Completely Contactless and Online Finger Knuckle Identification for Real World Application

Zhenyu ZHOU

July 26, 2022

1 Abstract

2 Introduction

Biometrics is the use of the human body's inherent physiological and behavioral characteristics for person identification. For the physiological characteristics, there are many human characteristics such as retina, iris, face, fingerprints and palm prints; as far as the behavioral characteristics, the available features include gait recognition, voice and other behavioral characteristics. These biometric characteristics are very popular areas of research, attracting many researchers, and have been applied in a wide range of industries. Among human physiological characteristics, finger knuckle, as an optional character, is easier to expose, convenient to collect, and the most importance is that it has been proven to be unique and stable [18]. Therefore, it has also attracted many researchers devoted to it, related works including from the earlier [20], [56] to nowadays [5], [39]. Contactless finger knuckle identification method, a research field of finger knuckle, is more user-friendly, security and more hygienic when compared to contact-based identification. Especially with today's epidemic, contactless identification method can efficiently inhibit the spread of the virus by preventing cross-contamination. It is due to the above-mentioned points that contactless finger knuckle have also gained a lot of attention. And due to the rapid development of neural networks, their powerful generalization capabilities and their great success in computer vision, researchers are increasingly inclined to use deep learning methods to solve the problem of contactless finger knuckle recognition.

Although the contactless finger knuckle identification offers a lot of convenience, there are two main problems at the contactless scenarios: one is the degradation of matching performance and the other is how to segment the finger knuckle region of interest (ROI) efficiently on the complexity background. Finger knuckle are more prone to deformation in the absence of fixation, which occurs not only in the 2D plane but also in the 3D plane, the presence of variations in ambient light, etc. These factors can cause crease texture of finger knuckle to vary considerably between individuals result in matching performance degradation. In the contactless scenario, finger knuckle are exposed to the interference of a complex background on captured contactless finger knuckle images or videos, and the position, size and number of finger knuckle appearing in each frame varies. This can make automatic finger knuckle detection and segmentation more difficult. If researchers can solve above two main problems, then the development and application of contactless finger knuckle identification will be very promising, and can enhance recognition performance with other biometrics identifiers.

2.1 Related Work

Completely Contactless Finger Knuckle Identification

Many early works on contactless finger knuckle identification, ranging from coding-based methods, subspace methods and texture analysis methods to 3D shape patterns based on 3D image reconstruction, and different methods have been used to achieve highly accurate recognition results. For the traditional recognition algorithms, there are generally two main types: one is holistic-based, and the other is local feature-based. The broad category can be divided into subspace and spectral representation methods for holistic-based [50], [28], [26]. Subspace methods are generally used for data dimensionality reduction and noise reduction [54], such as the use of principal component analysis to reduce the dimensionality of multidimensional data. In contrast to spectral representation methods, image space transformation can be performed as well as image feature enhancement and correlation coefficients for feature extraction [10]. For the processing of local information, there are many algorithms, including, for example, extracting information about the gradient of the image edges, obtaining the boundary points, using other edge extraction algorithms such as Hough change. For example, a 1D log-Gabor filter was used to extract the finger knuckles' features and for the matching phase, the hamming distance was used here for the matching score calculation since it is a local feature-based method [27]. Alternatively, a 2D Gabor filter is used to extract the domain orientation information features of the finger knuckles, and an angular distance calculation is used to calculate the similarity between the different features for the matching score [25]. High recognition accuracy has been achieved for these matching algorithms, even up to 98.67% [50]. Even early work considered application scenario using cell phones for finger knuckle recognition [3], but for finger knuckle segmentation using fixed finger position in the center of the image is not very convenient for the user, for recognition phase log-Gabor is used for feature extraction, and Hamming distance is used for matching. Meanwhile, these paper [19], [5], [15] are the latest in research on the topic.

With the development of neural network, the generalization ability of feature extraction and feature matching method becomes more robust. Especially, from the oldest LeNet [22] to nowadays EfficientNetV2 [38] model, the CNN models have achieved great success in computer vision tasks. Therefore, the CNN models also can be used on finger knuckle identification tasks to extract the robust features. Such as FKNet [5] based on the ResNet-50 [8], it uses the FKNet to classify the subject by the 3D deep feature information of finger knuckle. There are even many that combine traditional methods with CNN models. For example [15], it firstly uses initialized Gabor filters to extract features, and then uses CNN to learn the most important information, and the Gabor filter parameters can also update during backpropagation.

Completely Contactless Fingers Detection and Segmentation

For the efficiency of the matching algorithm, the accuracy of the region of interest of object is a factor that can determine the matching efficiency and accuracy. The size of the region of interest should be comparable to the size of the actual target object at the pixel level so that the size obtained after segmentation will not have redundant pixel information, which will reduce the pixel values to be computed for both the extraction and the subsequent matching sessions.

As for the traditional method, their [19], [3] approach is to fix the finger knuckle position in the image when taking the finger knuckle data without complexity background, and then extract the edges of the object, and then extract the finger knuckle crease part. Most importantly, the traditional segmentation algorithm 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. It is difficult to use traditional object segmentation methods to automatically segment finger knuckles for applications such as in the wild. Models of neural network models for object detection have achieved great success, whether it is the sliding window detection algorithm, the 2-stage series of R-CNN models [7], [6], [32], or the 1-stage YOLO series [31], [29], [30], [2] and SSD models [23] up to the current position, and even the anchor-free based object detection algorithm [49] as well. Each of these models has its advantages. For the 2-stage model, the object detection accuracy is guaranteed, the 1-stage based model is a speedup based on the positive accuracy, and the anchor-free is a further improvement in the detection speed. In this paper, the latest version of the YOLO model series, YOLOv5 [46], is used as the network model for finger knuckle detection because the YOLO series is famous for its fast detection speed and high accuracy.

Limitations and Challenges

Ref.	Deployment Scenario			Performance		
	Real-World Knuckle Detection	Completely Contactless	Online System	Database	EER	Recognition Rates at FAR=10^-4
[5]	No	No	No	D(1)	2.40%	94.70%
				A	3.90%	86.60%
				B	7.70%	74.50%
[51]	No	No	No	A	2.72%	N/A
				E	0.66%	N/A
[40]	No	No	No	A	3.97%	N/A
[4]	No	No	No	D	9.60%	∼81.5%
				D(2)	10.20%	∼76.0%
Ours	Yes	Yes	Yes	C	**2.21%**	**∼91%**
				B	**1.33%**	**∼97%**
				D	**1.05%**	**∼97.2%**
				D(2)	**1.60%**	**∼96%**
				E	**0.16%**	**∼99.9%**

Table 1: Database description: A: The HKPolyU Contactless Finger Knuckle Images Database (Version 1.0) [42]; B: The HKPolyU Contactless Hand Dorsal Images Database [44]; C: The HKPolyU Contactless Finger Knuckle Images Database (Version 3.0) [43]; D: The HKPolyU 3D Finger Knuckle Images Database (D(1) 3D Images, D(2) 2D Images) [41]; E: Tsinghua University Finger Vein and Finger Dorsal Texture Database [45].

We have listed some methods on the finger knuckle matching in the above section. In terms of these traditional algorithms, they have a common problem that these algorithms have to keep changing their corresponding filters and even the corresponding detection parameters under different scenario [57]. Although, based on the deep learning, different model have already achieved high matching accuracy. But at present, [5], [15] and [40] do not think about finger knuckle rotate and shift problem, and even finger knuckle crease is very easily deformation on the contactless identification. These problems are very vital for the matching performance on the contactless method. There is a common problem for the classification neural networks, is the output classes is fixed during training, just like the FKNet [5], result in re-training when a new subject is added. Although, we can use the feature vector before the soft-max to calculate the similarity score when matching.

As for the contactless finger knuckle detection and segmentation, [21] finger knuckle ROI extraction method is hard to deploy on the contactless scenario, and as for the [5], it based on the Mask R-CNN to segmentation, but the speed is too slow for a real-time identification system. 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, due to lack of contactless finger knuckle images under complex background, all these neural networks cannot detect, and segment finger knuckle under complex background result in the problem that how to efficiently detection and segment the finger knuckle in the wild. At present, object detection and segmentation neural networks 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.

The [3] paper is the first attempt for designing a contactless finger knuckle identification system. But it cannot deal with the online or real-time scenario, because it is too slow to verification. And it cannot detect finger knuckle in the wild, the system requires users put their finger on fix region when take pictures, and only support verify one subject at the same time. When designing a contactless and online finger knuckle identification system, how to segment and recognize finger knuckle in real time or online under complex backgrounds and how to recognize multiple subjects at the same time from one image.

2.2 Our Works

As for the contactless finger knuckle identification, the most difficult problem is how to deal with the deformable crease texture of finger knuckle while with different angle. Although many methods have obtained good recognition accuracy, the finger knuckles are easily deformed under practical application scenarios, and the finger knuckle features will change accordingly. Thus, the matching accuracy will be degraded. Because of the above problems, there are corresponding studies to solve the finger knuckle deformation problem and provide new data sets and new methods [19]. The paper [19] first matches on two images for a selected fixed number $32 * 32$ of point pairs for coping with the deformation problem and then uses local feature descriptors on each point pair for matching. And at the RFNet [24], it shifts the feature maps for getting the minimal matching score to solving the palmprint shift problem. Meanwhile, on the person re-identification problem, the paper [1] uses the cross-input neighborhood differences module to calculate difference of one feature maps with another with more range to add robustness to positional differences, and the paper [36] use the normalized correlation layer to achieve the similar problem. However, both of them just shift or calculate with more range but still on the horizontal and vertical direction, no one think about the rotation. Therefore, based on the Soft-Shifted Triplet Loss [24], we can also rotate to deal with more complex deformations.

Another problem is tow to automatically segment the finger knuckle region on the real time on the contactless and online finger knuckle identification. From the above contactless finger knuckle paper, they just use the traditional method to fix the finger knuckle position on an image or video. But just like I said before, the traditional method cannot solve the complexity phenomenon. And I have found that on the finger knuckle identification, it seems that no one use deep learning to detection and segmentation. A new finger knuckle object detection algorithm is used, which can automatically extract the region of interest of finger knuckle based on the YOLOv5 [46] model framework and integrates the angle prediction function. It is beneficial to the matching speed and accuracy of the matching algorithm and the automatic target segmentation using the object detection model to extract the ROI region of the major finger knuckle. The angle information of the finger knuckle can be obtained. If the algorithm of ROI segmentation is accurate enough and the accuracy of the rotation is also high enough, this will naturally improve the detection efficiency.

In conclusion, if we want to perform contactless finger knuckle recognition in a real-world scenario or an online scenario, the most critical problem comes from two aspects:how to match with high accuracy in real-world applications, and the second one is how to efficiently perform finger knuckle segmentation and correction. Therefore, our contribution can be summarized as below:

- We design a new loss function to solve finger knuckle rotation and translation problem for getting more robust features, called RSIL loss function. After comparison, our RSIL can increase matching perform when compare to STTL [24]. Even, our RSIL not only can be used on the finger knuckle identification, but also can be used on other biometric identification.
- Based on the YOLOv5, we use the CSL [52] to smooth angle classification, and add angular loss to prediction oriented bounding box for getting better quality segmented ROI. Meanwhile, we can use the angle information to normalize picture. From the segmented finger knuckle by YOLOv5-CSL, the matching perform can be increased when use these finger knuckle database offered. In comparison, such angular loss can significantly improve the performance for online finger knuckle identification.
- We design a cross-platform online finger knuckle identification system for completely contactless and online finger knuckle identification. The system can detect and match finger knuckle patterns from finger images acquired under complex background, using any ordinary smartphone or general laptop camera.

Section 3 will explain our designed RSIL loss function and prove that it can be differentiable by equation. Section 4 contains all experiments and corresponding results, including within database experiments and cross database experiments, and 2D finger knuckle and 3D finger knuckle. And also contain the ablation study with changing translation size and rotation size. Section 5 introduces the finger knuckle detection model and how to implement the detection of finger knuckle angle information. Section 6 is to prove the online finger knuckle identification performance.

3 Matching Contactless Finger Knuckle

We choose the Residual Feature Network (RFNet) [24] 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 [24] uses the soft-shifted triplet loss function, called SSTL to train the model and matching two features for dealing with palmprint shifting problem. As for the triplet loss function [34], it has been a great success in the field of biometrics recognition. However, in generally, the contactless finger knuckle of the same subject will not only just shift, but also will have local deformable transformation with rotation which is a common problem in the contactless biometrics identification. For solving it, we propose a new loss and also a new matching method. With our proposal loss function, the feature extraction backbone can learn the most robust features that can be rotation invariant, because our loss will get the minimal MSE between two feature maps after translating along the x-axis and y-axis, and rotating clockwise and counterclockwise with hyperparameter. In other words, we can get the minimum value regardless of how the features are rotated, therefore, we call our new loss function rotation and shift invariant triplet loss function (RSIL).

3.1 Rotation and Shift Invariant 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 RSIL loss function. Because our neural networks were trained using the architecture of triplet network [34], we used RSIL as loss function to update convolutional kernel of our models.

In generally, the RSIL is still a variant of triple loss, so that the RSIL 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.

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

In order to adapt to tasks with different degrees of rotation and translation, and balance performance and speed, we set translation and rotation parameter as a hyperparameter. The $L(T_1, T_2)$ will get the minimal distance of two templates $D_{s_w,s_h,\theta}(T_1, T_2)$ after shift and rotation in the range $-S_W \leq s_w \leq S_W$, $-S_H \leq s_h \leq S_H$, $-\Theta \leq \theta \leq \Theta$, called minimal rotate and shift distance (MRSD). Meanwhile, the distance $D_{s_w,s_h,\theta}(T_1, T_2)$ calculates the pixel-wise MSE value when template T_1 is shifted s_w pixel along x-axis and s_h pixel along y-axis and rotated θ angle in the Equation 3.

$$L(T_1, T_2) = \underset{-S_W \leq s_w \leq S_W, -S_H \leq s_h \leq S_H, -\Theta \leq \theta \leq \Theta}{min} D_{s_w,s_h,\theta}(T_1, T_2) \tag{2}$$

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

In terms of $C_{s_w,s_h,\theta}$, it represents the common region between two templates after one template shifted along x-axis with s_w, shifted along y-axis with s_h, and rotated with θ, as showed in the Figure 1. When templates

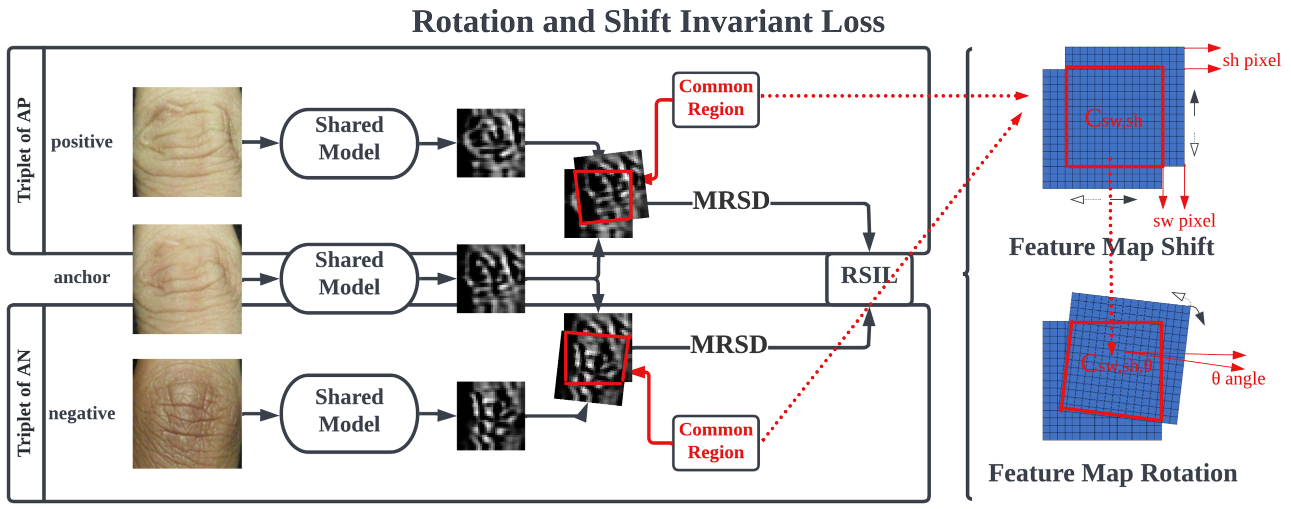


Figure 1: An overview of how to use our rotation and shift invariant loss function (RSIL) to train the triplet neural network. We will use the minimal rotate and shift distance (MRSD) to calculate the similarity of two output templates, the common region is the red box after shifting and rotating. During matching process instead of training process, we also use the MRSD to calculate matching scores.

are rotated and shifted, just affine transformation, the new $T^{(s_w,s_h,\theta)}[x,y]$ template can be sampled from the source $T[n,m]$ depending on the sampling method as the Equation 4, the H and W is the height and width of the $T[n,m]$. At here, we use the $[n,m]$ for representing the original pixel position before affine transformation.

$$T^{(s_w,s_h,\theta)}[x,y] = \sum_{n}^{H}\sum_{m}^{W} T[n,m]p(x-m)q(y-n) \tag{4}$$

$p()$ is the interpolation method along the x-axis, $q()$ is the interpolation method along the y-axis. At here, if we use the bilinear interpolation method, the above equation can be written as the Equation 5. Then the partial derivatives of $T^{(s_w,s_h,\theta)}[x,y]$ with respect to $T[x,y]$ can also be written as the Equation 6.

$$T^{(s_w,s_h,\theta)}[m,n] = \sum_{n}^{H}\sum_{m}^{W} T[n,m]max(0,1-|x-m|)max(0,1-|y-n|) \tag{5}$$

$$\frac{\partial T^{(s_w,s_h,\theta)}[x,y]}{\partial T[x,y]} = \sum_{n}^{H}\sum_{m}^{W} max(0,1-|x-m|)max(0,1-|y-n|) \tag{6}$$

As for the (T_a, T_p) pair, we can assume when the T_a is rotated angle of θ_{ap} and shifted with $(sw_{ap}$, $sh_{ap})$ pixels, $T_i^{a(sw_{ap},sh_{ap},\theta_{ap})}[x,y]$, can get the minimal $D_{sw_{ap},sh_{ap},\theta_{ap}}(T_a,T_p)$, then $L(T_a,T_p) = D_{sw_{ap},sh_{ap},\theta_{ap}}(T_a,T_p)$. Meanwhile, with the $(sw_{an}, sh_{an}, \theta_{an})$, $T_i^{a(sw_{an},sh_{an},\theta_{an})}[x,y]$, the (T_a,T_n) pair can get the minimal $D_{sw_{an},sh_{an},\theta_{an}}(T_a,T_n)$, therefore, the partial derivatives of RSIL with respect to T_i^p and T_i^n are showed as below.

$$\frac{\partial RSIL}{\partial T_i^p} = \begin{cases} 0, if(x,y) \notin C_{swap,shap,\theta_{ap}} \ or \ RSIL = 0 \\ -\frac{2(T_i^{a(swap,shap,\theta ap)}[x,y]-T_i^p[x,y])}{N*|C_{swap,shap,\theta_{ap}}|}, otherwise \end{cases} \tag{7}$$

$$\frac{\partial RSIL}{\partial T_i^n} = \begin{cases} 0, if(x,y) \notin C_{sw_{an},sh_{an},\theta_{an}} \ or \ RSIL = 0 \\ \frac{2(T_i^{a(sw_{an},sh_{an},\theta_{an})}[x,y] - T_i^n[x,y])}{N*|C_{sw_{an},sh_{an},\theta_{an}}|}, otherwise \end{cases} \tag{8}$$

As for the $T_i^a[x,y]$ partial derivation, because we shift and rotate the anchor template, then the anchor template will be interpolated to generate the new $T_i^{a(sw_{ap},sh_{ap},\theta_{ap})}[x,y]$ and $T_i^{a(sw_{an},sh_{an},\theta_{an})}[x,y]$, result in the result as below.

$$\frac{\partial RSIL}{\partial T_i^a} = \begin{cases} 0, if(x,y) \notin (C_{sw_{an},sh_{an},\theta_{an}} or C_{sw_{ap},sh_{ap},\theta_{ap}}) \ or \ RSIL = 0 \\ \frac{2(T_i^{a(sw_{ap},sh_{ap},\theta_{ap})}[x,y] - T_i^p[x,y])}{N*|C_{sw_{ap},sh_{ap},\theta_{ap}}|} * \frac{\partial T_i^{a(sw_{ap},sh_{ap},\theta_{ap})}[x,y]}{\partial T_i^a[x,y]} + \\ \frac{-2(T_i^{a(sw_{an},sh_{an},\theta_{an})}[x,y] - T_i^n[x,y])}{N*|C_{sw_{an},sh_{an},\theta_{an}}|} * \frac{\partial T_i^{a(sw_{an},sh_{an},\theta_{an})}[x,y]}{\partial T_i^a[x,y]}, otherwise \end{cases} \tag{9}$$

After affine transformation, the partial derivation of $\frac{\partial T_i^{a(sw_{ap},sh_{ap},\theta_{ap})}[x,y]}{\partial T_i^a[x,y]}$ and $\frac{\partial T_i^{a(sw_{an},sh_{an},\theta_{an})}[x,y]}{\partial T_i^a[x,y]}$ as show in the Equation 6. Now the T_i^a, T_i^p, and T_i^n is the output feature of models, then the RSIL loss will backpropagation with chain rule to update the learnable parameters when train neural networks. In terms of testing process or matching process, the MRSD will be used to calculate the matching score.

4 Experiments and Results

We choose the baseline model is the RFNet [24], its performance can outperform DenseNet-BC [12], CompCode [16], DoN [57], Ordinal Code [37], and RLOC [14] algorithms on the palmprint verification problem from the [24] experiments. For proving our RSIL loss function performance, we will compare its performance with Soft-Shift Triplet (STTL)[24] loss function on different public finger knuckle database based on the RFNet [24]. With RSIL loss function, the RFNet is represented by RFNet-RSIL, on the contrary, RFNet-STTL represents with STTL loss function. Compare to convolution layer or dilated convolution [53], the deformable convolution [58] 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) [42] as the pretrained weights.

Meanwhile, we will also compare with the FKNet [5] which get the state-of-the-art performance on 3D finger knuckle identification, and EfficientNetV2-S [38]. FKNet performance on the 3D finger knuckle database, 2D finger knuckle and even palmprint database can over SGD [4], CR_L1_DALM, CR_L2 [55], ResNet-50 [8], VGG-16 [35], AlexNet [17], DenseNet-121 [12], and SE-ResNet-50 [11]. 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 RSIL 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 RSIL and STTL. The changed EfficientNetV2-S model with RSIL called EfficientNetV2-S-RSIL, 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.

Algorithm 1 Comparing Contactless Finger Knuckle Using Trained Model

```
Input: $\boldsymbol{I_1}, \boldsymbol{I_2}$ ← input two images with dimension $h * w * c$ (i.e. $128 * 128 * 3$); s ← down sampling factor of the network (i.e. 4 of RFNet); s ← down sampling factor of the network (i.e. 4 of RFNet);
    SH ← shift template along the y-axis range;
    SW ← shift template along the x-axis range;
    $\Theta$ ← rotate template with angle range;
Output: MRSD: final matching score;
Convolve $I_1$ and $I_2$ with trained neural network to get two templates with dimension $\frac{h}{s} * \frac{w}{s} * 1$;
for $sh \in (-SH, SH)$ do
    for $sw \in (-SW, SW)$ do
        for $\theta \in (-\Theta, \Theta)$ do
            Calculate the $I_1$ pixel-wise MSE with $I_2$ after shifting (sh, sw) and rotating $\theta$
        end for
    end for
end for
Get the minimal MSE as the MRSD matching score between $I_1$ and $I_2$
```

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 on the Table 2. From the table, the RFNet costs the minimal time for extracting feature of finger knuckle crease with 0.01957s. And the matching time of all of these models with RSIL is longer than these models with STTL, because RSIL will not only shift but also rotate feature maps for getting the minimal matching score. However, the EfficientNetV2-S and FKNet do not shift or rotate feature maps during matching process, result in the shortest matching time with 0.00002s.

Model	Prams (M)	Input Size	Template Size	FLOPs (B)	Feature Extraction (s)	Matching (s)
DeConvRFNet-STTL	0.36M	$128 * 128$	$32 * 32$	1.29B	0.02880s	0.01212s
DeConvRFNet-RSIL	0.36M	$128 * 128$	$32 * 32$	1.29B	0.02880s	0.02250s
EfficientNetV2-S [38]	20.18M	$300 * 300$	$9 * 9$	5.40B	0.07501s	0.00002s
EfficientNetV2-S-STTL	20.00M	$300 * 300$	$9 * 9$	5.38B	0.07135s	0.00911s
EfficientNetV2-S-RSIL	20.00M	$300 * 300$	$9 * 9$	5.38B	0.07135s	0.02181s
FKNet [5]	7.28M	$96 * 64$	$8 * 12$	0.28B	0.04492s	0.00002s
RFNet-STTL [24]	0.46M	$128 * 128$	$32 * 32$	1.39B	0.01957s	0.01212s
RFNet-RSIL	0.46M	$128 * 128$	$32 * 32$	1.39B	0.01957s	0.02250s

Table 2: Comparison time and space complexity of different neural network. Time complexity is the average time on the Ubuntu 22.04 with GeForce RTX 2080 GPU and I7-7800X CPU. For the STTL, the shift size is 4; for the RSIL, the shift size is 4, and add rotation with 4 angles. The FKNet and EfficientNetV2-S serial models obey their same input image size of the original paper [5] and [38], respectively.

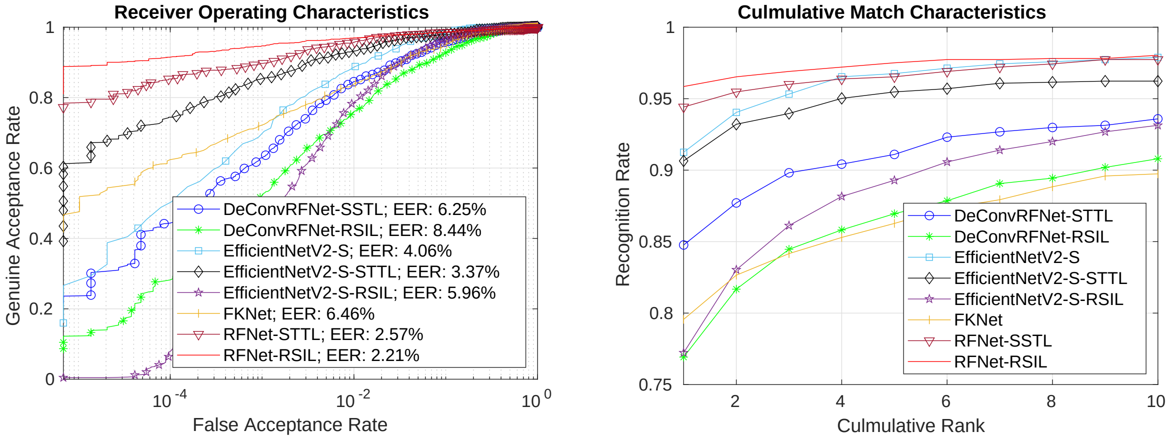
4.2 Within Database Performance Evaluation

4.2.1 Contactless Finger Knuckle Image Database (Version 3.0)

The finger knuckle database [43] 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 2, 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 RSIL 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-RSIL can arrive at 2.21%. And as top-2 ranking, the RFNet-RSIL 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)

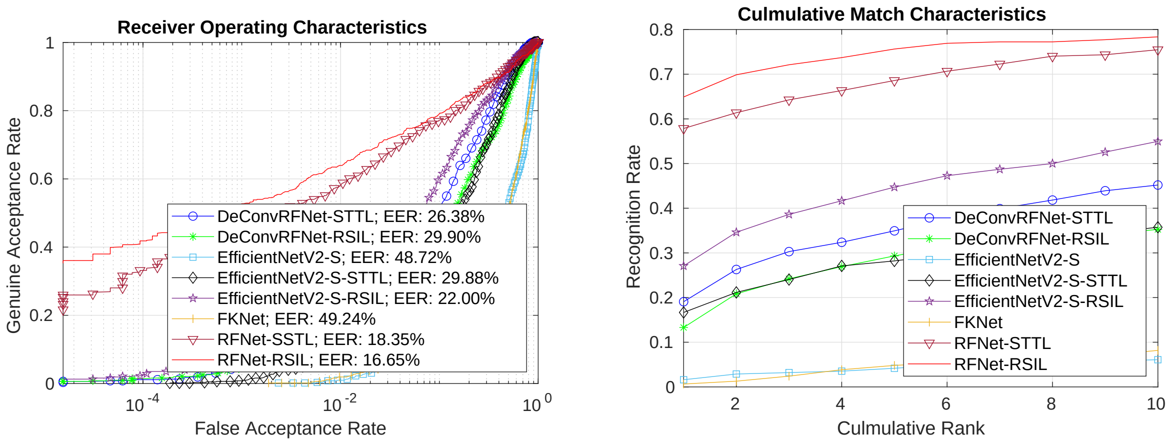
Figure 2: Comparative ROC (a) and corresponding CMC (b) for one-session on the contactless finger knuckle image database [43].

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 RSIL 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-RSIL can arrive at 0.7. From the ROC and CMC Figure 3, we can also get that the STTL and RSIL triplet loss function are better than classification task, because the FKNet and EfficientNetV2-S have the lowest accuracy.

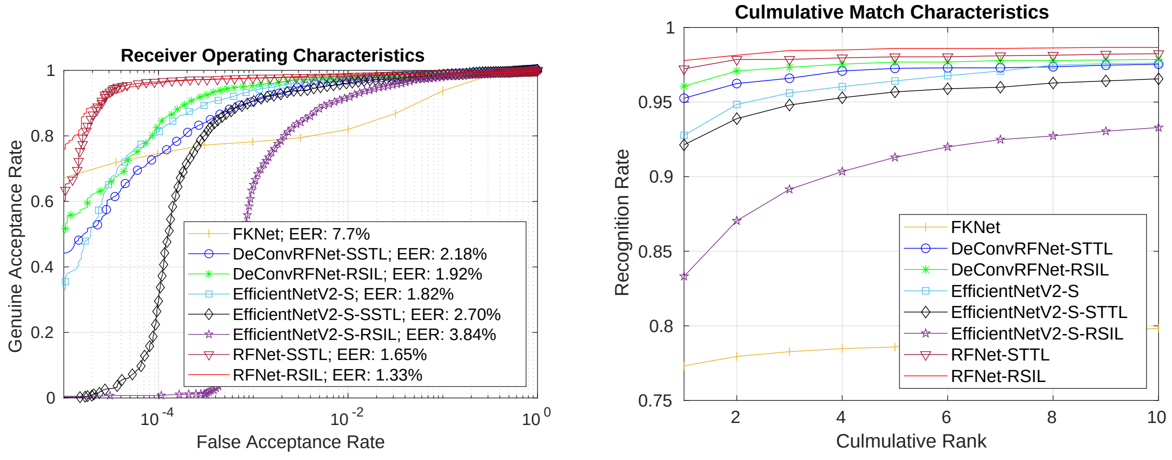
4.2.2 Index Finger Knuckle of Contactless Hand Dorsal Image Database

As for the experiment, the dataset [44] 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 [5].



(a) (b)

Figure 3: Comparative ROC (a) and corresponding CMC (b) for two-session on the contactless finger knuckle image database [43].



(a) (b)

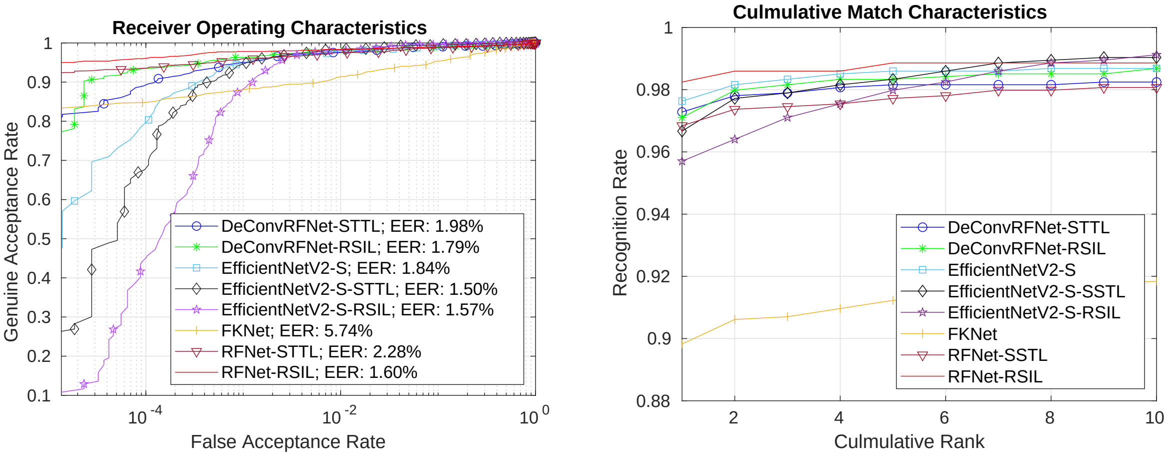
Figure 4: Comparative ROC (a) and corresponding CMC (b) for one-session on the contactless hand dorsal image database [44].

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-RSIL and RFNet-STTL is similar, but the RFNet-RSIL is slightly better than RFNet-STTL depend on the EER value on ROC. And on the CMC, the RFNet-RSIL 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 [41] 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 RSIL loss still get the best performance with 1.60% EER, even for the recognition rate on the CMC.



(a) (b)

Figure 5: Comparative ROC (a) and corresponding CMC (b) for one-session on the 3D finger knuckle database[41].

4.3 Cross Database Performance Evaluation

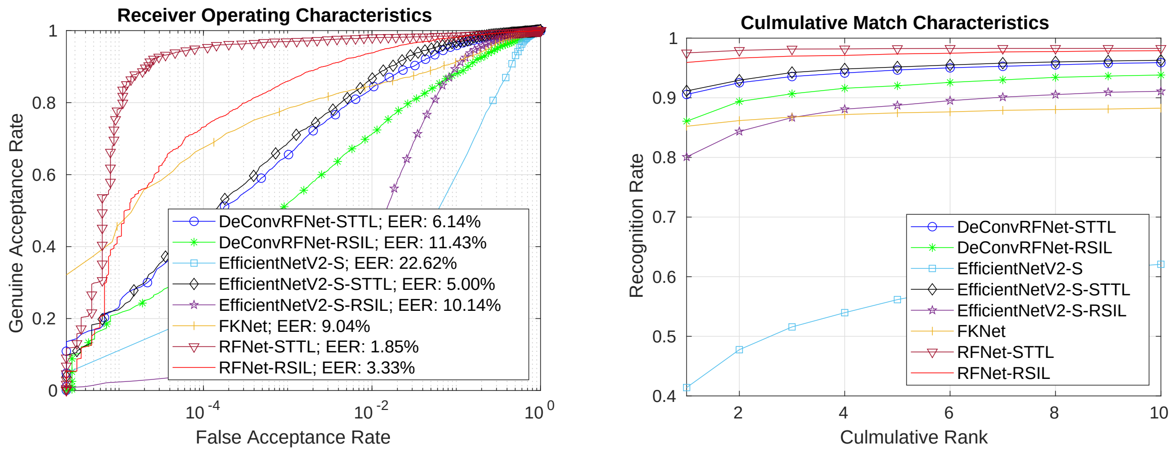
From the within database experiment, we can clearly get a conclusion that the RSIL loss function can increase the performance compare to the STTL loss function, and the RFNet is better than the DeConvRFNet, EfficientNetV2-S, and FKNet. Meanwhile, the FKNet performance is the worst one. In this section, 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 deformation). In the next step, we use our models to test all the finger knuckle of the Hand Dorsal Images Database and the Tsinghua Finger Vein and Finger Dorsal Texture Database (THU-FVFDT3) [45]. 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 (THU-FVFDT-FDT3_Train) as our evaluation dataset.

4.3.1 Hand Dorsal Images Database

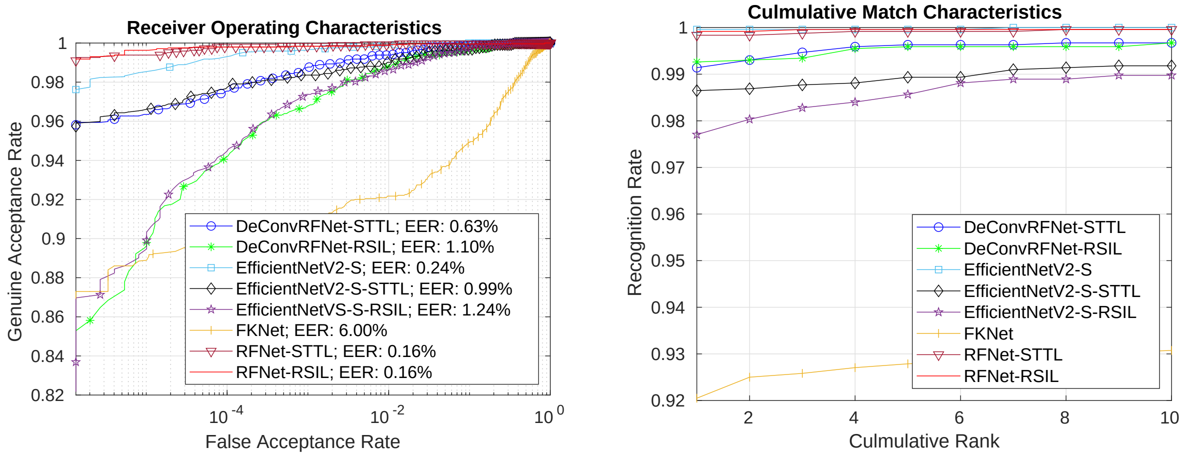
Index Finger Knuckle and Middle Finger Knuckle

The database totally has 712 subjects, and each subject has 5 samples of hand dorsal image. 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 6 is the performance result on the index finger. From Figure 6, all models' cross database performance is similar on the database regardless which finger. We should also notice that STTL is better than RSIL on the cross database experiment, while within database, the RSIL is better than STTL. It shows that the generalization ability of RSIL 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.



(a) (b)

Figure 6: Comparative ROC (a) and corresponding CMC (b) for one-session of the index finger knuckle on the contactless hand dorsal image database [44].



(a) (b)

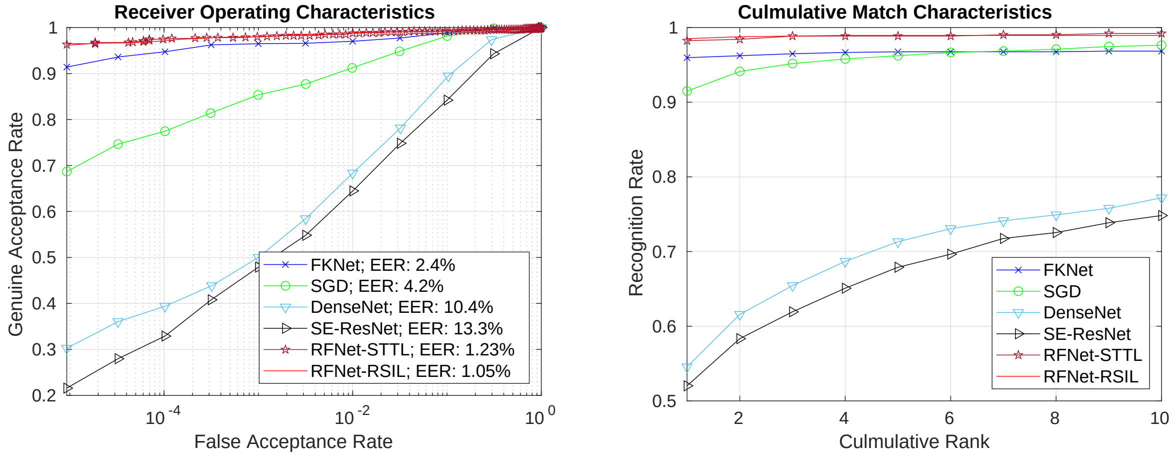
Figure 7: Comparative ROC (a) and corresponding CMC (b) for one-session of the finger dorsal texture images [45].

4.3.2 Tsinghua Finger Vein and Finger Dorsal Texture Database

The database [45] 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 RSIL 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

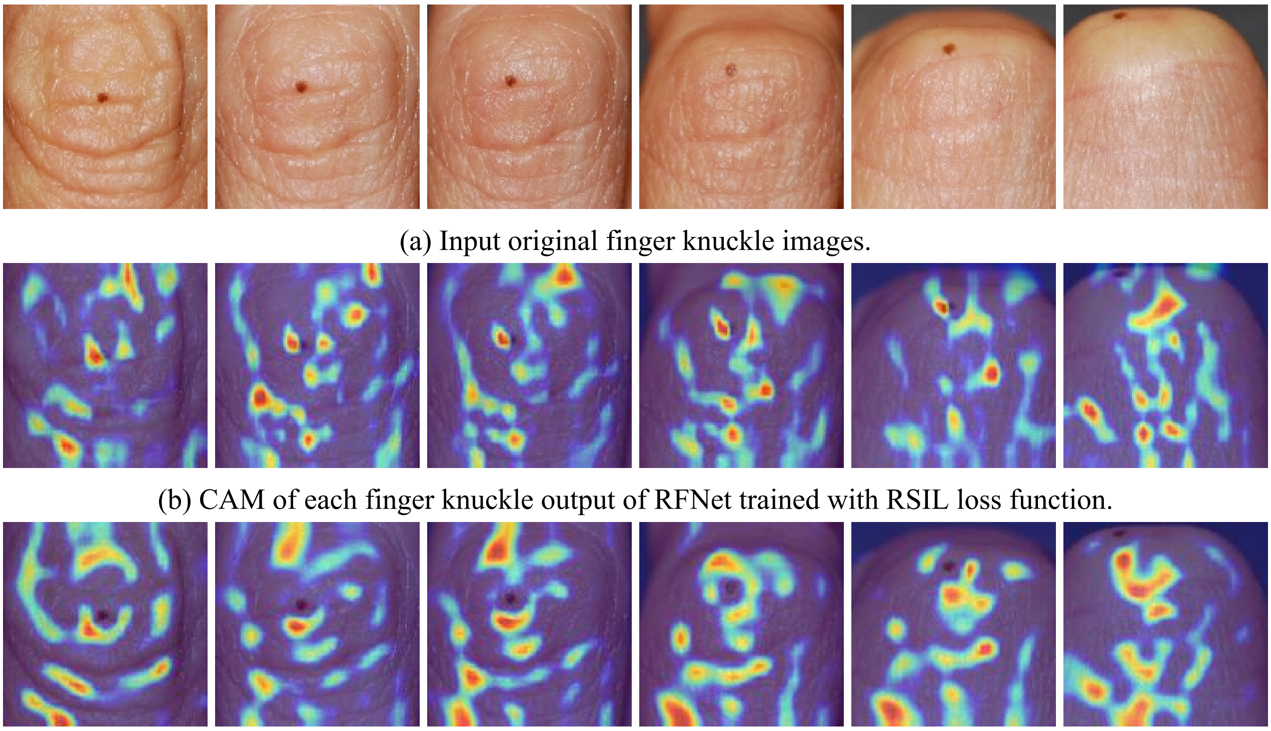
The 3D finger knuckle images database [41] can offer robust 3D information which can be invariant to changed illuminations, for example, the depth information of the crease of finger knuckle. With the 3D finger knuckle database, the FKNet is the state-of-the-art. Meanwhile, RFNet with RSIL loss function can get the best performance on the within database experiments and cross database experiments when compare to the FKNet on the 2D finger knuckle database. Therefore, we compare the RFNet with FKNet on the database to show the identification performance on 3D finger knuckle database. As for the protocol, it will generate $190 * 6 = 1140$ genuine matching scores, and $190 * 189 * 6 = 215,460$ imposter matching scores from matching matrix. From the Figure 8, RFNet-RSIL still can get the best performance for finger knuckle verification and identification. Form the ROC curve, the EER of the RFNet-RSIL can increase to 1.05% while the EER of the FKNet is 2.4%. Not only on the 2D finger knuckle database, but also on the 3D finger knuckle database, the RFNet-RSIL can outperform the state-of-the-art results.



(a) (b)

Figure 8: Comparative ROC (a) and corresponding CMC (b) for one-session of the 3D finger knuckle database [41].

4.5 Discussion



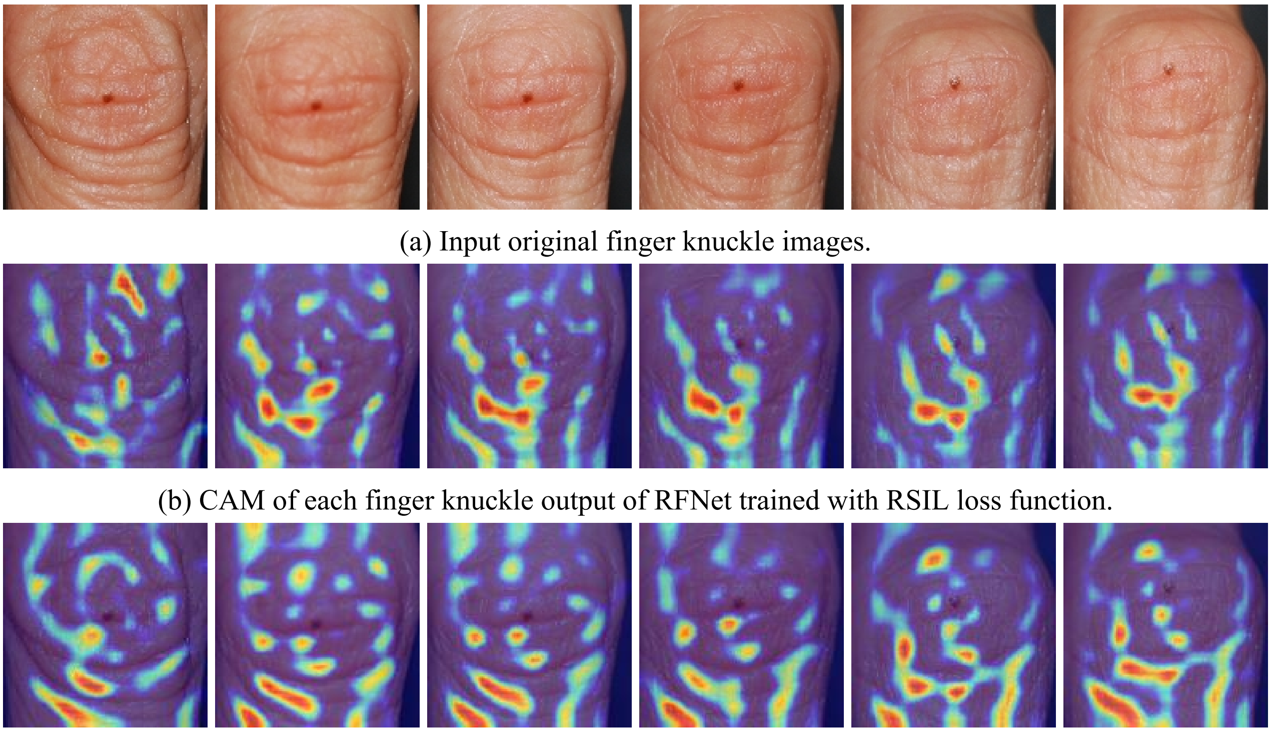
(a) Input original finger knuckle images.

(b) CAM of each finger knuckle output of RFNet trained with RSIL loss function.

(c) CAM of each finger knuckle output of RFNet trained with SSTL loss function.

Figure 9: Show the class activation maps for the first session samples of 104th subject of the HKPolyU Finger Knuckle Images Database [43] with RFNet.

We have compared the identification performance of RFNet with EfficientNetV2-S, DeConvRFNet, and FKNet on the 2D and 3D finger knuckle database, and on the within database and cross database experiments.



(a) Input original finger knuckle images.

(b) CAM of each finger knuckle output of RFNet trained with RSIL loss function.

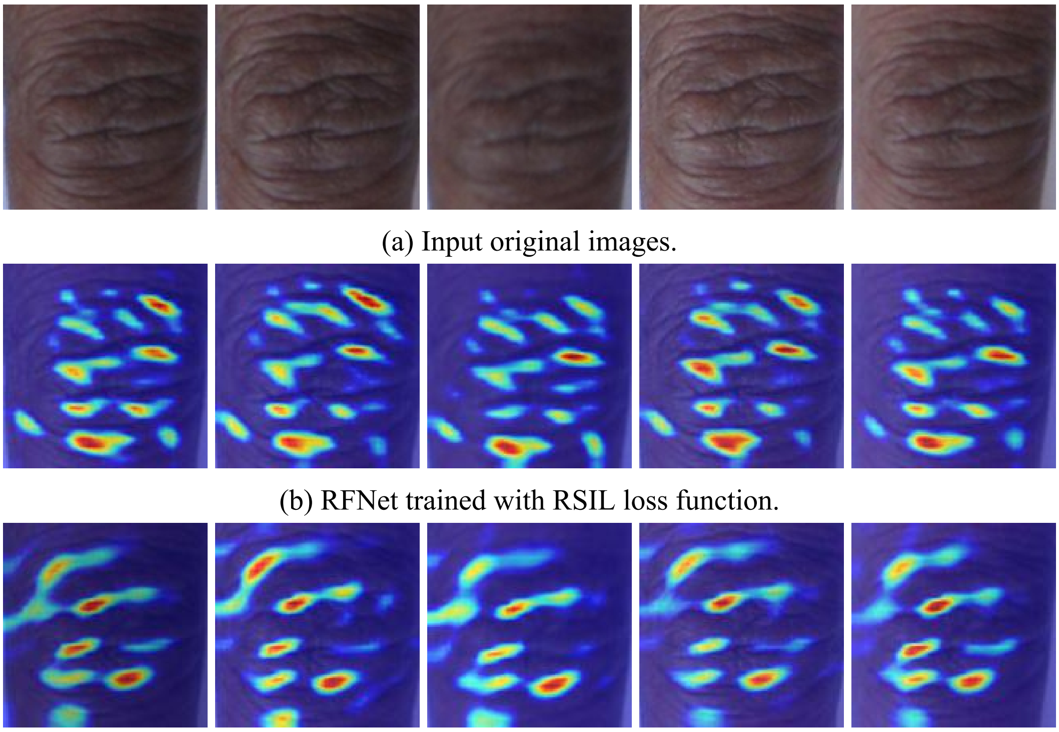
(c) CAM of each finger knuckle output of RFNet trained with SSTL loss function.

Figure 10: Show the class activation maps (CAM) for the second session samples of 104th subject of the HKPolyU Finger Knuckle Images Database [43] with RFNet.

Due to deeply learned residual features of the RFNet which have already outperformed the state-of-the-art results on the palmprint, it still can get the best performance on the finger knuckle crease when compare to the rest models from above experiment results. Because finger knuckle is very prone to flexing, causing crease texture distortion, if just shift the template, it cannot solve the deformation problem with rotation. Therefore, we design our new RSIL to further solve the problem. With our RSIL loss function when train RFNet and MTRD when matching finger knuckle, the RFNet can increase matching accuracy regardless on the ROC and CMC based on the STTL. Especially on the Finger Knuckle Images Database (Version 3.0) which offer bending finger knuckle with complexity deformation, RFNet-RSIL improved performance is relatively more compare to other database from the Figure 2 and Figure 3. Two-session protocol is more complexity because of changing finger knuckle crease and more complexity deformation when matching process. Form the Figure 3 (a) ROC and (b) CMC, our RSIL with RFNet get the best matching and recognition performance. Meanwhile, our RSIL not only work on the RFNet, but also can work on the DeConvRFNet from the Figure 4 and Figure 5, with RSIL loss function, the matching and recognition performance also can increase.

From these experiment results, we can also get a conclusion is that EfficientV2-S model is better than FKNet from the within database experiment, and even on the Tsinghua finger knuckle database. EfficientNetV2 model can outperform the ResNet on the ImageNet [33], in other words, the EfficientNetV2 model can extract robust feature than ResNet on the ImageNet. Because EfficientNetV2 replace the residual block with inverted residual block, and use MBConv as a block unit. As for MBConv block, it uses depth-wise convolutions to decrease training weights and use Squeeze-Excited block as channel attention. Meanwhile, the depth of EfficientNetV2-S is deeper than the FKNet. On the contrary, the FKNet use the ResNet-50 fist conv3 as the feature extract model. EfficientNetV2 use the more light, advance and efficient module than ResNet.

However, RSIL generalization ability is lower than STTL loss from the cross database experiment, except on the Tsinghua database. On the cross hand dorsal database, Figure 6, EER of RFNet with RSIL performance will drop from about 2.0% to 5.0%. As for the rest model, performance with RSIL also drop with corresponding value when compare to STTL. But in the within database experiment, these model with RSIL loss is better than STTL loss. It shows our RSIL can affect the back propagation during training process to the different model weights, in other words. And another phenomenon is that EfficientNetV2-S with SSTL and RSIL cannot work. From the Table 2, if the input image size is 300x300, EfficientNetV2-S with STTL and RSIL will output 9x9 template size. The output feature size is too small when use the STTL and RSIL loss function, inversely, the performance will drop while translation and rotation.



(a) Input original images.

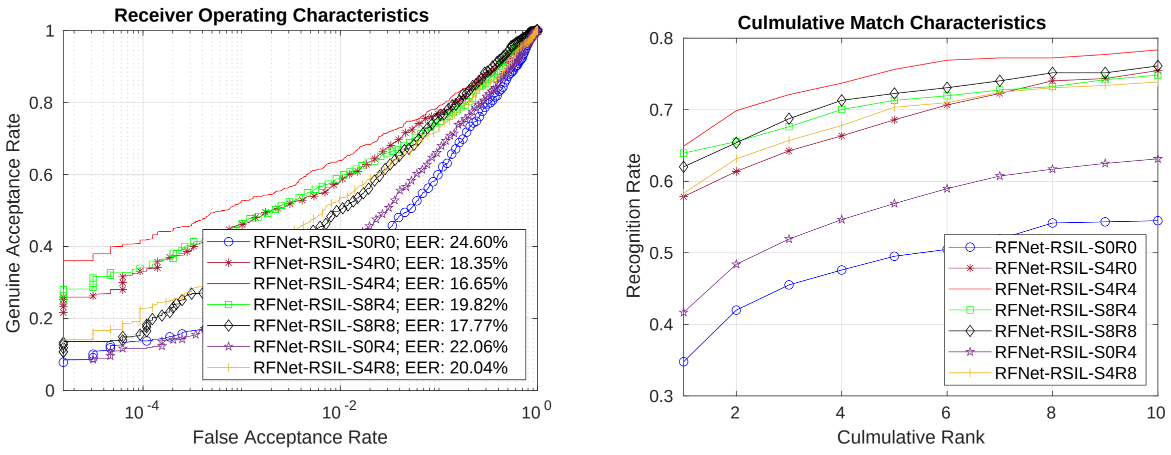
(b) RFNet trained with RSIL loss function.

(c) RFNet trained with SSTL loss function.

Figure 11: Show the class activation maps for one subject samples of the HKPolyU Hand Dorsal Images Database [44] with RFNet.

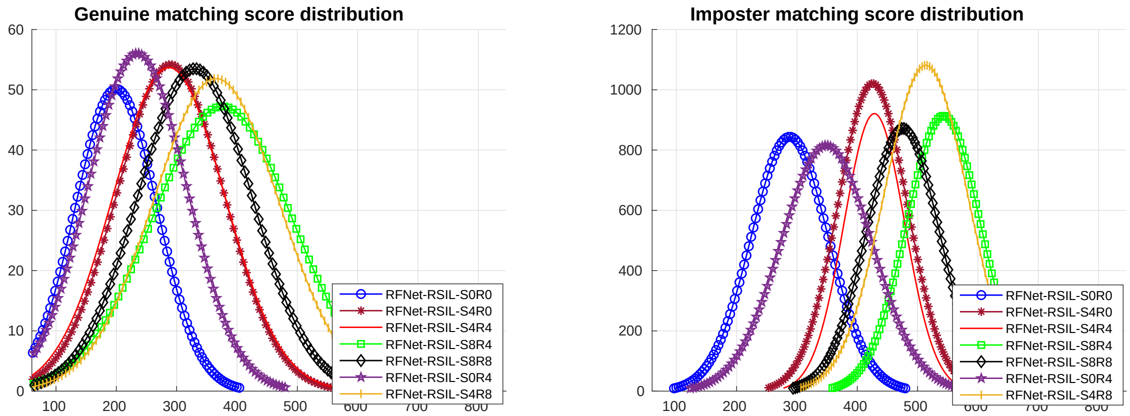
4.6 Ablation Study

One of the ablation study is change the shift and rotation hyperparameter to show the performance with the two-session protocol on the deformable finger knuckle database [43]. We change the shift size from 0 to 8, and the rotation angle also from 0 to 8, and the interval value is 4. From the Figure 12, we can get from the ROC and CMC



(a) (b)

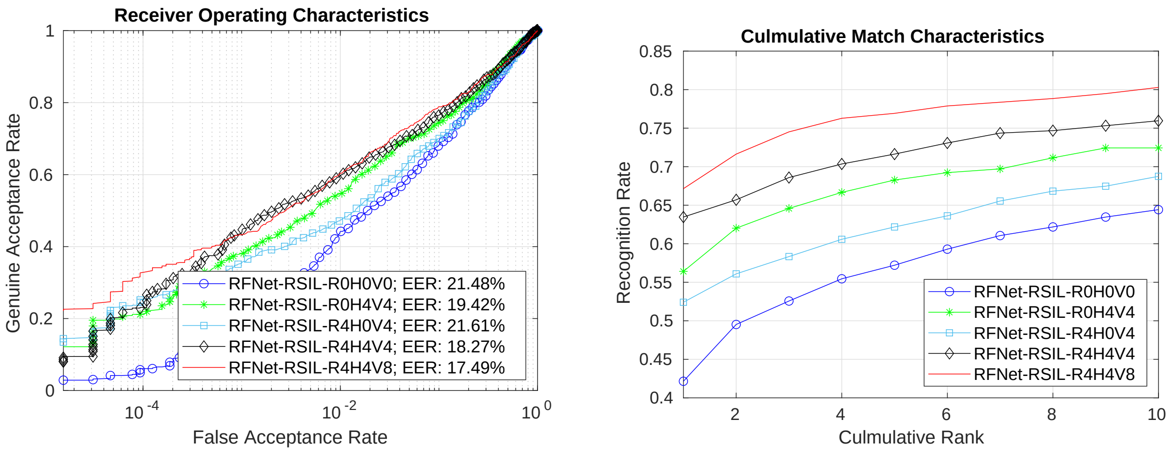
Figure 12: Comparative ROC (a) and corresponding CMC (b) for two-session of the Finger Knuckle Database (Version 3.0) [43]. For approving our RSIL loss function efficiency, we change the translation parameter and rotation parameter to show the different matching performance.



(a) (b)

Figure 13: Genuine matching score distribution (a) and Imposter matching score (b) for two-session of the Finger Knuckle Database (Version 3.0) [43]. For approving our RSIL loss function efficiency, we change the translation parameter and rotation parameter to show the different matching performance.

The other ablation study is change the input image size. From segmented finger knuckle by YOLOv5x-CSL, the minimal size of ROI is about $189 * 224$. Therefore, we keep the same ratio to change all segmented finger knuckle to $184 * 208$ due to rectangle bounding boxes. And in this kind of situation, we change the vertical and horizontal size with different size. And we make the vertical shifting size bigger than horizontal shifting size, get the result in the Figure 14.



(a) (b)

Figure 14

5 Online Contactless Finger Knuckle Identification

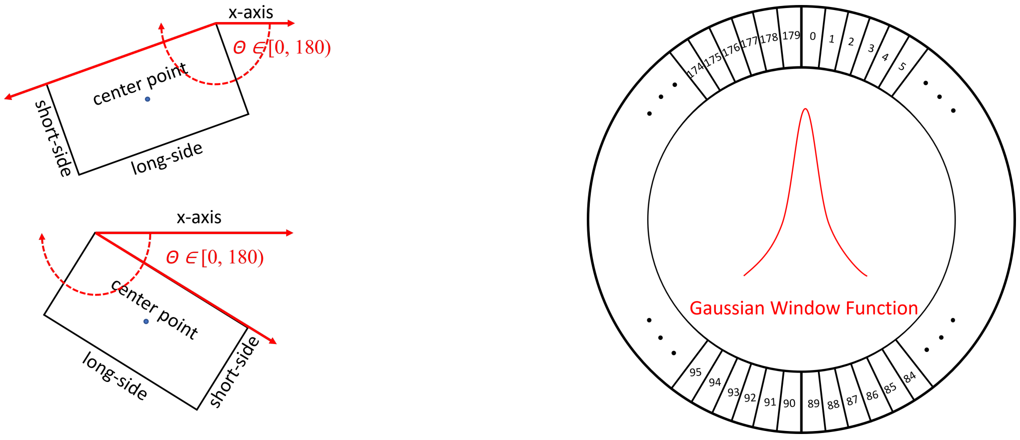
With RSIL loss, the RFNet [24] can outperform state-of-the-art methods. In the previous section, we have estimated its verification and identification performance on different public finger knuckle database, including within-db and cross-db experiments. As for a completely contactless and online finger knuckle identification, the finger knuckle detector is a very important module for automatically detect and segment finger knuckle region. As for neural network, the current based on YOLO [31], [29], [30], [2], [46] and R-CNN [7], [6], [32], [9] series object detection and segmentation approaches cannot obtain the angle of finger knuckle and the segmentation with high precision. 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 use the CSL[52] to smooth angular classes for predicting oriented bounding box based on YOLOv5 model for segmenting and getting angle information.

5.1 Completely Contactless Finger Knuckle Detection

5.1.1 Oriented 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. It also provides different variant of YOLOv5 models for different requirements on the inference time and detection accuracy. Especially, the YOLOv5's [46] speed can meet our online detection requirements for real-time finger knuckle detection. 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, an oriented bounding box will be predicted instead of a horizontal bounding box.

As analyzed in this paper [52], the oriented bounding boxes loss will mainly come from angular periodicity and the exchangeability of edges, if we use the $90°$ OpenCV oriented bounding boxes definition. When use the long side definition of oriented bounding box with $180°$, as shown on the Figure 15 (a), it can deal with the exchangeability of edges problem. Meanwhile, using classification task to predict angle can make model easier to train when compare to regression. A periodic coding method called Circular Smooth Label (CSL) [52] soft coding with Gaussian window function can solve two problems. One is that one-hot cannot distinguish class relationship, and another one is that the predicted angle have periodicity. Formula 10 $g(x)$ is the window function to smooth one-hot label, and r is a window function of the radius, in the paper $r = 6$ can get the best performance. Therefore, we used the Gaussian function for the Equation 10 window function, a commonly available function, and used a window radius of 6 to smooth the labels. And the



(a) Five-parameter definition with 180° angular range.

(b) Circle smooth label with Gaussian window function.

Figure 15: Oriented rectangle definition and the circle smooth label method.

period of this function is 180°, which is the same as the period of the predicted angle, as shown on the Figure 15 (b).

$$CSL(x) = \begin{cases} g(x), & \theta - r < x < r + \theta \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

The original YOLOv5 loss function can have three components. The formula can be simply written as $Loss = Loss_{bbox} + Loss_{obj} + Loss_{class}$. As for the $Loss_{bbox}$, the YOLOv5 uses the GIOU loss to calculate the loss between predicted bounding boxes and ground truth. Since the oriented 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 11, with the addition of $Loss_{angle}$ to YOLOv5 loss function. We use the binary cross-entropy to calculate the angle classification loss. In the Equation 12, the S^2 is the output grid size, and the I_{ij}^{obj} represents just calculate the loss when object fall in the grid cell.

$$Loss = Loss_{bbox} + Loss_{obj} + Loss_{class} + Loss_{angle} \tag{11}$$

$$Loss_{angle} = \sum_{i=0}^{S^2} I_{ij}^{obj} \sum_{a \in [0,180)} [P_i\hat{(a)} log(P_i(a)) + (1 - P_i\hat{(a)}) log(1 - P_i(a))] \tag{12}$$

5.1.2 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 [47] 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 [48]. 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) [42], and 64 finger knuckles images from the HKPolyU Hand Dorsal Database [44] as for the training set. And we use the rest data as testing set to evaluate performance. The most important part is the data augmentation which contains flip, rotation, resize, translate and mosaic.

5.1.3 Contactless Finger Knuckle Detection

Detection Performance

The YOLOv5x, and YOLOv5m model predict horizontal bounding box, while the remaining YOLOv5 model predict oriented bounding box with CSL to smooth angular classification, called YOLOv5-CSL. We can see the performance difference between these variations of the YOLOv5 model from the Table 3. Among the downloaded 2580 images, 100 images were randomly selected as the testing set. The YOLOv5x-CSL can get the highest mAP value with 89.9, especially, the AP of minor finger knuckle can get 90.1. On the contrary, without predicting oriented bounding box, the YOLOv5x detection accuracy is lower 1.6 mAP when compare to YOLOv5x-CSL.

Model	Total Time/ms (1024x1024)	Number of Layers	mAP^{val} 0.5	AP of Major FK	AP of Minor FK
YOLOv5x-CSL	40.9ms	407	**89.9**	**89.6**	**90.1**
YOLOv5m-CSL	23.3ms	263	85.7	88.9	80.4
YOLOv5x	33.3 ms	407	87.3	86.5	88.0
YOLOv5m	12.1ms	263	84.8	84.5	85.1

Table 3: Comparison of the accuracy of the different models of the YOLO series for detecting finger knuckle. The calculated values of mAP were measured at a detection threshold of 0.001 as well as an IOU threshold of 0.5. All the experiments are test on the GTX 2080 GPU and I7-7800X CPU, and the total time includes the inference and NMS process.

Segmentation Performance

This section aim to compare quality of finger knuckle between YOLOv5x-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).

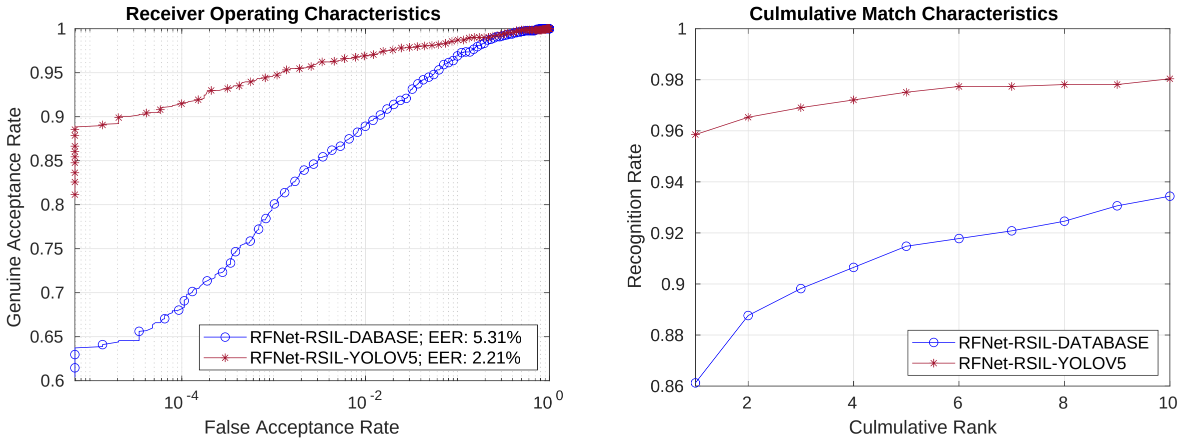


Figure 16: Compare performance on the Finger Knuckle V3 Dataset (with deformable)

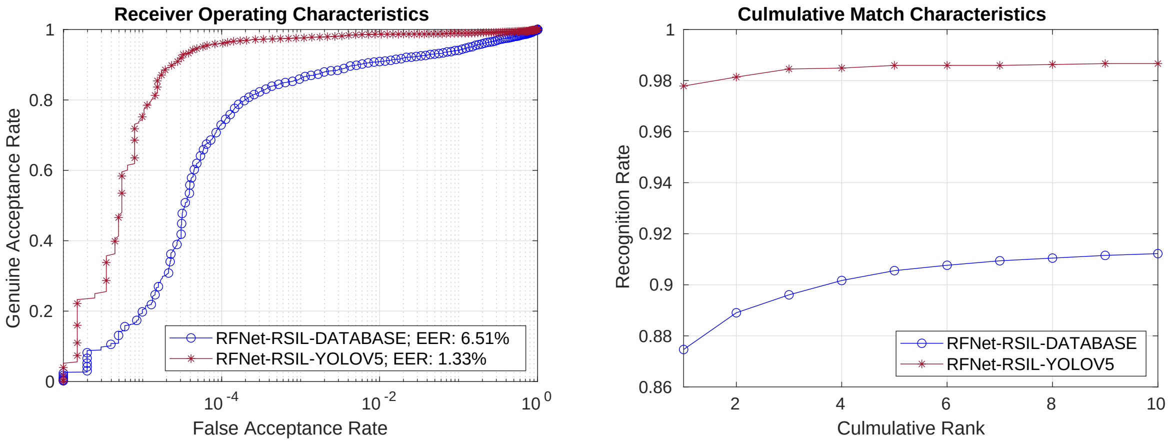


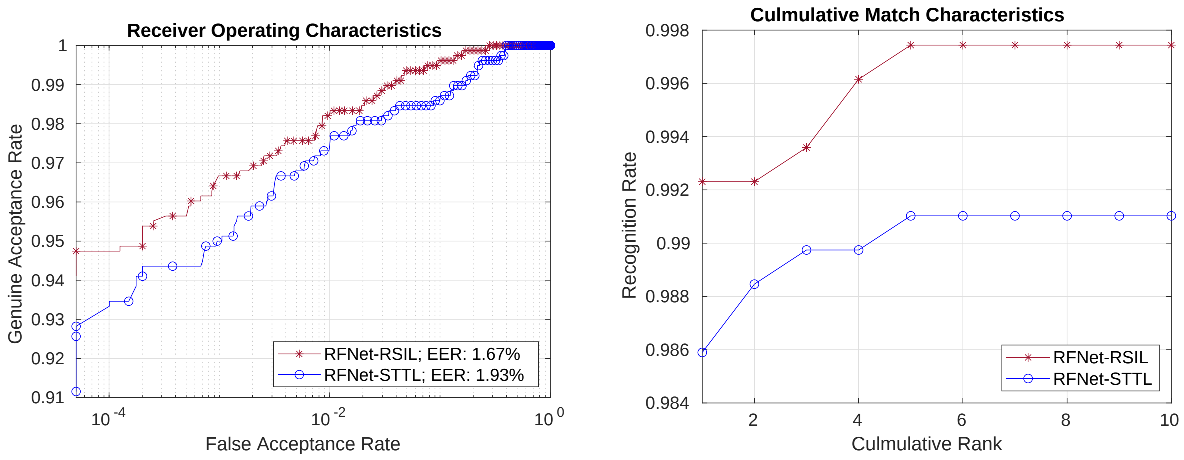
Figure 17: Compare performance on the Index Finger of the Hand Dorsal Image Database.

From the above ROC curve and ROC curve, we can clearly get the conclusion that quality of segmented finger knuckle of YOLOv5s-CSL is better than the segmented finger knuckle of dataset. When using YOLOv5x-CSL to segment finger knuckle, both of verification performance and recognition performance can increase. Especially on the Hand Dorsal Image Database, the EER value can drop from 6.51% to 1.33%, and on the CMC curve, the recognition rate of top 1 can increase from about 87.8% to 97.9%.

5.2 Online Contactless Finger Knuckle Identification Performance

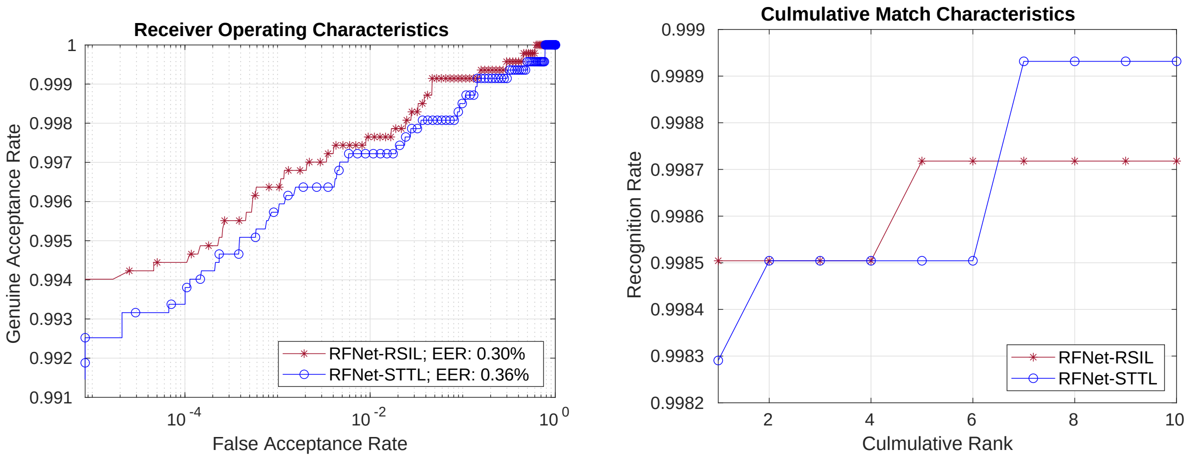
For proving our contactless and online finger knuckle identification performance, we capture 52 subjects and each subject can offer about 15-20s contactless finger knuckle video using smartphone. The minimum frame rate of the videos that we offer is 30 frames per second. When we capture these finger knuckle videos, we take into account practical application scenarios, therefore, each subject is on a complex background with traffic flow and pedestrian interference. And the finger knuckle rotate from 0 to 180 degrees and the distance from the camera changes, even capturing finger knuckle flexion scenes as well.

We choose two method to get the contactless finger knuckle images, one is that we get 1 image per second, another one is that we get 6 images per second for testing online matching performance. Because the shortest video is 15s, for 1 image per second, we can totally get $52 * 15 = 780$ finger knuckle images for keeping the same number of samples result in $52 * 15 = 1789$ genuine matching scores and $52 * 51 * 15 = 39780$ imposter matching scores. In terms of the 6 images per second, we can get $52 * 15 * 6 = 4680$ finger knuckle samples, result in $52 * 90 = 4680$ genuine matching scores and $52 * 51 * 90 = 238680$ imposter matching scores. From the below Figure 18 and 19, the online identification performance is good enough as an online finger knuckle identification system when using the RFNet to extract finger knuckle pattern. Both of them, the EER is lower than 2.00%, and the matching accuracy is higher than 93.00. At the same time, using RSIL loss, the matching performance can be further improved, although not by much.



(a) (b)

Figure 18: Comparative ROC (a) and corresponding CMC (b) with one session protocol for online finger knuckle video dataset with 1 frame image per second.



(a) (b)

Figure 19: Comparative ROC (a) and corresponding CMC (b) with one session protocol for online finger knuckle video dataset with 6 frame images per second.

6 Conclusions and Future Work

The generalization performance of convolutional neural networks is extremely powerful, and the features extracted by the convolutional kernel are so robust, for example with scale invariance, that neural networks have started to replace traditional feature description operators. However, neural networks do not have capabilities such as rotation invariance and translation invariance [13]. In practical application scenarios, contactless finger knuckle recognition system, are most susceptible to problems such as interference from complex backgrounds, and affine transformation such as rotation or translation of finger knuckle. Therefore, we propose the RSIL loss for increase the neural network rotating and shifting invariance performance. From the within database and cross database matching performance on the experiment section, it shows that with our RSIL loss, RFNet-RSIL can outperform state-of-the-art on the listed public finger knuckle database. We have also based the YOLOv5 model to resolve the interference of complex backgrounds and to obtain angular information to further mitigate the rotation problem. In Section 5, YOLOv5x-CSL can get a high mAP value on finger knuckle detection, and using it for segmenting finger knuckle can improve the matching accuracy.

Since we have design a completely contactless and online finger knuckle identification system, but we still have to solve several limitations. The first problem is the amount of data. For practical applications in various industries, the current amount of data on the finger knuckle is too sparse to allow the model to be adequately trained, unlike face recognition and fingerprint recognition which have such a large amount of data. To further improve performance is to increase the amount of data on the finger knuckle. Not only are the finger knuckle prone to rotation and translation, but an even greater headache is the problem of deformation, which occurs even in 3D space. In the deformable finger knuckle database [43], our method performance not very good. This is because the finger knuckle are partially flexible. Even though I used deformable convolution, it did not improve the matching performance. We think a future direction could deal with chunked feature maps or first extracting key points, such as ending points or bifurcation, and then getting a local match based on the location of these key points. Another problem is that our current approach is two-stage, where a network is trained to detect the position of the finger knuckle in order to segment. Then, another network is used to extract the finger knuckle features for matching. In fact, the finger knuckle features are already extracted during finger knuckle detection, therefore, why not use it to indirectly match the finger knuckle as an end-to-end model. Or do multitask model instead of such a tedious process.

7 References

[1] Ejaz Ahmed, Michael Jones, and Tim K Marks. “An improved deep learning architecture for person re-identification”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3908–3916.

[2] 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).

[3] KamYuen Cheng and Ajay Kumar. “Contactless finger knuckle identification using smartphones”. In: *2012 BIOSIG-Proceedings of the International Conference of Biometrics Special Interest Group (BIOSIG)*. IEEE. 2012, pp. 1–6.

[4] 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.

[5] 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.

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

[7] 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.

[8] 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.

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

[10] Pablo Hennings, Marios Savvides, and BVK Vijaya Kumar. “Verification of biometric palmprint patterns using optimal trade-off filter classifiers”. In: *International Conference Image Analysis and Recognition*. Springer. 2005, pp. 1081–1088.

[11] 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.

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

[13] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. “Spatial transformer networks”. In: *Advances in neural information processing systems* 28 (2015).

[14] 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.

[15] Rajiv Kapoor et al. “Completely Contactless Finger-Knuckle Recognition using Gabor Initialized Siamese Network”. In: *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*. IEEE. 2020, pp. 867–872.

[16] 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.

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

[18] Ajay Kumar. “Importance of being unique from finger dorsal patterns: Exploring minor finger knuckle patterns in verifying human identities”. In: *IEEE transactions on information forensics and security* 9.8 (2014), pp. 1288–1298.

[19] 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.

[20] Ajay Kumar and Ch Ravikanth. “Personal authentication using finger knuckle surface”. In: *IEEE Transactions on Information Forensics and Security* 4.1 (2009), pp. 98–110.

[21] Ajay Kumar and Zhihuan Xu. “Personal identification using minor knuckle patterns from palm dorsal surface”. In: *IEEE Transactions on Information Forensics and Security* 11.10 (2016), pp. 2338–2348.

[22] Yann LeCun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[23] Wei Liu et al. “Ssd: Single shot multibox detector”. In: *European conference on computer vision*. Springer. 2016, pp. 21–37.

[24] 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.

[25] Rajiv Mehrotra, Kameswara Rao Namuduri, and Nagarajan Ranganathan. “Gabor filter-based edge detection”. In: *Pattern recognition* 25.12 (1992), pp. 1479–1494.

[26] Abdallah Meraoumia, Salim Chitroub, and Ahmed Bouridane. “Fusion of finger-knuckle-print and palmprint for an efficient multi-biometric system of person recognition”. In: *2011 IEEE International Conference on Communications (ICC)*. IEEE. 2011, pp. 1–5.

[27] Abdallah Meraoumia, Salim Chitroub, and Ahmed Bouridane. “Personal Recognition by Finger-Knuckle-Print Based on Log-Gabor Filter Response”. In: ().

[28] Shubhangi Neware, Kamal Mehta, and AS Zadgaonkar. “Finger knuckle identification using principal component analysis and nearest mean classifier”. In: *International Journal of Computer Applications* 70.9 (2013).

[29] 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.

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

[31] 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.

[32] 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.

[33] Olga Russakovsky et al. “Imagenet large scale visual recognition challenge”. In: *International journal of computer vision* 115.3 (2015), pp. 211–252.

[34] 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.

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

[36] Arulkumar Subramaniam, Moitreya Chatterjee, and Anurag Mittal. “Deep neural networks with inexact matching for person re-identification”. In: *Advances in neural information processing systems* 29 (2016).

[37] 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.

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

[39] Ahmad S Tarawneh et al. “DeepKnuckle: Deep Learning for Finger Knuckle Print Recognition”. In: *Electronics* 11.4 (2022), p. 513.

[40] Daksh Thapar, Gaurav Jaswal, and Aditya Nigam. “Fkimnet: a finger dorsal image matching network comparing component (major, minor and nail) matching with holistic (finger dorsal) matching”. In: *2019 international joint conference on neural networks (IJCNN)*. IEEE. 2019, pp. 1–8.

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

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

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

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

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

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

[47] Unsplash. *Unsplash*. `https://unsplash.com/`.

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

[49] Ying Xin et al. “PAFNet: An Efficient Anchor-Free Object Detector Guidance”. In: *arXiv preprint arXiv:2104.13534* (2021).

[50] Wankou Yang, Changyin Sun, and Zhenyu Wang. “Finger-knuckle-print recognition using Gabor feature and MMDA”. In: *Frontiers of Electrical and Electronic Engineering in China* 6.2 (2011), pp. 374–380.

[51] Wenming Yang et al. “$alpha$-Trimmed Weber Representation and Cross Section Asymmetrical Coding for Human Identification Using Finger Images”. In: *IEEE Transactions on Information Forensics and Security* 14.1 (2018), pp. 90–101.

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

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

[54] David Zhang, Xiaoyuan Jing, and Jian Yang. *Biometric image discrimination technologies*. IGI Global, 2006.

[55] 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.

[56] Lin Zhang et al. “Online finger-knuckle-print verification for personal authentication”. In: *Pattern recognition* 43.7 (2010), pp. 2560–2571.

[57] 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.

[58] 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.