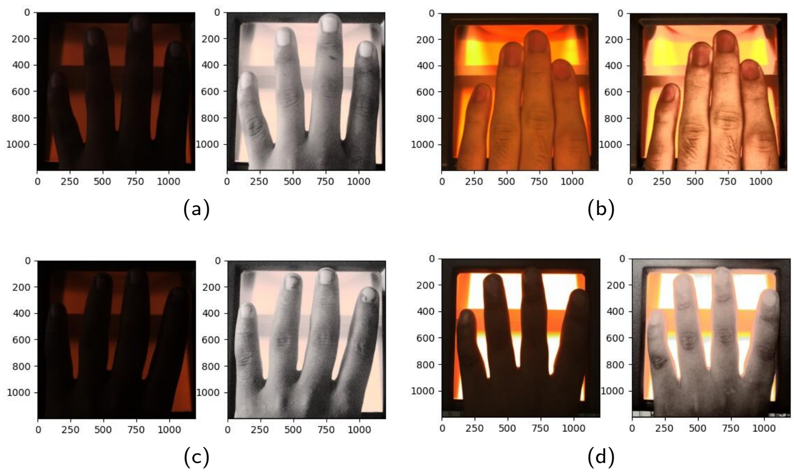


Figure 1: After using histogram equalization algorithm, the texture of finger knuckle is very clear for detecting.

1.2 Choose the hyperparameters

1.2.1 Input image size

I have tried that change the input image size $h \times w$ from 128×128 to 208×184. As for the former size, it just follows the RFN model parameter, and the latter input image size is the mean size of the segmented finger knuckle. When I train the RFN model, I also removed the values of ten pixels on each side of the finger knuckle for eliminating background interference. I just used the RFN to test the performance on the middle finger knuckle of left hand, as shown the Table 1. From the result, when compare to the 208×184, the 128×128 can get better performance. Although the difference of performance is very little, the rest experiments are based on the input 128×128 with all-to-all protocol.

Table 1: Verification performance under different input image size

Protocol	Size	Left-Little		Left-Ring		Left-Index	
		EER	GAR @ FAR10^{-4}	EER	GAR @ FAR10^{-4}	EER	GAR @ FAR10^{-4}
Leave One-Out	**128x128**	**3.34%**	**79.00%**	**0.84%**	**97.00%**	**2.00%**	**85.00%**
	208x184	4.17%	79.00%	1.00%	97.00%	2.00%	79.00%
All-to-All	**128x128**	**8.19%**	**52.00%**	**2.67%**	**84.00%**	**4.92%**	**64.00%**
	208x184	10.08%	49.00%	3.04%	83.00%	6.00%	60.00%

1.2.2 The hard margin of triplet loss

The hard margin will affect the triplet loss to push negative samples away positive samples (probe). But if the hard margin too large, the loss will hard to converge or the model will overfitting. On the contrary, the network cannot be adequately trained result in bad performance on testing set. During training, I change the hard margin on the range of [5, 10, 15, 20, 25, 30, 35, 40], as shown on the Table 2. It can be obviously noticed when $alpha = 10$ the $GAR@FAR = 10^{-4}$ can get the highest values. So in the next section, I will modify hard margin parameter around this value.

Table 2: Triplet loss with different hard margin

α	Left-Little		Left-Ring		Left-Index	
	EER	GAR @ FAR=10^{-4}	EER	GAR @ FAR=10^{-4}	EER	GAR @ FAR=10^{-4}
5	8.83%	40.00%	2.50%	79.00%	5.00%	63.00%
10	8.50%	**56.00%**	2.33%	**86.00%**	4.17%	**76.00%**
15	9.67%	55.00%	2.54%	81.00%	4.85%	65.00%
20	8.19%	52.00%	2.67%	84.00%	4.91%	64.00%
25	**7.67%**	54%	2.67%	76%	4.97%	64%
30	9.08%	50%	**1.76%**	79%	**4.00%**	55%
35	10.02%	35%	2.67%	52%	5.36%	47%
40	10.92%	48%	2.83%	79%	5.58%	64%

1.2.3 The hard margin of quadruplet loss

Because the triplet loss function only focus on the distance between probe images and negative images, the inter-class will ignore by some extent. For increasing intra-class variance and discreasing inter-class variance, the quadruplet loss using four samples (anchor, positive, negative, negative2). But the quadruplet loss has two hard margin, the first one alpha is same as triplet loss while the alpha2 is relative weak. From the Table 3, the performance will not have huge different, even the triplet loss can get the best performance on the index finger knuckle of left hand. As for the rest finger knuckle, the performance are similar. When I calculate the triplet, within the same subject, I choose the sample with the greatest distance from the anchor as the positive. For the negative, I compare it with all the samples of the other two subjects and choose the one with the smallest distance.

Table 3: Quadruplet loss with different hard margin

α	$\alpha 2$	Left-Little		Left-Ring		Left-Index	
		EER	GAR @ FAR=10ˆ-4	EER	GAR @ FAR=10ˆ-4	EER	GAR @ FAR=10ˆ-4
10	0	8.50%	56.00%	2.33%	86.00%	**4.17%**	**76.00%**
10	5	8.75%	56.00%	2.42%	79.00%	4.92%	60.00%
10	10	8.75%	55.00%	2.92%	72.00%	5.33%	60.00%
10	15	8.83%	**61.00%**	2.17%	**90.00%**	5.08%	72.00%
20	5	8.58%	59.00%	1.86%	86.00%	4.50%	**75.00%**
20	10	**8.17%**	59.00%	**1.50%**	81.00%	**3.92%**	70.00%
20	15	10.92%	45.00%	2.50%	82.00%	5.75%	61.00%
20	20	8.69%	**66.00%**	1.70%	**89.00%**	5.10%	63.00%
30	15	**7.50%**	59.00%	**1.54%**	76.00%	4.75%	58.00%
40	20	9.08%	58.00%	2.09%	86.00%	5.00%	70.00%

1.3 Using spatial transformer network to increase RFN performance

Spatial transformer network (STN) can learn the matrix of homography transform to affine finger knuckle. If the loss higher than a threshold during training process, I freeze the spatial transformer network layer without updating weights. And the STN just output $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$ matrix. When the loss lower than a threshold, the STN weights can be updated for outputting a transform matrix. When I use the STN module to improve RFN performance, from the Table 4, the STN module can improve the RFN model silght

Table 4: Peformance of spatial transfomer network based on RFN

Model	α	T	Left-Little		Left-Ring		Left-Index	
			EER	GAR @ FAR=10^{-4}	EER	GAR @ FAR=10^{-4}	EER	GAR @ FAR=10^{-4}
RFN	10	0	8.50%	56.00%	2.33%	**86.00%**	4.17%	**76.00%**
STNRFN	10	10	**6.92%**	**58.00%**	**1.42%**	**86.00%**	4.92%	60.00%
RFNSTN	10	10	8.06%	54.00%	2.00%	80.00%	3.91%	62.00%
STNRFN	10	0	**6.832%**	49.00%	1.50%	65.00%	**3.334%**	61.00%
ResSTN	10	NA	8.42%	40.00%	2.09%	70.00%	4.92%	48.00%
ResSTN	10	0	10.624%	45.00%	2.25%	84.00%	5.29%	70.00%

1.4 Using Rotation and Shift Invariant loss to train RFN model

Above experiments are based on the MSE loss function to train model, then I use the RSIL loss function to replace the MSE loss function to test the performance. From the below Table

Table 5: Peformance of RSIL loss based on RFN

Model	α	Left-Little		Left-Ring		Left-Index	
		EER	GAR @ FAR=10^{-4}	EER	GAR @ FAR=10^{-4}	EER	GAR @ FAR=10^{-4}
RFN-MSE	10	8.50%	56.00%	2.33%	86.00%	4.17%	76.00%
RFN-RSIL	10	**6.34%**	**58.00%**	**1.08%**	**96.00%**	**2.834%**	**87%**

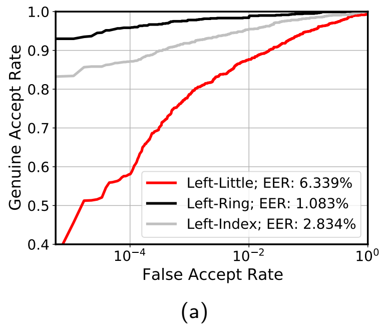


Figure 2: Comprative mathing performance with ROC cure on the littler finger knuckle, ring finger knuckle, and index finger knuckle of left hand.

1.4.1 Data augmentation

Table 6: Data augmentation

Augmentation	Parameter	Augmentaion	Parameter
hsv_h	0.015	shear	NA
hsv_s	0.7	perspective	NA
hsv_v	0.4	flipud	NA
degrees	10	fliplr	NA
translate	0.1	mosaic	NA
scale	0.1	mixup	NA

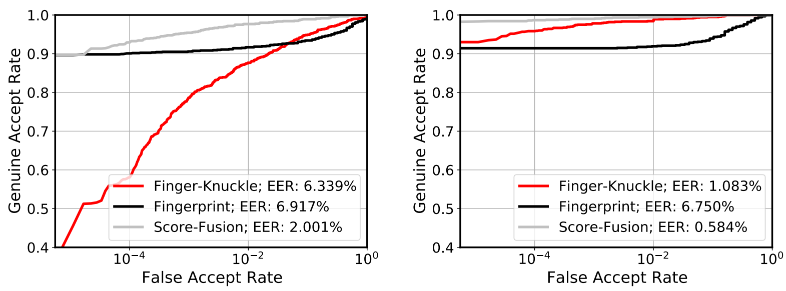
From the Table 6, I used the left part data augmentation algorithm, because the right part algorithm will totally change the finger knuckle texture which cannot be solved by RFN-RSIL model. Meanwhile, the RFN-RSIL model gets the best performance form above experiments, therefore, I continue to use the RFN-RSIL model. The model is still be trained, so I cannot show any performance result at here.

1.5 Find the best score fusion parameter

1.5.1 Dynamic fusion

$$fusion\_score = w \times knuckle\_score + (1 - w) \times print\_score \quad (1)$$

I have changed the w from 0 to 1, the step size is 0.05, and I can get the best fusion score when w equal to 0.35. I mainly tried on the little finger and ring finger of left hand. From the



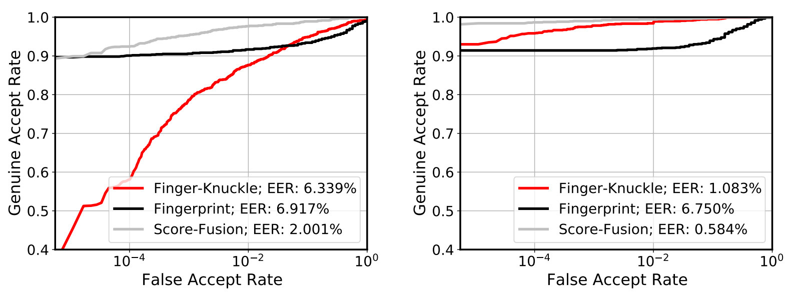
(a) (b)

Figure 3: When the w equal to 0.35, the dynamic fusion score algorithm can get the best matching performance on the (a)little finger of left hand (b) ring finger of left hand.

1.5.2 Holistic Fusion

$$fusion\_score = (w \times knuckle\_score + (1 - w) \times print\_score) \times (1 + 1/(2 - knuckle\_score)) \quad (2)$$

I also changed the w from 0 to 1 with step size 0.05 on the Equation (2). From the Fig. 4 and Fig. 3, their performance is same from the ROC curve.



(a) (b)

Figure 4: When the w equal to 0.2, the holistic fusion score algorithm can get the best matching performance on the (a)little finger of left hand (b) ring finger of left hand.

1.5.3 Nonlinear Fusion

$$fusion\_score = ((1 + print)/(1 + knuckle))^{\alpha} \times (1 + knuckle)^2 \quad (3)$$

As for the nonlinear fusion method, we can change the α from 1 to 2, then I use 0.05 step size to change the parameter. When the *alpha* is 1.45, then the nonlinear fusion method can get the best performance, from the Fig. 5. From only test on the little finger and ring finger of left hand, the nonlinear fusion method is the best.

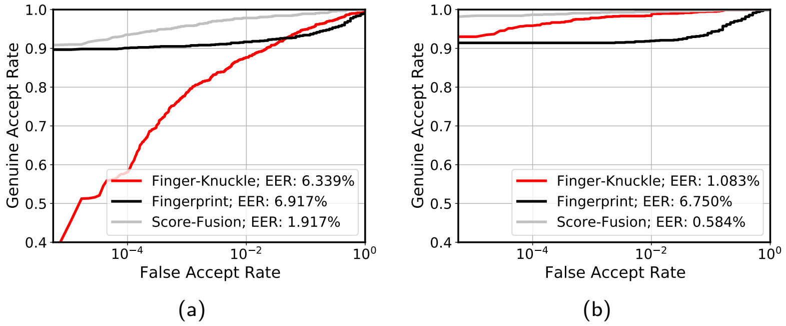


Figure 5: When the w equal to 0.45, the holistic fusion score algorithm can get the best matching performance on the (a)little finger of left hand (b) ring finger of left hand.