

Weekly Report

Zhenyu Zhou

November 26, 2022

1 Finger Knuckle Project

1.1 Data Augmentation

I mainly used color transformation on the HSV color domain (hsv_h, hsv_s, and hsv_v), and geometric transformation by rotating, translating, and scaling. **But the MSE is too sensitive about rotate, translate, and scale due to the fact that MSE is calculated by pixel-to-pixel. Therefore, the model is very hard to be trained when I used geometric transformation.**

Table 1: Recognition performance with data augmentation

Model	Loss	Data	α	Left-Little		Left-Ring		Left-Index	
				EER	GAR	EER	GAR	EER	GAR
RFNet	MSE	NA	10	8.50%	56.00%	2.33%	86.00%	4.17%	76.00%
RFNet	MSE	HSV	10	9.92%	56%	2.42%	86%	5.25%	76%
RFNet	Mask MSE	HSV Geo	10	11.58%	40%	3.58%	42%	3.58%	38%

From the above Table 1, the performance of using HSV augmentation is same as the performance of model that don't use any data augmentation. Just like I said before, the MSE is too sensitive to geometric transformation result in poor performance with geometric transformation data augmentation. As for the Mask-MSE on the Table 1, its aim is to mask the valid pixel, because it will have a lot of dark margin or invalid pixel after geometric transformation.

1.2 Structural similarity index measure (SSIM)

As we know, the MSE loss function is too sensitive to geometric transformation, although it is simple enough. There are lots of similarity index that can be used, such as PSNR, SSIM, etc. At here, I used SSIM to replace the MSE loss function for improving model performance. The SSIM can calculate the luminance similarity, contrast similarity, and structure similarity for get the final similarity score between two images. **Therefore, I used differentiable SSIM with PyTorch to replace the MSE loss function.**

The similarity score of SSIM is from 0 to 1, then I changed the triplet loss margin from 0.5 to 1 for testing performance under different hard margin. From the Table 2, when the alpha is equal to 0.5, the RFNet with SSIM can get the best performance. When compare to MSE, the SSIM is still can get better performance on the left-little and left-index finger knuckle. As for the left-ring finger knuckle, the GAR@FAR= 10^{-4} of SSIM is lower than MSE, but the EER is higher.

Table 2: Recognition performance with SSIM

Model	Loss	Alpha	Left-Little		Left-Ring		Left-Index	
			EER	GAR	EER	GAR	EER	GAR
RFNet	SSIM	0.5	8.87%	61.00%	1.83%	73.00%	4.20%	79.00%
RFNet	SSIM	0.6	9.67%	57%	2.40%	82%	4.38%	60%
RFNet	SSIM	0.7	9.33%	54%	2.82%	81%	3.92%	72%
RFNet	SSIM	0.8	9371%	N/A	2.75%	53%	5.17%	N/A
RFNet	SSIM	0.9	8.25%	N/A	2.33%	N/A	4.92%	N/A
RFNet	SSIM	1	8.92%	N/A	3.25%	N/A	4.68%	N/A
RFNet	MSE	10	8.50%	56%	2.33%	86%	4.17%	76%

1.3 Multi-channel feature map

As for the RFNet, it only output one channel feature map for matching. When we use different dimension kernel, then the CNN model can learn different texture from finger knuckle. In this kind of situation, I change the RFNet output feature maps with 64 channels. **From the Table 3, the recognition performance can be enhanced by increasing the output feature maps' channels. Especially, when hard margin is equal to 0.5, the GAR is the highest.**

Table 3: Recognition performance with multi-channel feature map

Model	Loss	Alpha	Left-Little		Left-Ring		Left-Index	
			EER	GAR	EER	GAR	EER	GAR
RFNet64	SSIM	0.3	6.50%	60.00%	1.33%	79.00%	4.50%	50.00%
RFNet64	SSIM	0.4	6.28%	60.00%	1.17%	88.00%	3.17%	71.00%
RFNet64	SSIM	0.5	7.83%	66.00%	1.91%	88.00%	4.17%	82.00%
RFNet64	SSIM	0.6	7.83%	52.00%	2.17%	80.00%	3.75%	73.00%
RFNet64	SSIM	0.7	8.42%	52.00%	2.02%	68.00%	4.83%	42.00%
RFNet	SSIM	0.5	8.87%	61.00%	1.83%	73.00%	4.20%	79.00%
RFNet	MSE	10	8.50%	56.00%	2.33%	86.00%	4.17%	76.00%

1.3.1 Multi-channel with data augmentation

From the Table 1 and 4, we can get a conclusion that my current data augmentation have a negative effects on finger knuckle recognition performance.

Table 4: Recognition performance with data augmentation

Model	Loss	Aug	α	Left-Little		Left-Ring		Left-Index	
				EER	GAR	EER	GAR	EER	GAR
RFNet64	SSIM	TRUE	0.4	9.00%	42%	2.33%	63%	3.67%	56%
RFNet64	SSIM	TRUE	0.5	10.46%	N/A	2.75%	N/A	3.92%	N/A
RFNet64	SSIM	N/A	0.5	7.83%	66.00%	1.91%	88.00%	4.17%	82.00%
RFNet	SSIM	N/A	0.5	8.87%	61.00%	1.83%	73.00%	4.20%	79.00%
RFNet	MSE	N/A	10	8.50%	56%	2.33%	86%	4.17%	76%

1.4 Attention mechanism

The CNN network and SSIM loss don't have the rotation and shift invariant capability which is very important in biometrics. At the same time, both of them just only use local feature information which limits their ability. On the contrary the attention mechanism can use global information result in robust features among NLP and CV tasks. In generally, the attention mechanism should be feed with huge data for learning the relation between local feature and global feature. I use the RFNet as the feature extract backbone to extract finger knuckle feature, and then use the self-attention and cross attention to get more robust feature maps. Finally, I use the SSIM to get the matching scores, called SSIMGNN loss. **From the Table 5, attention mechanism cannot get better performance, on the contrary, the performance has dropped a lot.**

Table 5: Recognition performance with attention mechanism

Model	Loss	Aug	α	Left-Little		Left-Ring		Left-Index	
				EER	GAR	EER	GAR	EER	GAR
RFNet64	SSIMG Relu	N/A	0.5	14.17%	N/A	9.08%	55.00%	11.50%	41.00%
RFNet64	SSIMG	N/A	0.5	15.17%	N/A	6.63%	N/A	8.59%	N/A
RFNet64	SSIMG2	N/A	0.5	23.50%	N/A	16.24%	N/A	25.86%	N/A
RFNet64	SSIMG	HSV Geo	0.5	22.58%	N/A	13.00%	44.00%	16.61%	N/A
RFNet64	SSIM	N/A	0.5	7.83%	66.00%	1.91%	88.00%	4.17%	82.00%
RFNet	SSIM	N/A	0.5	8.87%	61.00%	1.83%	73.00%	4.20%	76.00%
RFNet	MSE	N/A	10	8.50%	56.00%	2.33%	86.00%	4.17%	79.00%