# Robust Partial Fingerprint Recognition
## **Supplementary Material**

Yufei Zhang[1,*], Rui Zhao[2], Ziyi Zhao[2], Naveen Ramakrishnan[2]
Manoj Aggarwal[2], Gerard Medioni[2], Qiang Ji[1]
[1]Rensselaer Polytechnic Institute, [2]Amazon
{zhangy76, jiq}@rpi.edu, {zhaori, zhaziyi, navramk, manojagg, medioni}@amazon.com

## A. Hyperparameter Setting and Additional Implementation Details

**Hyperparameter Setting.** For the baseline model, the initial rate is 2e-3. The training epoch is 50. When using AugMix, the weight on Jensen divergence term is 0.1. For training RPF, the probability to apply alignment-related augmentation is 0.6 and the range of translation, rotation, and scaling is [0, 0.1], [0, 45], and [0.85, 1.15], respectively. The probability to generate partial fingerprints is 0.5 and the range of OR is [0.1,0.45]. The augmentation depth is $k = 5$. The parameter of the Dirichlet distribution is $\alpha = 1$, meaning random sampling the weights without preference to a specific augmentation operation. We initialize the model weights using the baseline model and train the model using O-AugMix with 10 epochs and learning rate of 1e-5. Then, we incorporate occlusion-aware modeling. The images generate by AugMix are mix of different partial fingerprints and hence can not be used to train the segmentation branch. We hence only train the model with the augmented data without being mixed to the original images. We first train the segmentation branch and then train the whole model 30 epochs with learning rate of 1e-5 with only the parameters of the mask prediction module being updated. The models are trained using four Nvidia V100 GPUs.

**Image Pre-processing.** The relative fingerprint position and rotation can vary significantly in the employed data. We pre-process the original fingerprint data to obtain aligned finger impression data during training and testing. Specifically, for a fingerprint, we assume the fingerprint area is an ellipse. We use OpenCV's built-in function to obtain the fitted ellipse for each finger impression image, which provides ellipse center $(c_{center}, r_{center})$ and the relative rotation $\theta$ to align the images. We then rectify the principle axes to normal direction to align the raw fingerprint.

[*]Work done while as an intern at Amazon.

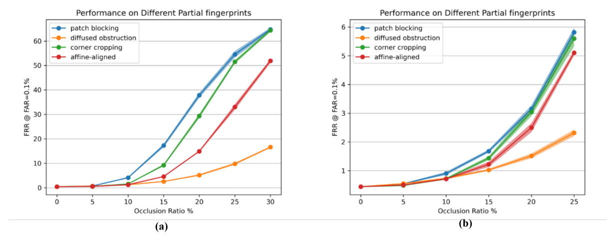


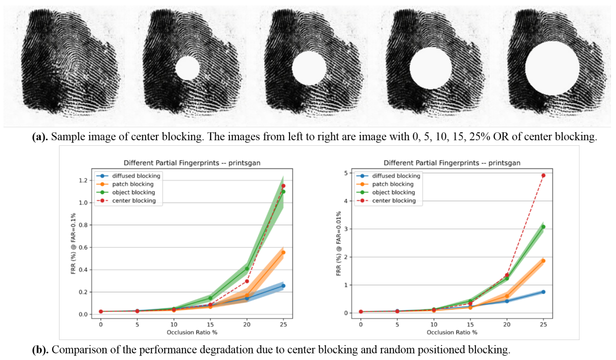

Figure 1. Evaluation on the image pairs with both partially observed template and query fingerprints. The curves describe FRR @ FAR=0.1% on different partial fingerprints at 0-25% OR. Evaluation in (a) and (b) considers images with only partially observed query fingerprints and with both partially observed template and query fingerprints, respectively.

## B. Partially Observed Template and Query Fingerprints

When both the template and query fingerprints are partially observed, matching becomes more challenging as less correspondences can be constructed. Here we show the evaluation when both of images with randomly introduced occlusion in Figure 1. We compare the performance degradation considering both the template and query fingerprints are partially observed (left) and only the query fingerprints are partially observed(right), respectively. The performance degradation becomes more severe when both the template and query fingerprints are partially observed. To handle such challenging cases, we may consider matching given multiple template images.

## C. Effect of Missing Information in Different Fingerprint Regions

We compare the performance degradation due to random blocking and due to center blocking. Specifically, we block the center position of fingerprints with a circle at different ORs (examples are shown in Figure 2(a)) and evaluate the recognition performance on these center blocked finger-




**(a).** Sample image of center blocking. The images from left to right are image with 0, 5, 10, 15, 25% OR of center blocking.

**(b).** Comparison of the performance degradation due to center blocking and random positioned blocking.

Figure 2. Comparison of the effect of missing information in random region with in fingerprint center.

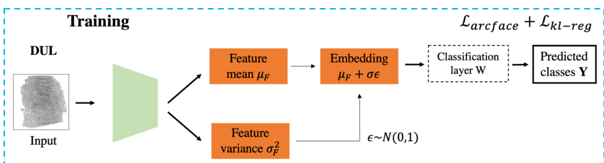



Figure 3. Overview of fingerprint recognition model with data uncertainty modeling.

prints. We compare the evaluation of center-blocked fingerprints with the evaluation of other blocking patterns that introduce missing regions randomly at different locations. As shown in Figure 2(b), the performance degradation due to center blocking and random blocking are similar when OR is smaller 25%. We hence use OR to sensibly measure the information loss.

## D. Uncertainty Modeling

In this section, we introduce the implementation details of DUL and study utilizing the captured data uncertainty to construct robust matching score. Then, we discuss model uncertainty modeling for partial fingerprint recognition.

There are two types of uncertainty: data uncertainty and model uncertainty [2]. Data uncertainty is inherent to the data itself, such as its low quality or noise level. Model uncertainty, on the other hand, correlates to the density of the data. Data that are distinct from the training data are more likely to result in larger model uncertainty.

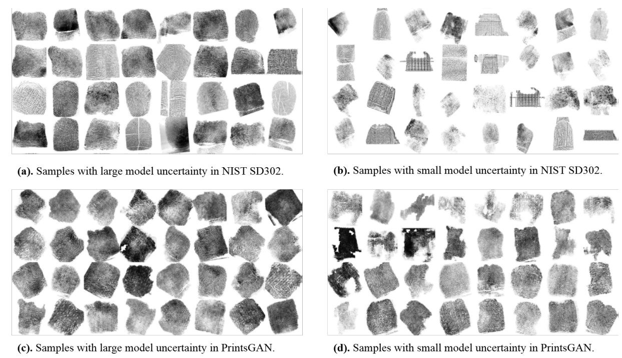
**Data Uncertainty Modeling.** To capture data uncertainty, the framework we employ is illustrated in Figure 3. The training loss for training sample $i$ is

$$\mathcal{L}_{cls,i} + \lambda_{kl}\mathcal{L}_{kl-reg,i}, \tag{1}$$

where $\mathcal{L}_{kl-reg,i}$ is the Kullback-Leibler (KL) divergence, constraining the output feature embedding distribution $\mathcal{N}(\mu, \sigma)$ to be close to a normal distribution, and $\lambda_{kl}$ is the corresponding weight. Instead of estimating a vector, we only predict a scalar value as the variance of a sample. During training, we add the DUL head to trained RPF and train the whole model end-to-end. The loss weight on kl regularization term is 0.1. We train the models 25 epochs with learning rate of 1e-5.

Table 1. Evaluation of using data uncertainty modeling on PrintsGAN, and NIST. DUL(S) indicates using data uncertainty for computing robust matching score.

| Methods | PrintsGAN, FRR@ | | NIST, FRR@ |
|---|---|---|---|
| | FAR=0.1% | FAR=0.01% | FAR=0.1% |
| DUL | 0.018 | 0.04 | 16.36 |
| DUL(S) | 0.017 | 0.038 | 16.27 |



**(a).** Samples with large model uncertainty in NIST SD302.

**(b).** Samples with small model uncertainty in NIST SD302.

**(c).** Samples with large model uncertainty in PrintsGAN.

**(d).** Samples with small model uncertainty in PrintsGAN.

Figure 4. Samples with large/small model uncertainty in NIST and PrintsGAN.

Given the captured data uncertainty can effectively detect the partial fingerprints, we propose to further utilize the learned uncertainty to construct a robust matching score function as:

$$score = -\log(\sigma_T^2 + \sigma_Q^2) - (\mathbf{e}_T - \mathbf{e}_Q)^t \mathbf{\Sigma} (\mathbf{e}_T - \mathbf{e}_Q), \tag{2}$$

where $\mathbf{\Sigma} = \sigma_T^2 + \sigma_Q^2$, $\{\sigma_T, \sigma_Q\}$ are the predicted variance, and $\{\mathbf{e}_T, \mathbf{e}_Q\}$ are the normalized feature vector for the template and query fingerprint.

We present the evaluation results of using data uncertainty to construct more robust matching score in Table 1. Comparing to the model only incorporating the data uncertainty into the feature learning process (DUL), further incorporating the data uncertainty to compute robust matching scores provides extra improvements.

**Model Uncertainty Modeling.** We capture model uncertainty using MC dropout [1]. The method can be described as: During training, we include dropout layers in the model and follow the same strategy for training. During testing, we do not turn off the dropout layer and run the model $M$ times from where we obtain $M$ feature predictions $F_1, ..., F_M$ for an input image. By definition, model uncertainty is the variance of the predicted feature vector. We approximate the model uncertainty with sample variance of $F_1, ..., F_M$ and obtain a scalar value by calculating the trace of the sample covariance matrix. Current fingerprint data are mostly complete, while partial fingerprints are rare cases in the training data. Partial fingerprint data hence should have larger model uncertainty.

We studied whether the captured model uncertainty can effectively characterize the complete fingerprints with high confidence while the partial prints with low confidence. Sample fingerprints with small and large model uncertainty are shown in Figure 4. Partial fingerprints should have large model uncertainty as they typically are rare cases in the training data. Not as expected, the incomplete fingerprints have smaller model uncertainty. When doing experiments, we find using Arcface loss leads to such "inverse" results. The models are confident on these rare cases.

## References

[1] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016. 2

[2] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017. 2