Towards Explainable and Unprecedented Accuracy in Matching Challenging Finger Crease Patterns

Supplementary Material

A. Additional Network Configuration Details

The layers of the modules or models in Fig. 2 are empirically determined. Firstly, the CNN backbone of the KnckleCreasePoint follows the same backbone as in [25]. Therefore, the dimension (length) of the feature $\boldsymbol{f}$ in the detected knuckle crease keypoint templates is 256. The number of self-attention and cross-attention layers in the KnuckllePointPair model is set to $l = 9$ and with $t = 20$ iterations. From the detected keypoint templates and estimated correspondences, only the top-k (k= 96) mutually matched correspondences are selected by sorting with scores. During the graph formation, nine neighboring nodes with k-NN are employed. Lastly, we empirically set $l = 3$ graph neural network layers for both the "Cross-Node and Self-Graph" module and the graph classification module.

The average image size (width $\times$ height) of detected finger knuckle by the detector [6] is 158$\times$185 for the dataset in [47], 385$\times$463 for the dataset in [11], and 599$\times$801 on our multi-pose dataset. To achieve a judicious trade-off between the model complexity and the performance, the input image size for training our CGN model on images from different finger knuckle datasets is automatically normalized to 152$\times$200.

B. Additional Details on Model Architecture

Cross-Node and Self-Graph: Current graph similarity models [28, 29, 31] use computationally expansive attention to compute node similarity along graph-level representation. More specifically, they add the node similarity that each node of G^{p} will be compared with all nodes of G^{g} with attentional mechanism based on the graph-level similarity [26] to improve the performance. The node similarity calculation and attention mechanism are computationally expansive. However, for the mutually matched pair $\boldsymbol{k}^{\mathrm{p}}_{i'}$ and $\boldsymbol{k}^{\mathrm{g}}_{j'}$, the cosine similarity between $\boldsymbol{h}^{\mathrm{p}}_{i'}$ and $\boldsymbol{h}^{\mathrm{p}}_{j'}$ already has high positive similarity, on Fig. 4. Meanwhile, the neighborhood node $\boldsymbol{k}^{\mathrm{p}}_{u}(\boldsymbol{k}^{\mathrm{p}}_{u} \in N(\boldsymbol{k}^{\mathrm{p}}_{i'}))$ and $\boldsymbol{k}^{\mathrm{g}}_{u}(\boldsymbol{k}^{\mathrm{g}}_{u} \in N(\boldsymbol{k}^{\mathrm{g}}_{j'}))$ also have a high similarity under a highly similar graph structure among genuine pairs in Fig. 1 and also in **??**. Naturally, from Eq. 12, the node feature ${}^{(l-1)}\boldsymbol{h}^{\mathrm{p}}_{i'}$ and ${}^{(l-1)}\boldsymbol{h}^{\mathrm{g}}_{j'}$ will still be similar (cross-node similarity score) after one round node feature updating by Self-Graph of Eq. (12) to ${}^{(l)}\boldsymbol{h}^{\mathrm{p}}_{i'}$ and ${}^{(l)}\boldsymbol{h}^{\mathrm{g}}_{j'}$ among genuine pairs.

Graph Similarity: To get the graph-level information on the tracked graph G^{pg}, summation, weighted mean, and max can be used to combine all graph node features. Genuine image pairs can have more significant arccos value when compared to the imposter pairs on the Eq. (13). Furthermore, from the [53], the summation performance is the best compared to the weighted mean and max operations.

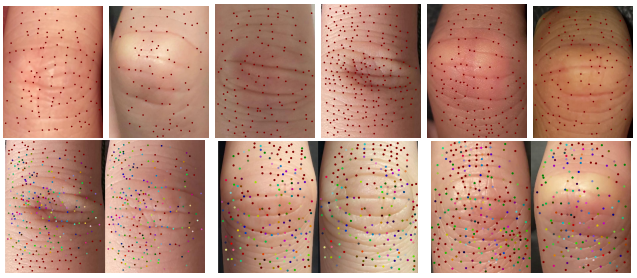


Figure A. Visualization of ground truth keypoints (first row) and the respective correspondences (matching color pairs in second row).

C. Network Training and Protocols

This section details the training protocols for optimizing the proposed matching framework with different modules. The summarized number of images of different finger knuckle database used to fine-tuning, training, and evaluation are listed in Tab. A.

Finger Knuckle Detection: We should first segment the region of interest (ROI) of the finger knuckle, the proximal interphalangeal (PIP) joint, on the dorsal finger image. Regarding our collected bending finger video dataset, multiple finger knuckles are present in one video sequence and interfere with each other when we segment the finger knuckle of one finger, such as the middle finger. Therefore, we use the trained YOLOv5 [6] and Deep SORT to track and segment the finger knuckle to keep accuracy and continuity on our captured finger knuckle videos dataset. In the rest of the public finger knuckle database, [47] and [11], we use the trained YOLOv5 [6] model to segment the finger knuckle images because they cannot offer continuous image sequence for Kalman filter of Deep SORT.

C.1. Training KnuckleCreasePoint Model

Choice of Knuckle Crease Point Detection Model: A careful analysis of finger-knuckle patterns acquired under deformations reveals that the minor crease edge patterns, and even some parts of such crease edges and intersection points, can disappear under extreme finger mobility (e.g., holding a mouse or a coffee cup). The finger knuckle is es-

Table A. Dataset organization for training and evaluation.

Dataset	KnuckleCreasePoint		KnucklePointPair		Indentification Performance		
	Fine-tuning	Training	Fine-tuning	Training	Training	Within-Database	Cross-Database
[47]	0	0	0	0	0	0	3,560
[11]	0	0	0	0	0	0	1,368
Ours	414 [1]	25,414	274 [1]	25,274	25,000	3,510	0

[1] Ground truth of keypoint and correspondence are labeled manually

sentially a 3D surface, and therefore, ambient illumination changes can significantly influence the visibility of such key feature points. Under such a complex environment, conventional methods, such as SIFT [18] or SURF [19], cannot robustly detect such keypoints. The position of keypoint-based templates must have high repeatability to ensure the effectiveness of the recovered local feature, and such descriptors should also be robust with illumination or affine invariance. Therefore, a deep neural network-based method with sufficient training data can meet such expectations and is preferred. The most significant advantage of the SuperPoint [25] is that it is a self-supervised learning method with pre-training on the synthetic shapes and then uses the homographic adaption to generate pseudo-ground truth interest point labels for unlabeled images.

Training Protocol: For training the KnuckleCreasePoint model, we manually labeled 414 finger knuckle images with keypoint ground truth in Fig. A to fine-tune the KnuckleCreasePoint model. Following, we use the fine-tuned KnuckleCreasePoint and homographic adaption detection strategy to generate the ground truth of the chosen 25,000 finger knuckle images (explained in Sec. 4.1) from the left hand of the captured video dataset to fine-tune the KnuckleCreasePoint again to get a more robust keypoint detection. Then, the KnuckleCreasePoint model, which has been trained twice, will be used to generate the new ground truth of these 25,000 images. Finally, we use the updated ground truth of 25,414 images to fine-tune the KnuckleCreasePoint model for detecting knuckle crease keypoint templates (location and feature).

C.2. Train KnucklePointPair Model

The knuckle crease keypoint feature correspondences are identified using the respective cost matrix between two sets of nodes. We use the Sinkhorn algorithm [44] instead of the Hungarian algorithm, as the time complexity of Hungarian is $O(n^3)$, and the time complexity of Sinkhorn is $O(n^2/\epsilon^2)$. (n is the dimension of the cost matrix, and the ϵ is the desired precision.) In addition, the Sinkhorn is optimized to be derivative and can be efficiently deployed on GPU. The Sinkhorn optimizer is expected to be more stable under log-domain [44] for computations.

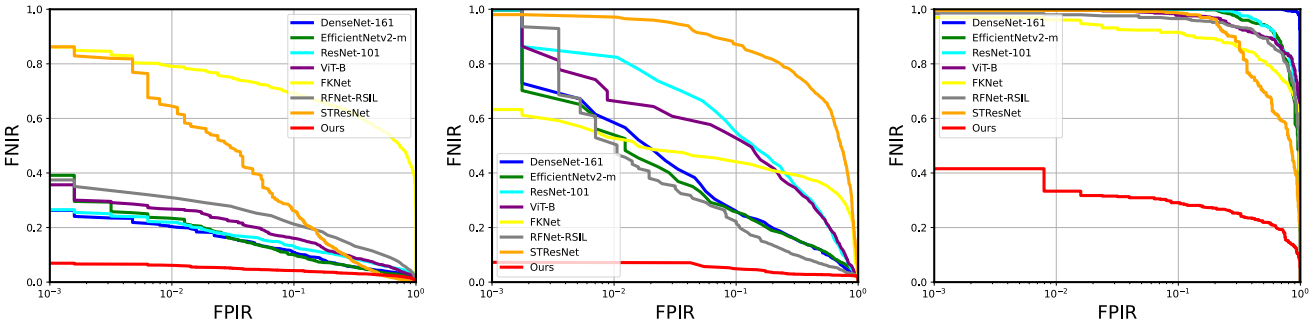
To fine-tune the KnucklePointPair model on our captured dataset, we have manually labeled 137 image pairs (274 images) for the ground truths and the correspondences as shown in Fig. A. Using the trained KnuckleCreasePoint to extract the descriptor F by given K (while K is the set of location of labeled correspondences), we fine-tune the [43] with the first round on the 137 image pairs. Then, we use the fine-tuned model and homographic adaption strategy to fine-tune the model again on the 25,274 finger knuckle images, and the resulting fine-tuned model is referred to as KnucklePointPair.

C.3. Additional Details on Baseline Models

The baseline models chosen for the comparative performance evaluation were fine-tuned or trained per the training protocol outlined in the respective references. These pre-trained ResNet-101, DenseNet-161, EfficientNetv2-m, and ViT-B models on ImageNet, were fine-tuned by perspective affine, horizontal flip, crop, and color-shifting data augmentation on the selected 25,000 images as explained in Sec. 4.1. As for the finger knuckle identification methods baselines, the FKNet, RFNet-RSIL, and STResNet models are trained using the protocols stated in respective references. Our CGN model is trained on the images by randomly selecting image pairs without any data augmentation for classification. The number of imposter pairs 624,375,000 ($1000\times999\times25\times25$) is the 1041 times the number of genuine pairs 600,000 ($1000\times25\times24$) from the 25,000 images. Therefore, focal loss is employed to address the imbalance samples between the genuine and imposter pairs, and hard samples (the farther distance of genuine pairs and closer distance of imposter pairs).

D. Additional Details on the Performance

Dataset Selection: For the cross-database performance evaluation, the dataset [47] was chosen resulting from largest 712 subjects, and the dataset [11] is the most challenging knuckle image dataset available to date in the public domain. However, the database [41] is a subset of the largest subject dataset [47], and the database [13] is largely acquired to reveal 3D finger knuckle patterns. The dataset provided by [12] is a two-session dataset (the interval is about dozens of seconds), with the least intra-class variations, as the fingers presented during the imaging were fixed (straight), and the imaging device was also in a fixed position. As evaluated in [6], the matching performance by RFNet-RSIL (employed as one of the baseline model) is already superior with higher 95% of GAR at FAR of



(a) (b) (c)

Figure B. Comparative DET plots using (a) our dataset, (b) the hand dorsal dataset [47], and (c) the most challenging dataset in[11].

0.001%. Therefore, these three datasets are excluded from cross-database evaluation. The dataset download links are provided on these respective references.

We choose ROC to evaluate comparative performance as it's the key performance metric widely used for the performance evaluation and is also required as per the ISO/IEC 19795-1:2021 for the biometrics systems. However, we also provide comparative performance evaluation, both using within-database and cross-database performance evaluation, using the false positive identification (FRIR) versus false negative identification rate (FNIR) in Fig. B. These plots estimate the identification performance with open-set access scenarios, and achieved by splitting the respective dataset with 80% identities as enrolled and the rest 20% as non-enrolled identities. These results are shown in Fig. B and are consistent with the significant performance enhancement over the existing methods observed in the results presented in Sec. 4 of this paper. It can be observed that the FNIR of our method is much lower than that of the baseline methods at the FPIR of 10^{-3}, and this difference is quite significant in the cross-database performance evaluation.

D.1. Multipose Finger Knuckle Video Dataset

Database Details: This dataset is acquired from volunteers who slowly bent their fingers from $0°$ to about $90°$ and then back from $90°$ to $0°$, repeating this motion 2 to 3 times. About 10 seconds of 4K videos, by either iPhone or Samsung smartphone, were acquired from 351 different subjects. Most volunteers are from Asia, while few are from Europe and Africa. More details on this dataset acquisition, composition, and download links are provided via [52]. We use finger knuckles of the left hand as the training set and the middle finger knuckle of the right hand as the testing set. However, the difference between contiguous frames is too small, resulting in noticeable similarity scores between contiguous frames. Therefore, our experiments do not use all frames for training or evaluation. We select an average of 25 frames per video (about two per second) for training and 10 frames per video (about one frame per second) for the performance evaluation.

The baseline CNN models and the ViT-B offer relatively similar performance in this dataset, while the EfficientNetv2-m is the best and the ViT-B is the worst. From the corresponding experimental results in Fig. 6a and summarized Tab. 2 on Sec. 4.1, the ViT-B [51] stated that the ViT can only outperform the CNN models when fed with enough data. As for the rest of the lightweight (shallow) matching models for finger knuckle, their performance is the lowest because they are designed for training and testing on small finger knuckle datasets, while these heavyweight SOTA CNN models will overfit [7]. These finger knuckle models can outperform the most lightweight version of these SOTA CNN models. However, we have more training sets and testing sets that can make full use of the ability of these heavyweight SOTA CNN models to avoid overfitting.

D.2. PolyU Hand Dorsal Image Dataset

These results also indicate high generalization ability based on the interpretable method proposed in this paper. The performance of our model doesn't drop with too much on GAR values, and the performance gap between our model and other models is increased in Fig. 6b and summarized Tab. 3 on Sec. 4.2. It shows that our model can perform best on our captured finger knuckle dataset (Sec. 4.1) and tackle the flat finger knuckle images with the highest generalization ability. As the reset model, their performance has dropped significantly compared to our method on the EER and GAR values. The baseline CNN-based models use the learned pattern kernels to recover activation feature maps. However, this dorsal image dataset is flatter than our captured finger knuckle and has different texture information; these CNN-based models' performance has degraded as expected. Lastly, the ViT-B, with an attentional mechanism, shows degraded performance, which can be attributed to the

changes in the correlation between image patches, resulting in different deep features. The texture changes at different keypoint locations can also be observed in Fig. F.

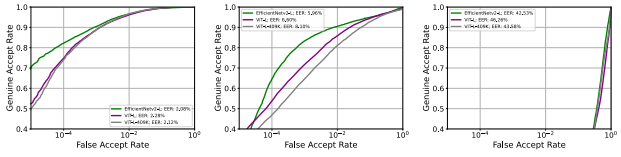
D.3. PolyU Contactless Finer Knuckle v3.0 Dataset

Table B. Comparative performance summary for the dataset in [11] with protocols as in [13].

Model	GAR (FAR=10^{-5})	GAR (FAR=10^{-4})	GAR (FAR=10^{-2})
DenseNet-161 [49]	0.00%	0.00%	0.00%
EfficientNetv2-m [50]	0.00%	0.00%	0.00%
ResNet-101 [48]	0.00%	0.00%	0.00%
ViT-B [51]	0.00%	0.00%	1.28%
FKNet [7]	3.68%	7.52%	27.04%
RFNet-RSIL [6]	4.18%	5.79%	19.55%
STResNet [15]	0.32%	0.64%	17.31%
Ours	**77.51%**	**90.23%**	**96.31%**

Our outperformed performance can be attributed to the robustness of (interpretable) knuckle crease points compared to local or global textures, which can still be detected even when the view angle, pose, or illumination changes. Furthermore, our proposed graph neural network can capture the graph structure similarity and the descriptor similarity of such keypoints at the same time. The global knuckle pattern appearance can significantly change with pose changes due to significant agility in the proximal interphalangeal joint. These changes are dramatic in [11] two-session dataset and often result in fatal feature changes. This can explain the relatively low performance achieved on this dataset in Fig. 6c and Fig. 6d and summarized Tab. 4 and Tab. B, respectively. The CNN-based, ViT-B, and SOTA finger knuckle models that use global features cannot deal with such deformations in finger knuckle features, resulting in poor performance. Comparative results in Fig. 6c and Fig. 6d indicate that our keypoint-based method with a graph neural network offers significantly enhanced generalizability.

D.4. EfficientNetv2-L and ViT-L



(a) (b) (c)

Figure C. Comparative ROC result from the EfficientNetv2-L and ViT-L under the all-to-all protocol on (a) our dataset, (b) hand dorsal [47] dataset, and (c) the most challenging finger knuckle [11] dataset.

From the above within-database and cross-database performance, the EfficientNetv2-m [50] can outperform the rest of the baselines compared in this paper. The ViT [51] has been successfully used in many applications as the backbone of the current large vision models, largely due to its high generalization ability with ultra-large-scale datasets. Therefore, a more extensive version of these two models (EfficientNetv2-L and ViT-L) was also employed for the performance evaluation and used the same training steps discussed earlier Supplementary C.3. Instead of only using 25,000 images (explained in Sec. 4.1), the ViT-L (referred to as ViT-L-409K) was also trained using *all* the knuckle image samples from the left hands, i.e., with 409,267 images, to ascertain the performance from the ViT model using a rich training set. Such experimental results to show these models performance are presented in Fig. C, and it can also ascertain our proposed matching method efficiency when compared to Fig. 6.

D.5. Estimation of Model Complexity

From the within-database and cross-database, our CGN model can significantly outperform these employed baseline models, especially on the most challenging dataset [11]. In Tab. C, we present relative model complexities, which have been estimated with different indices. The complexity of our model is not high, and our complete system is live, which can perform completely contactless finger knuckle identification online.

Table C. Comparative complexity of different models.

Model [1]	Params [2] (M)	Model Size (MB)	FLOPs [2] (G)	Inference Time [3] (ms)
DenseNet-161	28.7	116.2	15.7	11.4
EfficientNetv2-m	54.2	218.3	49.9	15.0
ResNet-101	44.7	179.0	15.7	15.4
ViT-B	87.05	348.8	33.7	8.6
FKNet [4]	53.91	210.1	2.7	24.8
RFNet-RSIL	1.35	5.4	2.8	4.3
STResNet	19.47	77.9	12.5	13.8
Keypoint+GMN	14.5	59.4	22.4	18.7
Keypoint+SimGNN	13.3	53.8	14.4	17.2
Keypoint+MGMN	13.5	54.5	14.5	17.0
Keypoint+ERIC	13.4	53.9	14.4	16.5
Ours	14.8	61.9	14.7	16.8

[1] These graph similarity models (including ours) rely on the detected keypoint template. Therefore, the model complexity from keypoint detection and correspondence estimation module, represented by Keypoint, should be added.
[2] The thop library estimated the Params and FLOPs of all models except FKNet, while NetScope estimates the FKNet.
[3] The inference time (feature extracting and matching) is the average time for one image pair and is estimated from 3270 image pairs in our captured dataset [52] using Ubuntu 20.04 LTS OS machine with GeForce RTX 4090 and i5-12400F CPU.
[4] The FKNet was implemented under Caffe architecture, and its inference time is estimated using Matlab as in [7]. The rest of the models are implemented using PyTorch architecture (PyTorch 2.0.0).

D.6. Additional Details of Ablation Study

Graph Similarity Model: To ensure a fair comparison, we change the input node feature of these models (graph similarity models) to fit descriptors of detected keypoints from the KnuckleCreasePoint. We then keep the same default

Table D. Comparative performance for our proposed different modules using challenging all-to-all protocol on hand dorsal database [47].

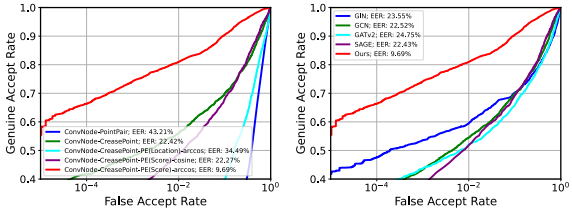
Model				GAR	GAR	GAR
Graph Convolution	Feature	Positional Embedding	Cross Similarity	(FAR=10^{-5})	(FAR=10^{-4})	(FAR=10^{-2})
ConvNode	KnucklePointPair	-	arccos	27.79%	89.78%	96.98%
ConvNode	KnuckleCreasePoint	-	arccos	60.14%	94.07%	97.06%
ConvNode	KnuckleCreasePoint	Sorted by Score	cosine	64.61%	94.54%	96.94%
ConvNode	KnuckleCreasePoint	Sorted by Locate	arccos	76.96%	93.59%	97.05%
ConvNode	KnuckleCreasePoint	Sorted by Score	arccos	**80.19%**	92.88%	97.09%
GIN [53]	KnuckleCreasePoint	Sorted by Score	arccos	61.77%	92.85%	97.12%
GCN [56]	KnuckleCreasePoint	Sorted by Score	arccos	62.89%	91.23%	97.32%
GATv2 [55]	KnuckleCreasePoint	Sorted by Score	arccos	71.74%	81.78%	97.27%
SAGE [54]	KnuckleCreasePoint	Sorted by Score	arccos	48.12%	89.95%	97.09%

Table E. Comparative performance between the SOTA graph matching models and ours by using challenging all-to-all protocol on our captured, the hand dorsal [47], and the most challenging [11] dataset.

Model	GAR (FAR=10^{-5})	GAR (FAR=10^{-4})	GAR (FAR=10^{-2})
GMN [26]	11.28%	26.69%	76.72%
SimGNN [28]	28.26%	58.02%	93.94%
MGMN [29]	12.18%	42.41%	93.97%
ERIC [31]	13.46%	37.74%	90.52%
Ours	**89.99%**	**93.19%**	**97.32%**
GMN [26]	0.00%	13.45%	77.49%
SimGNN [28]	4.28%	21.38%	83.51%
MGMN [29]	4.38%	9.22%	69.90%
ERIC [31]	6.51%	20.94%	73.60%
Ours	**80.19%**	**92.88%**	**97.09%**
GMN [26]	0.00%	4.30%	41.20%
SimGNN [28]	16.95%	34.10%	69.51%
MGMN [29]	0.34%	2.40%	48.65%
ERIC [31]	1.04%	3.91%	49.56%
Ours	**55.41%**	**66.35%**	**80.95%**

configurations of these models for the performance evaluation on the corresponding graphs composed of mutually matched correspondences. From the experimental result in Fig. 8 and summarized Tab. E, the main reason why our model can outperform these models is that these models were introduced for *graph edit distance datasets*, *chemical compound graphs*, *program dependence graphs*, and *ego-network of movie graphs*. These datasets have the one-hot labeled node feature and have a very high similarity graph structure among the same graph class. However, our correspondence graphs are more complex because node features differ, unlike using one-hot nodes with high cosine similarity. The graph structure is also different even in the same class, resulting from different correspondences on different image pairs. These graph similarity models attempt to learn the graph structure similarity between graph pairs. Additionally, the CGN can understand the node-to-node similarity along the graph structure. It can also be observed from Fig. 8c to the Fig. 6c that graph-matching methods offer much better performance than the other baseline methods based on CNN or ViT models.

Module Efficiency: From the Sec. 3.3, our proposed model uses the feature from the KnuckleCreasePoint, adds the positional embedding, and incorporates the arccos similarity to calculate the node-to-node similarity among two graphs that have better performance on the cross-database in Fig. 8d when compared to without these modules. Another set of ablation studies involved replacing our ConvNode with other graph convolutions, such as GIN [53], SAGE [54], GATv2 [55], and GCN [56], while using the best configurations and the same model architecture. Furthermore, we also evaluate the cross-database performance, *with* and *without*, on the most challenging two-session dataset [11] in Fig. D.



(a) (b)

Figure D. Comparative ROC plots for proving our performance with ablation study on the deformable dataset [11] (a) ablation study of our proposed methods; (b) compare to other graph convolution modules.

Such results in Fig. 8e, Fig. D, and summarized Tab. D, indicate that our convolution on node dimension can significantly enhance the performance with higher TAR (FAR@10^{-5}) values. It can also be observed from these results that all other considered graph convolutions offer quite similar recognition performance. The feature vectors resulting from the KnucklePointPair through self-attention and cross-attention are expected to have high similarity, regardless of genuine and imposter pairs. Compared to the KnuckleCreasePoint features from Fig. 8d and Fig. D, it can explain relatively poor performance. In addition, our model adds the positional embedding, based on the ConvNode, which is expected to improve the matching accuracy compared to without positional embedding. Superior performance using arccos similarity instead of cosine similarity can be attributed to the enhancement in the values from the relative differences. Lastly, it can also be observed that positional embedding based on the mutually matched scores

offers better results than positional embedding based on the keypoint locations. When sorted by the match scores, the top scores or positions are expected to be the keypoint pairs and relatively generate higher similarity.

D.7. Analysis on Keypoints and Descriptors

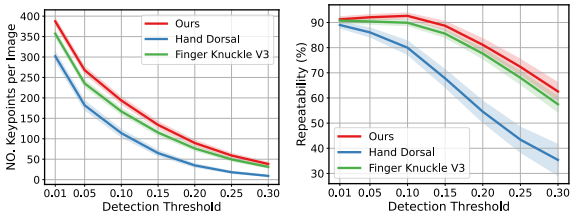
Finger knuckle crease keypoints offer rich and interpretable source of information for the matching. This section presents an experimental analysis of extracting such keypoints and respective features using the approach adopted in this paper. We present such analysis like those for SIFT [18] (i.e. each image will be randomly rotated from $-45°$ to $45°$, be randomly scaled from 0.85 to 1.15, and each pixel value will be randomly added 1% percent noise) to evaluate the repeatability in the detection of knuckle crease keypoints and the reliability of respective feature descriptors detected by KnuckleCreasePoint using 152×200 size ROI images that are used for the CGN model.

D.7.1. Knuckle Crease Keypoints Repeatability

We count the number of detected keypoints per image and the repeatability of the location (within $\pm$ 10 pixels) in the detection of keypoints by varing the detection threshold, as shown from plots in Fig. E with a standard variance of 0.2. With a higher detection threshold, a smaller number of keypoints will be detected by KnuckleCreasePoint model. A lower detection threshold is expected to result in higher number of detected keypoints with much of the noise. Therefore, to maintain high repeatability and to alleviate noise or spurious keypoints, we chose the detection threshold of 0.15 for computing similarity based on the number of mutually matched correspondences in Fig. I. The number of detected keypoints and their repeatability is influenced by the image quality. In this context, the quality of segmented finger knuckles of the hand dorsal dataset [47] is poor as compared to the other datasets used in our study. This is also the reason that the average number of keypoints detected in images from this dataset is lowest. Based on the same detection threshold, we also generate the statistics for the distribution of the spatial location of keypoints in the three considered public datasets, as shown in Fig. F. Because our dataset [52] and the dataset in [11] contains challenging or deformed finger knuckle images, distribution of the detected keypoints is much higher towards the lower regions of the finger knuckles. In contrast, the images in [47] have a fixed or straight pose, and therefore, the distribution of keypoints in the ROI regions is relatively even. The heatmaps in Fig. F also indicate that the probability that a keypoint will be detected at the boundary regions of ROI images is low.

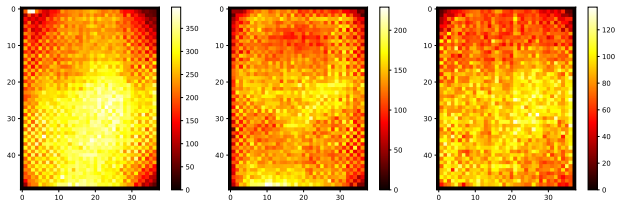
D.7.2. Reliability of Keypoint Descriptors

Besides the statistics on the detection of keypoints, we also analyze the reliability in estimating detected feature descriptors for the respective keypoints. Since the KnuckleCreasePoint model uses the dot product similarity score as the match score, our proposed CGN model uses the CrossNode to compute the node-to-node arccos similarity. The plots in Fig. Gb indicate that we can detect the keypoints and descriptors on all finger knuckle images using a predetermined detection threshold and generate all the keypoint descriptors as a templates database. Then, we match a descriptor of a detected keypoint of an image to the database with the nearest dot product value to ascertain whether it can be matched with itself. We obseve that the descriptor is reliable enough so that almost all of them can be matched with itself on the descriptor database under different detection thresholds. Furthermore, we also determine the descriptor reliability when we set the pixel values outside a specific block size (the value of Fig. Gc is the length of the square block) to zero to simulate the extent of the influence from neighboring pixels. As shown in Fig. Ga, for a keypoint labeled with red color, only the original pixel values within the block size are considered with blue color. We use the KnuckleCreasePoint model to extract the keypoint feature vector (descriptor) again from such masked knuckle ROI image and match the resulting descriptor with the descriptors in the database with the same method as for Fig. Gb to compute the matching accuracy of descriptors. From the plots in Fig. Gc, the matching accuracy is significantly degraded or very low when the block size is smaller than 48 pixels. Such reduction in reliability can be considered as a shortcoming of the CNN models because a pixel of a fea-



(a) (b)

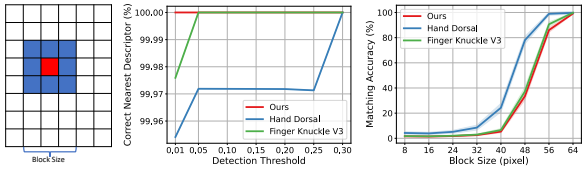
Figure E. Under different detection thresholds, (a) the number of keypoints per image, (b) the repeatability of keypoints.



(a) (b) (c)

Figure F. Visualization of the location distribution of keypoints on different finger knuckle datasets under detection threshold 0.15. (a) our finger knuckle video dataset, (b) hand dorsal image dataset [47], (c) finger knuckle v3 dataset [11].

ture map will be affected by the neighborhood pixel values within the current field of view (the estimated filed of view for the detection head of KnuckleCreasePoint is 64), which in turn is influenced by the choice of kernel, stride, and pooling size. If the block size exceeds 56, the descriptor match accuracy can be considered as reasonable or acceptable.

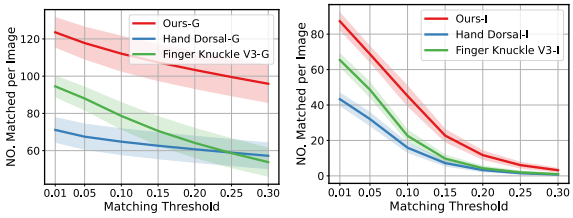


(a) (b) (c)

Figure G. Analyzing effectiveness of knuckle crease keypoint feature descriptor: (a) visualization of a pixel-centered block within the masked region, (b) variation in the average match accuracy of the feature descriptor, with the nearest ones, for different detection thresholds, (c) variation of the average match accuracy of the feature descriptor, with the nearest ones, for different sizes of blocks.

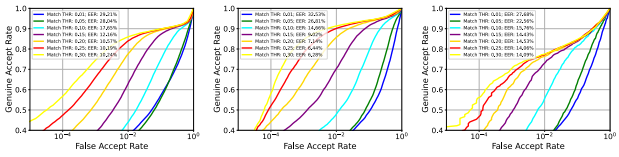
D.7.3. Number of Mutually Matched Keypoints

Finally, based on the detection threshold of 0.15 for KnuckleCreasePoint model, we can determine how many correspondences are matched under a given match threshold. Such a plot is shown in Fig. H, with a standard variance of 0.2, is shown for the different datasets. The genuine image pairs have more mutually matched correspondences as compared to those for the imposter pairs and this is expected since the similarity of respective descriptors in genuine pairs will be larger than those in imposter pairs, as also discussed in Sec. 3.3. The number of mutually matched correspondences decrease with the increase in the matching threshold. As shown in Fig. Hb, the average number of matched correspondences tends to zero as the match threshold increases. After locating correspondences between two images, we can use the number of mutually matched keypoints as the match score between two finger knuckle images, which can be used to ascertain the expected performance of the correspondences with a simple matching method. Such an evaluation is shown in Fig. I, and uses the all-to-all protocol for such matching performance. With the match threshold gets stricter, the matching performance is enhanced on the three datasets as it is expected to alleviate several falsely matched correspondences. If we compare this match performance with those in Fig. 6, it can be observed that the performance achieved is even better than some baselilne SOTA models, largely due to the robustness of the powerful keypoint descriptors as compared to texture-based analysis. Especially for the deformable or challenging finger knuckle dataset [11] performance in Fig. Ic, it's easy to observe that the keypoint-based matching performance is better than using the texture information on the respective (deformable) images as it outperforms the other SOTA models.



(a) (b)

Figure H. Variation in the *number* of mutually matched knuckle crease keypoints with different match thresholds (under a fixed keypoint detection threshold of 0.15), (a) for genuine match pairs and (b) imposter match pairs.



(a) (b) (c)

Figure I. Comparative ROC plot using the *number* of mutually matched keypoint pairs under the all-to-all protocol: (a) using the index finger knuckle images from the right hands in our finger knuckle video dataset [52], (b) using middle finger knuckle images in the hand dorsal dataset [47], (c) using two-session samples on the deformable finger knuckle dataset [11].

E. Additional Details on Uniqueness Analysis

Any study on the uniqueness of finger knuckle patterns should utilize images acquired from the fixed pose that can reveal rich features, just as for the other biometric modalities like fingerprints [39]. Therefore, we also selected knuckle images with fixed or upright poses for the uniqueness evaluation. Unlike other popular biometrics, this is the first attempt to estimate the uniqueness of 2D finger knuckle patterns, and therefore, a comparative analysis of such uniqueness from the different image resolutions (appearing in the public datasets) is also presented using the FRC index. In addition to our dataset [52], the hand dorsal dataset in [47] is also used to estimate the FRC of relatively low-resolution images. From the finger knuckle detection results, the average image resolution of our dataset is 598.65×800.87 (width $\times$ height and about 760 dpi), and such values using the hand dorsal image dataset [47] is 158.34×185.10 (about 200 dpi). We resized the detected finger knuckle images in our dataset to 296×400 and resized such images from hand dorsal dataset to 152×184 for estimating the FRC. Such an image size was empirically chosen as a trade-off the accuracy between accuracy

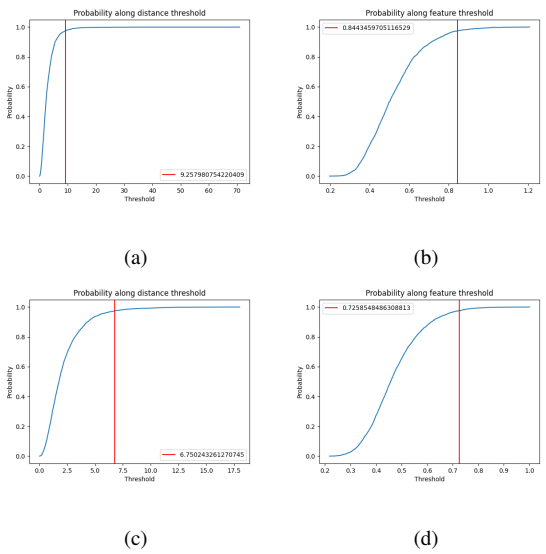


Figure J. Plots in (a) and (b) illustrate the spatial position and feature threshold estimated for our finger knuckle video dataset using 296×400 (width $\times$ height) size ROI images, while the (c) and (d) are the respective estimation of thresholds from the hand dorsal image dataset using 152×184 size ROI images.

and size of finger knuckle images. It should be noted that the knuckle crease keypoint location detection branch of KnuckleCreasePoint model is trained on the feature maps ($1/8$ of input image size). If we set such image size from our dataset to 592×800 for training the KnuckleCreasePoint model, the detected feature map, with 74×100 size, has many negative samples resulting in inferior detection accuracy. The KnuckleCreasePoint model can be modified for such high resolution. However, to ensure fairness in the comparisons of the uniqueness, the same model architecture and loss function as employed in Sec. 3.1 should be used. Lastly, the KnuckleCreasePoint model needs to be trained again at these two image resolutions by following the same training protocol in Supplementary C.2. It is reasonable to assume that the distribution of different client's keypoint templates (position and feature) differs. Therefore, the n-components of BIC were computed for each subject in our database to estimate the parameters for the distribution in Eq. (18).

E.1. Estimation on Tolerance for Keypoint Matches

The knuckle crease keypoint features in the corresponding locations are considered a match if the spatial locations and feature distance are close or within a predetermined threshold (referred to as the tolerance). In this study, we utilize a database comprised of ground truths of knuckle keypoint matching pairs to establish the protocol for an authentic *match*, thereby computing $p(T^{\mathrm{p}}, T^{\mathrm{g}})$. Given the genuine matches among ground truths, the rigid transformation matrices between such pairs can be determined and used for the template alignment. Similar to the fingerprints in [32], after such alignment, the spatial distances $\delta(\boldsymbol{k}^{\mathrm{p}}, \boldsymbol{k}^{\mathrm{g}})$ and feature value differences $\delta(\boldsymbol{f}^{\mathrm{p}}, \boldsymbol{f}^{\mathrm{g}})$ between each pair of matched keypoint $\langle \boldsymbol{k}^{\mathrm{p}}, \boldsymbol{f}^{\mathrm{p}} \rangle$, $\langle \boldsymbol{k}^{\mathrm{g}}, \boldsymbol{f}^{\mathrm{g}} \rangle$ of knuckle are computed. From the estimation illustrated in Fig. J, the $\Delta_{\boldsymbol{k}}$ and $\Delta_{\boldsymbol{f}}$ tolerance for our dataset is respectively 9.258 and 0.844, while respective values of the hand dorsal dataset [47] are estimated as 6.750 and 0.726.

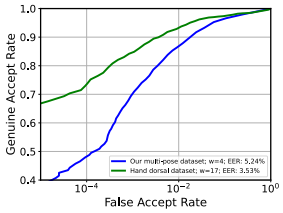


Figure K. Estimating the minimum number of matched keypoints w for the uniqueness analysis in this paper. The above plots is for our [52] and hand dorsal dataset [47] images and were generated using the estimated spatial distance and feature distance thresholds.

E.2. Number of Matched Keypoints

This section outlines the process of estimating the minimum value of w, i.e. an unknown pair of knuckle templates is considered as *matched* if the *number* of keypoint correspondences between the gallery and probe templates $\geq w$. Conversely, the input pair is considered as a *non-match* if the number of keypoint correspondences does not exceed w. We first compute the number of matched correspondences using the Euclidean distance ($\delta(\boldsymbol{k}^{\mathrm{p}}, \boldsymbol{k}^{\mathrm{g}}) \leq \Delta_{\boldsymbol{k}}$ and $\delta(\boldsymbol{f}^{\mathrm{p}}, \boldsymbol{f}^{\mathrm{g}}) \leq \Delta_{\boldsymbol{f}}$) generated from all-to-all match protocols. The number of such matched keypoints is considered as the match score to differentiate between genuine and imposter matches resulting from the input pair of templates. The optimal value of w is determined from a performance metric, equal error rate in our experiments, to differentiate between the genuine and imposter finger knuckle pairs.

From the labeled ground truth correspondences in the subset of images from the hand dorsal image dataset, the average number of correspondences is 59.03 among genuine image pairs. Our trained KnuckleCreasePoint model on 152×184 size ROI images, with 0.19 detection threshold, generates an average of 59.15 keypoint correspondences among the genuine pairs while the average number of keypoints in such images is 69.412. This is the key reason we set $m = n = 69$ (Tab. 5) for a fair uniqueness analysis in Tab. 5 and Tab. G. Following the same rule of finding the optimal value of w, we determine the optimal value of w for this study, which is found to be 17 for hand dorsal [47] and 4 for our dataset in Fig. K.

Table F. Uniqueness analysis using the keypoints generated from knuckle crease bifurcations and endings.

Dataset	$\Delta_{\boldsymbol{k}}$	$\Delta_{\boldsymbol{f}}$	(m, n, w, α)		$\overline{p(T^{\mathrm{p}}, T^{\mathrm{g}})}$		$\overline{\mathrm{FRC}}_{\alpha}$	
			Bifurcation	Ending	Bifurcation	Ending	Bifurcation	Ending
Hand Dorsal [47]	6.750	0.726	(55,55,11,0.05)	(14,14,2,0.05)	3.929×10^{-4}	4.300×10^{-4}	1.233×10^{-3}	5.843×10^{-4}
	6.750	0.726	(55,55,13,0.05)	(14,14,4,0.05)	3.929×10^{-4}	4.300×10^{-4}	2.807×10^{-4}	6.523×10^{-6}
	6.750	0.726	(55,55,15,0.05)	(14,14,6,0.05)	3.929×10^{-4}	4.300×10^{-4}	5.292×10^{-5}	4.640×10^{-8}
	5.250	0.726	(55,55,11,0.05)	(14,14,2,0.05)	2.541×10^{-4}	2.661×10^{-4}	7.947×10^{-5}	1.454×10^{-4}
	8.250	0.726	(55,55,11,0.05)	(14,14,2,0.05)	5.445×10^{-4}	6.267×10^{-4}	6.028×10^{-3}	1.642×10^{-3}
	6.750	0.676	(55,55,11,0.05)	(14,14,2,0.05)	7.376×10^{-5}	1.357×10^{-4}	5.252×10^{-10}	1.903×10^{-5}
	6.750	0.776	(55,55,11,0.05)	(14,14,2,0.05)	1.508×10^{-3}	1.106×10^{-3}	1.021×10^{-1}	6.641×10^{-3}
Ours	9.258	0.844	(61,61,3,0.05)	(8,8,1,0.05)	5.646×10^{-5}	2.068×10^{-4}	3.220×10^{-4}	9.609×10^{-5}
	9.258	0.844	(61,61,5,0.05)	(8,8,3,0.05)	5.646×10^{-5}	2.068×10^{-4}	5.419×10^{-6}	2.174×10^{-8}
	9.258	0.844	(61,61,7,0.05)	(8,8,5,0.05)	5.646×10^{-5}	2.068×10^{-4}	6.383×10^{-8}	3.146×10^{-12}
	7.758	0.844	(61,61,3,0.05)	(8,8,1,0.05)	4.006×10^{-5}	1.467×10^{-4}	9.134×10^{-5}	4.670×10^{-5}
	10.758	0.844	(61,61,3,0.05)	(8,8,1,0.05)	7.548×10^{-5}	2.781×10^{-4}	9.101×10^{-4}	1.779×10^{-4}
	9.258	0.794	(61,61,3,0.05)	(8,8,1,0.05)	2.103×10^{-5}	9.755×10^{-5}	9.293×10^{-6}	2.078×10^{-5}
	9.258	0.894	(61,61,3,0.05)	(8,8,1,0.05)	1.303×10^{-4}	3.858×10^{-4}	4.425×10^{-3}	3.373×10^{-4}

Table G. Uniqueness analysis using our database [52].

$\Delta_{\boldsymbol{k}}$	$\Delta_{\boldsymbol{f}}$	(m, n, w, α)	$\overline{p(T^{\mathrm{p}}, T^{\mathrm{g}})}$	$\overline{\lambda}_{\alpha}$	$\overline{\mathrm{FRC}}_{\alpha}$
9.258	0.844	(69,69,2,0.05)	5.387×10^{-5}	0.316	4.161×10^{-3}
9.258	0.844	(69,69,4,0.05)	5.387×10^{-5}	0.478	1.400×10^{-4}
9.258	0.844	(69,69,6,0.05)	5.387×10^{-5}	0.601	3.328×10^{-6}
9.258	0.844	(69,69,8,0.05)	5.387×10^{-5}	0.694	5.497×10^{-8}
9.258	0.844	(69,69,17,0.05)	5.387×10^{-5}	0.880	1.515×10^{-17}
6.258	0.844	(69,69,4,0.05)	2.459×10^{-5}	0.230	4.385×10^{-6}
7.758	0.844	(69,69,4,0.05)	3.765×10^{-5}	0.341	2.915×10^{-5}
10.758	0.844	(69,69,4,0.05)	7.306×10^{-5}	0.624	4.709×10^{-4}
12.258	0.844	(69,69,4,0.05)	9.494×10^{-5}	0.784	1.290×10^{-3}
9.258	0.744	(69,69,4,0.05)	5.932×10^{-6}	N/A	8.678×10^{-9}
9.258	0.794	(69,69,4,0.05)	1.954×10^{-5}	0.194	1.968×10^{-6}
9.258	0.894	(69,69,4,0.05)	1.247×10^{-4}	0.954	2.999×10^{-3}
9.258	0.944	(69,69,4,0.05)	2.510×10^{-4}	1.551	2.106×10^{-2}

E.3. Uniqueness with Higher Image Resolution

Contactless finger knuckle image resolution can vary even within the images in a dataset. However, the resolution of images in our dataset [52] is higher than that of those in [47] used for analysis in Tab. 5. Therefore, it can be useful to estimate the finger knuckle uniqueness on such relatively higher-resolution ROI images in our dataset. In Tab. 5 (Tab. G), we present the $\overline{\mathrm{FRC}}_{\alpha}$ score when $\Delta_{\boldsymbol{k}}$, $\Delta_{\boldsymbol{f}}$, and w are respectively set to 6.750, 0.726, and 17 (9.258, 0.844, and 4). Additionally, we adjust the value of $\Delta_{\boldsymbol{k}}$ from 3.750 to 9.750 pixels (6.258 to 12.258) in increments of 1.5, the value of $\Delta_{\boldsymbol{f}}$ from 0.626 to 0.826 (0.744 to 0.944) in increments of 0.05, and the value of w from 15 to 21 (2 to 8) in increments of 2. This is done to observe the variations in the $\overline{\mathrm{FRC}}_{\alpha}$ score about the Poisson parameter λ_{α}. Our findings suggest that as the match requirements become more stringent (increase of values in the first two columns in Tab. G), the estimated uniqueness of the finger knuckle templates increases, as evidenced by the observed decrease in the FRC_{α} scores. It can be observed from the comparisons in Tab. 5 and Tab. G, that the FRC_{α} score in Tab. G is relatively lower under the same thresholds, which is expected due to relatively higher image resolution.

E.4. Uniqueness with Bifurcations and Endings

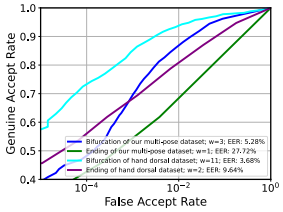


Figure L. Estimating the minimum number of matched keypoints w, separately from the knuckle crease bifurcations and endings, using the images in our [52] and hand dorsal dataset [47].

It is reasonable to believe that the keypoints detected from the knuckle crease bifurcations are more reliable than those from the knuckle crease endings. Unlike fingerprints, the thickness of knuckle creases varies significantly, and the number of keypoints from knuckle crease bifurcations is significantly much larger than those from the crease endings. Therefore, the uniqueness analysis presented in Tab. 5 has not differentiated the nature of keypoints. However, it can be interesting to estimate the uniqueness of finger knuckle patterns considering only the match among the keypoints generated from knuckle crease bifurcations or the endings. In Tab. F, we separately analyze the $p(T^{\mathrm{p}}, T^{\mathrm{g}})$ match probability of a pair of random matches between a pair of keypoints (one is on the probe template, the other is on the gallery template), specifically generated from the knuckle crease bifurcation and ending types. Following the same steps as earlier, the average number of bifurcations and endings is respectively 55 and 14 for the hand dorsal dataset [47], and the parameter w of bifurcation and ending is respectively 11 and 2 in Fig. L, respectively. For our dataset [52], the average number of bifurcations and endings is respectively 61 and 8, while the parameter w respec-

tively for the bifurcation and ending is 3 and 1 in Fig. L. It can be observed from our results in Tab. F, the probability $p(T^{\mathrm{p}}, T^{\mathrm{g}})$ of bifurcation is lower than that of ending, especially on high resolution (while high-resolution images are expected to offer more minutiae details), thereby contributing more significantly to the uniqueness of knuckle keypoint templates. While the probability $\overline{\mathrm{FRC}}_\alpha$ is calculated by the Poisson distribution with $\lambda = mnp(T^{\mathrm{p}}, T^{\mathrm{g}})$, the $\overline{\mathrm{FRC}}_\alpha$ increases for the larger values of m and n. Therefore, the $\overline{\mathrm{FRC}}_\alpha$ from the crease bifurcations is higher than those from the crease endings, while the average number of crease bifurcation is as 3.9 (7.6) times as the average number of crease ending on hand dorsal dataset [47] (our dataset [52]).

E.5. Societal and Privacy Related Impact

This work is expected to generate positive societal impact especially for the forensic community, e.g. in timely and accurately detecting real-world cases, e.g. child abuse [2–4], where finger dorsal images are the only pieces of scientific evidence available to establish identity of suspects. Development of advanced and accurate algorithms to accurately match real-world contactless finger knuckle patterns can also lead to its deployment in mobile phone security, multimodal or other applications that can generate further positive impact. We have acquired the datasets as per the IRB guidelines and ensured complete anonymity for the distribution of dataset. Therefore, our work in this paper will not have any adverse privacy related impact.