1 INTRODUCTION

Smartphone biometrics usage increases with time, attributed to high-quality smartphone cameras, increased compute capability, and dedicated sensors on smartphones. The broader use of smartphone biometrics is due to their portability, cost-effectiveness, and growing consumer market acceptance (Das et al., 2018). With the advent of smartphone-banking applications such as Apple-Pay (App, ), it is essential to have biometric authentication-based solutions in them (Stokkenes et al., 2018). Traditionally biometric authentication is performed using face, iris, and fingerprint modalities. Apple Face ID (App, 2017), & Touch ID (App, 2013) are used for face and fingerprint authentication on an iPhone, which provides a high level of accuracy. However, these require dedicated sensors, which are an additional cost to the smartphone manufacturer.

Recently, Finger-Photo, a.k.a 2D touchless fingerprint, a new modality for biometric verification, has emerged due to its direct usage with a smartphone camera. Finger-Photo verification is an active research area, as pointed out in a recent survey by Busch et al. (Priesnitz et al., 2021). Malhotra et al. (Malhotra et al., 2017) performed a short survey on Finger-Photo recognition with the smartphone as a capture device. Labati et al. (Labati et al., 2019) conducted a more detailed survey on fingerprint recognition systems, including their weakness and challenges. The main challenges in Finger-Photo verification systems based on smartphones are sensor-to-finger distance, sharpness, quality, and focus, which can be alleviated using preprocessing, the region of interest (ROI), and a robust color space (Priesnitz et al., 2021). Stein et al. (Stein et al., 2012) had one of the early works in finger photo recognition. They developed an Android Application for this purpose which took a sequence offinger photo images in multiple lighting conditions, followed by Region of Interest (ROI) extraction and binarization. The matching scores on binarized images were computed by open-source minutiae extractor FingerJetFXOSE from Digital Persona (Persona, 2020). However, this method was unable to handle defocussed finger photo images. Lee et al. (Lee et al., 2005b) by a real-time scheme that took the most focussed image from an image sequence addressed the issue of sharpness during finger-photo capture and was achieved by using Variance-Modified-Laplacian of Gaussian (VMLOG) algorithm. The issue of contrast, and color which is an important issue, has been addressed by many authors. Lee et al. (Lee et al., 2005a) handled this issue by using skin color properties & guided machine learning. The main limitation of their approach was that they required user input for Finger-Photo segmentation. Malhotra et al. (Malhotra et al., 2017) handled this by using the magenta channel in CMYK color space. Raghavendra et al. (Raghavendra et al., 2013) proposed an approach using the Mean Shift Segmentation (MSS) based sys-
tem. They used multiple features after MSS consisted of preprocessing and scaling to segment the finger in challenging real-world environments accurately.

Wasnik et al. (Wasnik et al., 2018) used the Frangi Vesselness filter (Frangi et al., 1998) filter to convert a Finger-Photo to a fingerprint-like pattern as shown in Figure 1. However, their method requires commercial-off-the-shelf (COTS) for verification. Malhotra et al. (Malhotra et al., 2020) proposed the use of Invariant Scattering Networks from Bruna et al. (Bruna and Mallat, 2013). The input to their method is a full-frame finger photo which was then segmented using a weighted combination of visual saliency (Erdem and Erdem, 2013) followed by OTSU thresholding (Otsu, 1979), and Skin Colour based segmentation (Sawicki and Miziolek, 2015). This is followed by computing wavelet-like features from the second-order decomposition of the cropped finger photo image (Bruna and Mallat, 2013). The dimension of these features is very high \(209 \times \text{width}/8 \times \text{height}/8 = 809875\), for which they use PCA (Joliffe, 1986) to reduce the dimension to 99% of its energy. This is followed by the use of Random Decision Forests (Ho, 1995).

1.1 Contributions of the Paper

We summarize the contributions of the proposed approach:

- Finger-Photo segmentation is a challenge for most of the presented methods. This issue is not entirely resolved in previous methods as some part of the background which acts as noise for the matching algorithm is present. We handled this issue by the Mask RCNN based Finger-Photo segmentation and produced a tightly cropped Finger-Photo. We require only a loosely cropped bounding box as an input to the Mask RCNN network.

- We provide an end-to-end approach for Finger-Photo verification, including segmentation and matching. The matching is done by the use of the proposed siamese neural network. This is another contribution of this paper, as most previous methods depend on either COTS or existing open-source algorithms.

- The proposed approach uses the proposed convolutional neural (SNN) decisions directly as the classification labels.

- We define challenging protocols for matching to simulate the real-world scenario where the data environment or subjects are not seen by the verification algorithm beforehand.

In the rest of the paper, we present the proposed method in Section 2, describe the experimental setup & results in Section 3, and conclude the paper by providing conclusions & future work in Section 4.

2 PROPOSED METHOD

In this section, we describe the proposed method as shown in Figure 2. The proposed method consists of the following components, Finger-Photo segmentation & cropping for a pair of input images, feature extraction with the proposed Siamese Network-based Architecture for finger photo verification, and then classification. We now describe each of these components in the following sub-sections:

2.0.1 Finger-photo Segmentation & Cropping

We describe the approach for finger-photo segmentation in this subsection. The public dataset consists of finger videos from which frames are extracted. This is followed by bounding box-based cropping, where a loose bounding box with some extra region is defined on the once for the dataset basis. The bounding box crop is followed by Finger-Photo segmentation using fine-tuned MaskRCNN Network from He et al. (He et al., 2017). Then we apply a Region of Interest (ROI) on the segmented Finger-Photo. The stages are shown in Figure 2. This approach’s main advantage is that the user does not need to do either finer level of cropping or bounding box generation for a single Finger-Photo (Wasnik et al., 2018) (Lee et al., 2005a).

2.0.2 Classification using Proposed Siamese Architecture based Network

Feature Extraction uses Siamese Neural Network Architecture proposed by Chopra et al. (Chopra et al., 2005) for Face Verification which consists of two convolution neural networks with shared weights. Our
proposed siamese network has the architecture as shown in Figure 2. It consists of the identical layers for both left, & right halves, and each half includes a Resnet-50 as the backbone network with an image dimension of $208 \times 224$. The output is taken at the last activation of 2048 dimensions connected to our custom layers of flattening, a dense layer (1024 dimensions with ‘relu’ activation), a dense layer (128 dimensions with sigmoid activation). A distance layer (‘euclidean’ distance) is then used. We choose Resnet as the base network as it offers good generalization according to He et al. (He et al., 2020), and Resnet-50 because it provides a balance between computing required, & accuracy. We further use only a small number of layers for fine-tuning as the number of images we use in training are much lower than Imagenet dataset (Deng et al., 2009). The loss function used during training is binary cross-entropy, mainly chosen as a balanced two-class classification problem. The features for each Finger-Photo are computed as the output from the dense layer of 128 dimensions. The features are extracted from the final dense layer of 128 dimensions, and euclidean distance is computed between them to perform classification. We then apply a threshold of 0.5 to classify the input pair of fingers as genuine or impostor. We use a classifier-less system, and our proposed approach does not require other classifiers after using the proposed network.

### 3 EXPERIMENTAL SETUP & RESULTS

This section describes the training methodology, results, and a discussion on the obtained results.

#### 3.1 Dataset Details

We now describe the generation of training, validation, and testing pairs. We use the public dataset (Raghavendra et al., 2020) for training and evaluating our proposed method. The dataset consists of 200 unique fingers, where each finger is captured in 5 sessions, 2 sample videos with 200 frames each. Thus, overall number of frames in the current dataset uses is $200 \times 5 \times 2 \times 200 = 40000$ which is summarized in Table 1.

#### 3.2 Protocol Details

##### 3.2.1 Training Mated/Non-mated Pairs

We include 100 subjects for training, and for training mated pairs, a total of 50 frames are selected from the two video samples, three pairs for each frame. The frame number is the same in the first pair, and in the second & third pairs, the frame numbers are...
chosen randomly between 0 to 50. This makes the total number of mated pairs $50 \times 3 \times 100 = 15000$, which due to failure to extract from Mask RCNN of some samples is 13800. For training non-mated pairs, we choose all pairs of fingers ($100 \times 99$, with multiple pairs for each selected pair of a finger (2), we thus get $100 \times 99 \times 2 = 18000$, which due to failure to extract from Mask RCNN of some samples is 16744.

### 3.2.2 Validation Mated/Non-mated Pairs

For validation mated pairs we select 25 frames, and 3 pairs, giving $(100 \times 25 \times 3 = 7500$, with 6900 actual pairs), and validation non mated pairs, we select 1 pair giving $(100 \times 99 = 9900$, with 8372 actual pairs).

### 3.2.3 Testing Mated/Non-mated Pairs

We include the rest 100 subjects for testing, and for testing mated pairs we select 20 frames, and 3 pairs for each frame, which makes the total number of testing mated pairs $100 \times 20 \times 3 = 6000$ with 5760 actual pairs, and testing non mated pairs are $100 \times 99 = 9900$ with 9120 actual pairs.

### 3.3 Evaluation Protocols

- **TD1**: Testing on the Same Session, with Unseen Reference and Probe Subjects.
  This protocol uses the 100 subjects from the testing partition to evaluate the proposed method, which gives S2 vs. S2.

- **TD2**: Testing on a Different Session, with Seen Reference and Unseen Probe.
  In this protocol, we use the 100 subjects used during training where reference is from the training session 2, and a probe is from unseen session 2 to 6, which gives S2 vs. S3, S2 vs. S4, S2 vs. S5, and S2 vs. S6.

- **TD3**: Testing on Different Session, with Unseen Reference and Unseen Probe.
  In this protocol, we use the 100 subjects from the testing partition where reference is from the testing partition of session 2, and the probe is from
Table 3: EER of the built models.

<table>
<thead>
<tr>
<th>Sessions</th>
<th>Error Equal Rate (EER)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Baseline</td>
</tr>
<tr>
<td>TD1</td>
<td>S2 vs S2</td>
</tr>
<tr>
<td>TD2</td>
<td>S2 vs S5</td>
</tr>
<tr>
<td>TD3</td>
<td>S2 vs S5</td>
</tr>
<tr>
<td>TD4</td>
<td>S4 vs S4</td>
</tr>
<tr>
<td>TD4</td>
<td>S5 vs S6</td>
</tr>
</tbody>
</table>

unseen session 3 to 6, which gives S2 vs. S3, S2 vs. S4, S2 vs. S5, and S2 vs. S6.

- **TD4:** Testing on Different Session, with Unseen Reference and Unseen Probe.
  In this protocol, we use the 100 subjects which are from the testing partition where reference is from the testing partition of session 3 to 6, and the probe is from unseen session 3 to 6, where we choose S3 vs. S3, S4 vs. S4, S5 vs. S5, S5 vs. S6, and S6 vs. S6.

### 3.4 Training Methodology

We now describe the finger photo segmentation performed using Mask RCNN post the bounding box-based crop. Mask RCNN network is fine-tuned using 40000 manually segmented images using the Matlab Image Segmenter Tool. The fine-tuning is performed for 100 epochs on a standard laptop. The proposed Siamese-based network is trained for 100 epochs on NVidia Tesla P40 GPU using SGD optimizer and learning rate of 0.001, the momentum of 0.9, and weight decay 0.1 with Nesterov solver. The loss curve obtained during the training is shown in Figure 3. The accuracy shows overfitting in the current network design, but the fact validation accuracy is close to 96% is helpful in an uncontrolled scenario.

### 3.5 Results

The error metrics are used in accordance with (ISO/IEC TR 2382-37:2012, 2012) where False Match Rate (FMR) is the number of non-mated comparisons which result in a false match, and False Non-Match Rate (FNMR) is the number of mated comparisons which result in false non-match. The plot of FMR vs. FNMR is the DET (Detection Error Tradeoff) Curve and EER (Equal Error Rate), which is the threshold where FMR equals FNMR. The DET Curves are shown in Figure 4. DET Curve is the False Match Rate (FMR) vs False Non-Match Rate (FNMR) plot. Table 3 shows EER for a few different cases from different protocols.

The EER is shown in tabular form in Table 3, and as DET Curves in Figure 4. We perform data augmentation (DA) over the baseline proposed method. The augmentation techniques include increasing/decreasing brightness, sharpness, or Gaussian noise to samples randomly during training/testing, which improves performance as shown in Table 3. In terms of data normalization, we use Frangi Filter output images instead of ROI images in the baseline model, which further improves many protocols’ performance.

### 3.6 Analysis of Results

The performance difference of the proposed method with & without data augmentation/normalization is analyzed as follows:

- The proposed approach provides an end-to-end system for score computation and is currently not robust to illumination, brightness, and contrast changes as the baseline EER is high. However, this is alleviated to a certain extent by the use of data augmentation techniques which make it slightly robust to these changes.
- The proposed approach, when used in combination with data normalization based on the generation of fingerprint-like patterns which are obtained using Frangi filter (Frangi et al., 1998) results in lower error rates.
- The performance of the proposed approach & DA achieves a low EER of 8.95% when both the train & test session are the same (TD1). This is mainly due to the fact as transformations of illumination, brightness, and contrast are low.
- The performance for an unseen session during the testing is lowest for baseline + DN, where we achieve an EER of 20.53% for TD2. This can be attributed to DA., and DN doesn’t generalize to the unseen testing session scenario.
- The current feature extraction technique has issues in generalization to unseen training data, attributed to different capture conditions including illumination, contrast, and brightness.
(a) TD1: S2 vs S2 (same session, with unseen reference and probe subjects)
(b) TD2: S2 vs S5 (different session, with seen reference and unseen probe)
(c) TD3: S2 vs S5 (different session, with unseen reference and unseen probe)
(d) TD4: S4 vs S4 (different session, with unseen reference and unseen probe)
(e) TD4: S5 vs S6 (different session, with unseen reference and unseen probe)

Figure 4: DET Curves for different evaluation protocols of the proposed method.

4 CONCLUSIONS & FUTURE-WORK

In this paper, we presented an end-to-end system for finger photo verification. The method presented in this paper has several advantages over previous techniques. The first is the use of MaskRCNN for finger photo segmentation, which allows the user to have a loose bounding box, unlike previous approaches. The second advantage is that the system behaves as an end-to-end system without the use of COTS for verification. We performed an extensive evaluation for both seen and unseen session scenarios. We would make the network robust to noise, illumination, and quality changes in the input images, especially to achieve good performance in an unseen testing session. This could be achieved in multiple ways, such as using different color spaces that are more robust to these changes, such as CMYK or CIE Lab. The robustness to an unseen testing session can be improved by using more data augmentation & data normalization techniques. We want to compare our proposed approach with more methods in SOTA (Malhotra et al., 2020) & datasets, and especially for a cross-dataset scenario in future work.

ACKNOWLEDGMENT

This work is carried out under the partial funding of the Research Council of Norway (Grant No. IKT-PLUSS 248030/O70).

REFERENCES


