Chapter 2
User Authentication via Finger-Selfies

Aakarsh Malhotra, Shaan Chopra, Mayank Vatsa and Richa Singh

Abstract In the last one decade, the usage and capabilities of smartphones have increased multifold. To keep data and devices secure, fingerprint and face recognition-based unlocking are gaining popularity. However, the additional cost of installing fingerprint sensors on smartphones questions the use of fingerprints. Alternatively, finger-selfie, an image of a person's finger acquired using a built-in smartphone camera, can act as a cost-effective solution. Unlike capturing face selfies, capturing good-quality finger-selfies may not be a trivial task. The captured finger-selfie might incorporate several challenges such as illumination, in- and out-of-plane rotations, blur, and occlusion. Users may even present multiple fingers together in the same frame. In this chapter, we propose authentication using finger-selfies taken in an unconstrained environment. The research contributions include the UNconstrained FIngerphoTo (UNFIT) database which is captured under challenging unconstrained conditions. The database also contains the manual annotation of identities and location of the fingers. We further present a segmentation algorithm to segment finger regions and, finally, perform feature extraction and matching using CompCode and ResNet50. Experimental results show that despite multiple challenges present in the UNFIT database, the segmentation algorithm can segment and perform authentication using finger-selfies.

Aakarsh Malhotra and Shaan Chopra: Equal contribution by student authors.

A. Malhotra · S. Chopra · M. Vatsa · R. Singh (✉)
IIIT-Delhi, Delhi, India
e-mail: rsingh@iiitd.ac.in

A. Malhotra
e-mail: aakarshm@iiitd.ac.in

S. Chopra
e-mail: shaan15090@iiitd.ac.in

M. Vatsa
e-mail: mayank@iiitd.ac.in

A. Rattani et al. (eds.), *Selfie Biometrics*, Advances in Computer Vision and Pattern Recognition, https://doi.org/10.1007/978-3-030-26972-2_2

2.1 Introduction

In the current digital era, smartphones and mobile devices are ubiquitous. With the growth of smartphone usage, people store enormous amounts of personal and confidential information on their smartphones. Storing such information on smartphones demands suitable security mechanisms. Traditional security measures include passwords, patterns, or pins. However, these methods need to be memorized by the users and are vulnerable to shoulder surfing attacks [1]. Alternatively, biometric-based user authentication is now more popular and requires minimal effort from the users.

As illustrated in Fig. 2.1, modern smartphones have multiple sensors that can facilitate user authentication. For instance, cameras can be used to capture face [2] and finger-selfies, while fingerprint sensors can be used to acquire fingerprints. Researchers and commercial entities have explored the usability of all three, and each posing certain advantages and constraints. For instance, traditional fingerprints are accurate but require the installation of additional capacitive sensors [3]. Face selfies are easy to capture, but they may be affected by several external factors. Similarly, finger-selfies do not need any additional sensors, but the technology requires more

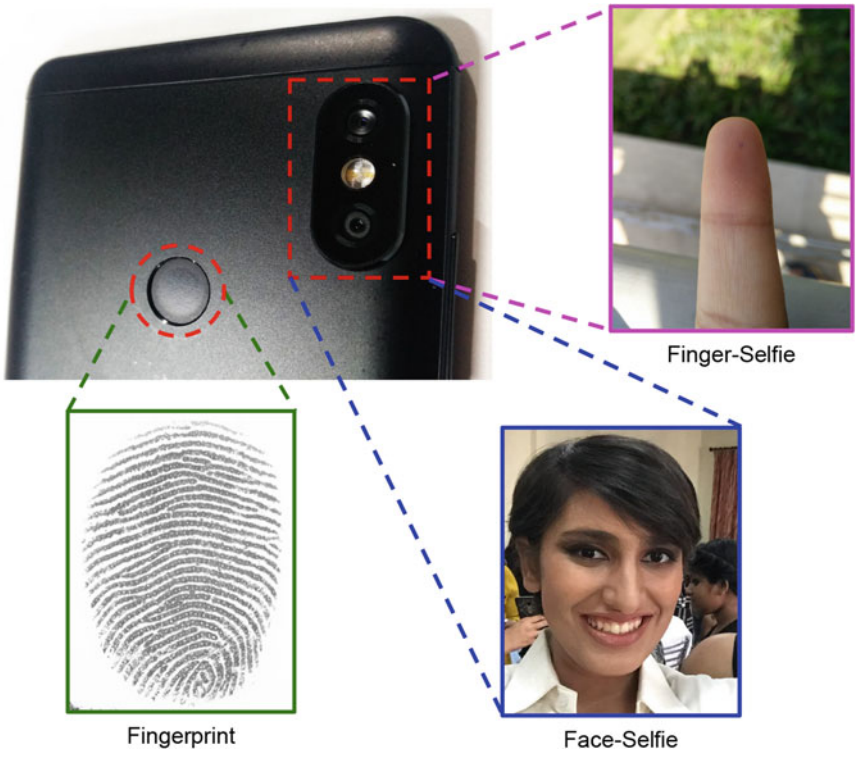

Fig. 2.1 Acquisition sensors and their corresponding captured modalities

Fig. 2.2 An illustration of acquisition mechanism of finger-selfie and the corresponding finger-selfie

research to demonstrate the effectiveness. This chapter focuses on finger-selfie, presenting a review of the research efforts related to improving the usability and accuracy of finger-selfie recognition.

As shown in Fig. 2.2, finger-selfie acquisition involves capturing ridge-valley details present on the tip of the finger using a device camera by the user. Overcoming the drawback of traditional biometric-based authentication, a finger-selfie does not require an additional sensor. All it needs is the smartphone's in-built camera. As per Tim Ahonen's Phone book [4] and Statista [5], approximately 89% of all digital photographs arise from handheld devices such as tablets and smartphones. While these statistics motivate the use of finger-selfies as a cost-effective method for authentication, there are other advantages as well. Finger-selfies act as a contactless fingerprint acquisition technique, which is hygienic and secure, leaving no latent impressions on the surface of the sensor. Over the flattened live scan fingerprints, finger-selfies also contain additional information such as finger shape and phalanx lines. While these lines may not have global uniqueness, a localized correlation with ridge-valley patterns in the neighborhood may aid person identification [6].

Other than authentication for device unlocking, law enforcement agencies have also shown their interest toward finger-selfies. For instance, on finding a finger-selfie of a potential drug dealer holding drugs on his fingers, the South Wales Police and the scientific support unit utilized the finger-selfie to identify the culprit [7]. Similarly, a hacker used an image of a German minister's finger, acquired from a distance of three meters, to generate fingerprints [8]. Such use cases highlight the need for finger-selfie-based recognition systems.

Emphasizing on the other side of the coin, finger-selfie-based user authentication is not perfect either. As illustrated in Fig. 2.3, a finger-selfie looks drastically different from a traditional fingerprint, with skin and background visible along with ridge-

(a) Finger-selfie acquired under different conditions

(b) Corresponding livescan images of the same subject

Fig. 2.3 Visual difference between a finger-selfie and a legacy fingerprint image

valley details. While its acquisition requires minimal effort from the user, their lack of cooperation might induce many challenges. Unlike capturing face selfies, acquiring a good-quality finger-selfie may not be a trivial task, and the captured finger-selfie might comprise several variations such as illumination, in- and out-of-plane rotations, blur, and occlusion. Users might even present multiple fingers in the same frame. A summary of these challenges is illustrated in Fig. 2.4. While these challenges highlight a real-life unconstrained acquisition scenario, detection and recognition of these finger-selfies for smartphone authentication become a cumbersome task.

To promote unconstrained finger-selfie-based recognition, this chapter first provides a review of existing research on finger-selfie followed by finger-selfie-based authentication in an unconstrained environment. This research is inspired by our preliminary work, which showcased the application of finger-selfies in an unconstrained environment [9]. The important research contributions of this chapter are:

1. A review of existing databases utilized in the literature for finger-selfie/image/photograph-based recognition and a detailed summary of existing approaches for finger-selfie recognition are discussed.
2. A novel publicly available UNconstrained FIngerphoTo (UNFIT) database, which is captured under challenging unconstrained conditions. The database also contains manual annotation of identities and location for 3450 images from 115 subjects.

Fig. 2.4 Sample finger-selfie images from the proposed UNFIT database. While the database incorporates numerous challenges, a real-life unconstrained acquisition of finger-selfies might contain one or more challenges together, making finger-selfie recognition a complex problem. Varying resolutions of the camera adds to the challenges of finger-selfie recognition

3. A segmentation algorithm to segment finger regions from a finger-selfie using the existing VGG SegNet [10] model. The performance of the segmentation algorithm is compared with other segmentation methods such as FCN 8 [11]. We show that existing deep learning algorithms for segmentation can easily outperform the traditional skin color-based segmentation [12] methods used in the literature.
4. Finally, recognition of the segmented finger is performed. The benchmarking for feature extraction and matching is performed using CompCode [13] and ResNet50 [14] followed by Hamming distance and cosine similarity, respectively. Experimental results show that despite multiple challenges present in the UNFIT database, finger-selfie-based biometric authentication is feasible and pragmatic.

2.2 Related Work

Recent studies have demonstrated the usage of fingerphoto/contactless fingerprints acquired using smartphones and other digital cameras toward benchmarking of contactless fingerprint recognition. However, a significant limitation of these studies is the use of constrained or semi-constrained fingerphoto datasets. A summary of the datasets is presented in Table 2.1, and their details are given below.

Table 2.1 Literature review of existing databases of contactless fingerprints/fingerphotos

Research	Device	Subjects	# Samples	Challenges	Public	Nature
Song et al. [15]	CCD	–	–	None	✗	Constrained
Lee et al. [16]	Phone	150 + 168	400 + 840	Background, orientation	✗	Semi-constrained
Lee et al. [17]	Phone	15	60 + 30 + 30 videos	Blur, orientation/ movement	✗	Semi-constrained
Piuri and Scotti [18]	Webcam	15	150	Background	✗	Semi-constrained
Hiew et.al [19]	Digital Camera	103 classes	1938	None	✗	Constrained
Kumar and Zhou [6]	Webcam	156	1566	Resolution	✓	Semi-constrained
Derawi et al. [20]	Phone	22	1320	None	✗	Constrained
Yang et al. [21–23]	Phone	25	2100	Background, illumination	✗	Semi-constrained
Stein et al. [24]	Phone	11 + 37	66 videos, 990 photographs	None	✗	Constrained
Tiwari and Gupta [25]	Phone	50	150	Illumination	✗	Constrained
Sankaran et al. [12]	Phone	64	4096	Background, illumination	✓	Semi-constrained
Taneja et al. [26]	Multiple	64	8192	Fingerphoto Spoofing	✓	–
Lin and Kumar [27]	-	300 classes	1800	None	✓	Constrained
Proposed	**Phone**	**230 classes**	**3450**	Background, blur, multiple fingers, illumination, affine variation, resolution, deformations	✓	**Unconstrained**

2.2.1 Existing Databases

Several researchers have designed algorithms and shown results on contactless fingerprint recognition. However, a significant limitation related to the research on finger-selfie recognition is the unavailability of public datasets. While four of the datasets are publicly available, these datasets incur just one or two variations, which lack common challenging scenarios of acquisition present in finger-selfies. A summary of these datasets is presented below.

2.2.1.1 Publicly Available Databases

As illustrated in Table 2.1, there exist databases for contactless fingerprints; however, for benchmarking and algorithmic evaluation, only the following databases are publicly available in the research community:

- HKPU Low-Resolution Fingerprint Database [6]: The database has a total of 1566 low-resolution contactless fingerprint images from 156 subjects. The contactless fingerprints are acquired using a webcam in two different sessions. While the database is acquired at a low resolution, it incorporates no other challenge during acquisition. Hence, the database can be termed as semi-constrained.
- IIITD Smartphone Fingerphoto Database [12]: In 2015, Sankaran et al. proposed this database, containing 4096 fingerphoto images from 64 participants acquired using a smartphone camera. The database also includes 1024 livescan images to promote matching of fingerphoto with legacy fingerprint databases. The subsets of the database include varying background and illumination. Hence, this database can also be considered as semi-constrained.
- PolyU Contactless to Contact-based Fingerprint Database [27]: Recently, Lin and Kumar proposed a constrained dataset, with 1800 contactless fingerprint samples from 300 different fingers. While the images of fingers were acquired in a constrained setting, the database aimed to establish the matching of contactless fingerprints with contact-based livescan fingerprints. Hence, the database also includes 1800 contact-based livescan images.
- Other than the databases mentioned above, Taneja et al. [26] proposed a Spoofed Fingerphoto Database, which aimed to establish the effect of spoofing of fingerphotos using display and print attack. This database was created using fingerphotos taken from the IIITD Smartphone Fingerphoto Database [12].

Using the in-house and publicly available touchless fingerprint databases, researchers have demonstrated benchmarking results of their proposed algorithms. A summary of these algorithms is presented below.

2.2.2 Finger-Selfie Recognition Techniques

For touchless fingerprint recognition, Song et al. [15] used only blue channel information of finger images. They utilized mean and coherence for segmentation and Gabor filters to enhance ridge details. Their results were illustrated visually on a touchless fingerprint image. In 2006, Lee et al. [16] performed segmentation by combining normalized color (RB) model and frequency information extracted using the Tenengrad method. Minutiae were extracted from the segmented image, following which the authors reported about 80% GAR at 0.01% FAR. In 2008, Lee et al. [17] aimed at focus estimation by estimating blur. They also used coherence and symmetry for quality estimation and difference in frames (contour extraction) for pose estimation. On the Samsung Database (SDB)—I, II, III, IV—with 60, 30, 30 image sequences and 1200 fingerprint images, respectively, authors reported a rejection rate of 5.67% and EER of 3.02%.

Piuri and Scotti [18] performed blur reduction using Lucy-Richardson algorithm and Wiener filter algorithm followed by color model and morphology-based segmentation. After performing fingerphoto registration, enhancement, and minutia extraction using MINDTCT, authors reported an EER of 0.042% for 150 images. Hiew et al. [19] utilized Gabor features, followed by PCA and SVM for verification. They reported an EER of 1.23%. In 2011, while proposing a publicly available dataset, Kumar and Zhou [6] performed enhancement by Sobel filtering and area thresholding on the acquired image, followed by Gaussian sharpening. Using LRT and CompCode features followed by Hamming distance, the authors reported a cross-session EER of 3.95% with 93.97% accuracy on the proposed dataset. In the same year, Derawi et al. [20] performed feature extraction and matching using COTS and reported an EER of 0.00–23.62% for different fingers on their in-house database.

Yang et al. [21–23] utilized their semi-constrained database with 2100 samples toward quality assessment of fingerprint images captured from a smartphone camera. They defined a total of seven [21] and twelve [22] quality metrics to determine the quality of contactless fingerprint image. Using the same dataset, Raghavendra et al. [23] performed mean shift clustering to segment the probable finger regions. The final finger is detected from top five-sized regions using a fusion of Pearson, Fourier magnitude, and energy measure based on the wavelet transform. They reported an average segmentation accuracy of 96.46%. Using NBIS MINDTCT for minutia extraction followed by matching, authors report an EER of 3.74%. In 2013, Stein et al. [24] performed spoof detection, followed by minutia extraction and matching. The authors reported 1.20% EER for contactless fingerprints and 3.00% EER for finger videos. Tiwari and Gupta [25] found ROI in fingerphoto by adaptive thresholding followed by morphological operations. They aligned the image using PCA followed by image enhancement using adaptive histogram equalization. Using SURF features, authors report an EER of 3.33% on their proposed in-house database.

In 2015, Sankaran et al. [12] created IIITD Smartphone Fingerphoto Database and proposed a fingerphoto-to-fingerphoto and fingerphoto-to-livescan matching algorithm. With segmentation performed using adaptive thresholding, authors per-

formed image sharpening and median filtering to enhance the image [28]. From the enhanced image, ScatNet features were extracted, followed by PCA and matching using RDF classifier. On the proposed semi-constrained dataset, authors reported an EER of 3.65–7.45% on different subsets of fingerphoto-to-fingerphoto matching and 7.07–10.43% for fingerphoto-to-livescan matching. Later, in 2017, Malhotra et al. [29] further improved the state-of-the-art performance on IIITD Smartphone Fingerphoto Database. Using an LBP-based enhancement, the authors reported an EER of 1.47–8.36% on different subsets of fingerphoto-to-fingerphoto matching and 6.44–7.61% for fingerphoto-to-livescan matching. Recently, Lin and Kumar [27] proposed a livescan and contactless fingerprint image database. To align the contactless images with livescan images, the authors proposed an RTPS-based fingerprint deformation correction model. By performing minutiae- and ridge-based matching, the authors reported a rank-1 accuracy of 94.11% using their proposed algorithm.

While these algorithms have shown good accuracies and low error rates, their performance is not evaluated in a real-life scenario of unconstrained finger-selfie recognition. A primary reason is the absence of an unconstrained finger-selfie database. To address this concern and to promote finger-selfie recognition in an uncontrolled scenario, we present UNFIT: an unconstrained fingerphoto database in the next section.

2.3 UNconstrained FingerPhoto (UNFIT) Dataset

In Sect. 2.2.1.1, we highlighted publicly available databases for contactless fingerprint recognition. While these datasets have an ample number of samples, these samples are acquired in a constrained or semi-constrained environment. In this research, we create the first unconstrained fingerphoto (UNFIT) database and make it available for the research community.[1] The database has many challenges, which would be present in a finger-selfie acquired in an uncontrolled environment with minimal user cooperation. The details of the dataset are presented below.

2.3.1 Database Acquisition

Forty-five different smartphones belonging to the subjects are used to capture finger-selfies. This brings variations in terms of resolution and camera sensor to the database. OnePlus and iPhone devices are used to acquire 48% of images in the database followed by other phones including Redmi devices, Google Nexus, Lenovo K3 Note, Lenovo K4, Mi 4, Le 1s, Samsung Galaxy, Micromax Canvas, Moto G, Moto C, Moto M, and HTC devices. The camera resolutions of these smartphones varied from 8 to 16 MP. The distribution of different smartphone devices used for finger-selfie acquisition can be seen in Fig. 2.5a.

[1] The UNFIT database can be downloaded from: http://iab-rubric.org/resources/UNFIT.html.

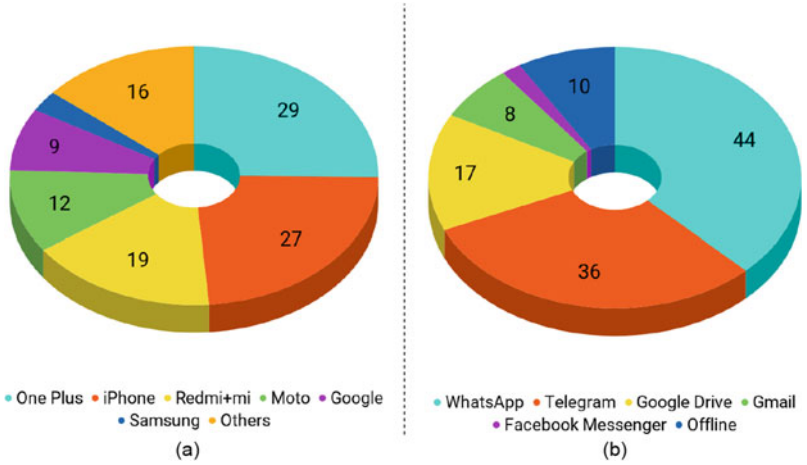

Fig. 2.5 Acquisition details: **a** Devices used for finger-selfie acquisition, and **b** Offline and online mechanisms used for obtaining finger-selfies

The database is collected via both online and offline methods which helps incorporate the effect of image compression due to transmission. WhatsApp, Telegram, Google Drive, Gmail, and Facebook messenger are used for online data collection, whereas for offline data collection, different phone devices belonging to the subjects are used followed by transmission via a pen drive. Figure 2.5b shows the distribution of images collected using different modes of online and offline data collection. Adding on, variations in illumination, intensity, and blur are present in the database due to the optional usage of auto-focus and flash for acquiring finger-selfies.

During database acquisition, no constraints are enforced for distance of the finger from the camera sensor. Varying distance allows the presence of more challenges, such as position and scale variation. However, the appearance of ridge-valley details stays limited with respect to the camera sensor. The minimum and maximum distances for a focussed detailed acquisition depend upon the camera's aperture and len's focal length. With 45 different smartphones used to obtain finger-selfies, the camera's aperture and len's focal length vary across the smartphone devices. Hence, a generic claim for a minimum and maximum distance for a focussed image cannot be made. Thus, varying sensors, lens, the distance of finger, illumination, and background variations makes locating, segmenting, and recognizing ridge-valley details in the finger challenging.

2.3.2 Database Statistics

Over a span of three months, we collated a novel finger-selfie database consisting of 3450 images and termed it as Unconstrained FIngerphoTo (UNFIT) database. The

database has multiple images of the index and the middle finger for each subject, where both the fingers of the same participants are considered as different classes. We refrained from acquiring thumb finger-selfies since capturing frontal region of thumb while holding a phone facing downward in the other hand is inconvenient for subjects. During acquisition, the participants are allowed to use either of the hand for capturing the finger-selfies, as long as all the finger-selfies arise from the same hand. The database contains 230 different classes belonging to 115 participants. Out of the 115 subjects from whom finger-selfies were captured in the UNFIT database, 38 were female participants, and 77 were male participants. The details of the database can be seen in Table 2.2. Figure 2.6 exhibits some sample images from the database. Two different sets of finger-selfies are collected from each subject:

- **Set I: Single Finger**—Images of the index and middle fingers belonging to the same hand of a user are captured. Finger-selfies are collected from either the left

Table 2.2 A summary of various subsets presents in the UNFIT database

Subset	Fingers	Classes	Images
Set I	Index	115	1150
	Middle	115	1150
Subtotal:		230	2300
Set II	Multiple fingers	115	1150
Total:			3450

Fig. 2.6 Sample finger-selfie images from different subsets of the proposed UNFIT database

or right hand of the user as per his/her convenience without enforcement of any constraints regarding background, illumination, resolution, position, or orientation of the finger. Figure 2.6a and b demonstrates sample images belonging to this set. The set contains a total of 2300 images (=115 subjects $\times$ 2 fingers $\times$ 10 instances per finger).

- **Set II: Multiple Fingers**—At times, users may capture multiple fingers, intentionally or unintentionally, and this additional information can be useful for improving finger-selfie recognition performance. Thus, this is useful for demonstrating the effect of multiple fingers on finger-selfie recognition. Figure 2.6c shows the sample images belonging to this set. The set contains a total of 1150 samples (=115 subjects $\times$ 10 instances per participant) of both index and middle fingers belonging to the same hand taken together.

2.3.3 Challenges

In a scenario where the user cooperation is minimal, intra-class variations may increase. Some of these variations are shown in Fig. 2.4. A detailed description of challenges included in the proposed UNFIT database is as follows:

- **Affine variations**: Finger-selfie acquisition involves presenting the finger in front of the rear or front camera of the smartphone. While this task sounds trivial, there can be enormous affine variations. These variations may include translation and rotation of finger. Rotation variation may be caused both by rotation of finger in the 2D image plane (Fig. 2.4c–d) and by rolling of the finger on axis of the finger. While rotation in the 2D image plane does not lead to any information loss, a rotation along the finger axis may result in different amount of acquired ridge-valley detail. The varying distance from the acquisition camera would result in scale variations.
- **Multiple fingers**: As a part of the UNFIT dataset, index and middle fingers are collected together. While the multiple fingers can be placed in any order and may experience all variations a single finger can, multiple fingers may encounter other challenges as well. As illustrated in Fig. 2.4e–f, the multiple fingers may be split or may be presented together. The split-finger scenario aids in the robust testing of segmentation algorithms, since the algorithms should be able to segment the fingers in both situations.
- **Illumination**: The finger-selfies can be captured in both indoor and outdoor environments. It induces illumination variations, which may result in dull or bright finger-selfies. Usage of camera flash, as illustrated in Fig. 2.4h, may result in targeted bright regions too.
- **Background**: Allowing any natural background to be present, finger-selfies may have similar looking backgrounds. Adding on, there may be regions in the background with skin (Fig. 2.4k). In such a scenario, selection of salient fingers becomes a tedious task.

- **Blur:** During the capture process, a common problem is unfocused acquisition of an image. It may lead to a blurred finger-selfie due to which ridge-valley details might not be prominent. Similarly, finger-selfie may incur motion blur due to hand movement or unstable holding of smartphones.
- **Deformation:** In some cases, participants provided finger-selfies with crooked fingers.

2.3.4 Ground-Truth Annotation

Due to various challenges incorporated in the proposed database (as mentioned in Sect. 2.3.3), the position and appearance of fingers in the images vary. To determine the exact location of the finger, it is necessary to generate ground-truth annotations for the same. A segmentation tool is developed in MATLAB using Piotr Dollar's toolbox [30]. The GUI of the toolbox allows the user to utilize rotatable and resizable rectangular boxes to manually bound the finger region. With a rectangular region representing a finger region, only a minimal amount of background pixels are labeled as foreground. It acts as a loose bound for the finger, making sure that there is only a negligible loss of ridge-valley details. The rectangular region can easily be cropped and fed to recognition modules. The ground-truth annotations, which are represented as a mask, are also publicly available along with the database with the same image name in a different folder.

2.3.5 Experimental Protocol

As mentioned in Sect. 2.3.2, the UNFIT database is collected from 115 subjects with 30 images taken from each participant. While training and testing, a 50:50 subject disjoint split is maintained. Hence, training data includes 1740 images corresponding to 58 subjects, and testing data consists of remaining 1710 images from 57 participants. The index and middle fingers of the same subject are considered as different classes, resulting in 116 classes during training and 114 classes while testing. During testing, the first five images of each case (index, middle, or both fingers) are treated as the gallery, whereas the remaining images (sample #6–10) are considered as the query images. While generating scores, the genuine scores are generated when index–index, middle–middle, and multiple–multiple fingers of the same subjects are matched. All other combinations of match scores generated by matching query with gallery images are treated as imposter scores.

2.4 Segmentation Framework

The unique and discriminative features of a fingerprint lie in its ridge-valley pattern. These details are present on the finger-tip, which constitutes for the foreground of the finger-selfie image. Hence, a framework is presented which aims to discard the background pixels and keep only the foreground information. A summary of the segmentation framework is illustrated in Fig. 2.7, and its details are elaborated below.

2.4.1 Segmentation Using VGG SegNet

The segmentation framework primarily utilizes VGG SegNet for classifying pixels as foreground or background. The VGG SegNet architecture has encoder and decoder network. While the role of the encoder is to convert the input data into a meaningful feature map at a lower dimension, the decoder upsamples the lower-dimensional feature map. The lower-dimensional feature map is produced due to max-pooling operation after a sequential process of convolution, batch normalization, and ReLU activation to produce nonlinearity. The locations of features, which are propagated in the network after max-pooling, are stored for further computation.

The decoder network utilizes pooling indices (the ones stored during encoding) to perform a nonlinear upsampling in order to counter the effect of max-pooling. The stored pooling indices guide the decoder network to map a lower input feature map to a higher-dimensional feature map. Hence, the upsampled feature map obtained from the decoder network has a sparse representation of the input. The upsampling approach using pooling indices is a training-free method, hence reducing the number of training parameters of the model.

While pooling is known to have local invariance, in this work, a standard encoder–decoder network with pooling layers is utilized. The previous encoder–decoder architectures also use a standard pooling in their model (or global average pooling at the end of the network). It can be noted that networks that have used pooling [11, 31, 32]

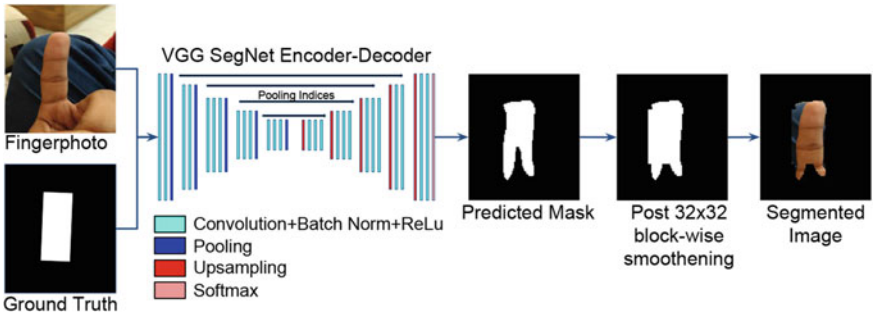

Fig. 2.7 Illustration of the segmentation framework using VGG SegNet followed by 32 × 32 block-wise smoothening

have worked well for the task of object segmentation. However, to eliminate pooling, the entire model has to be revamped and replaced by a capsule-net style architecture. Such scenario would require training from scratch, disallowing us to use pre-trained network. With a limited number of training instances, training a pooling-free network would be beyond the scope of the proposed framework.

The sparse representation is fed as input to a convolutional layer, which is succeeded by a Softmax classification layer. The Softmax layer classifies each of the image pixels as foreground or background. Thus, the VGG SegNet-based segmentation algorithm utilizes a pre-trained model of VGG SegNet for finger-selfie segmentation. The model is fine-tuned using finger-selfies. However, as we explain in Sect. 2.4.4.1, the predicted mask is tightly bound, due to which a significant foreground area is lost. Therefore, VGG SegNet architecture is succeeded by a 32×32 block-wise smoothening layer to increase the number of foreground pixels. The full segmentation pipeline is shown in Fig. 2.7. Algorithm 1 summarizes the complete segmentation algorithm.

Input: 224×224 finger-selfie image
Output: Segmented mask for finger-selfie
Fine-tune VGG SegNet Architecture using training finger-selfies and their masks;
Use trained model to predict mask for test finger-selfies;
Binarize the predicted masks;
f_p = *Count of finger (foreground) pixels;*
b_p = *Count of non-finger (background) pixels;*
N = *Number of test images;*
Region = Number of non-overlapping blocks of dimension 32×32 pixels in a finger-selfie;
while $N \neq 0$ **do**
  Divide test image into blocks of size 32×32 pixels;
  while *Region* **do**
    if $f_p \geq b_p$ **then**
      Set all pixels of the region as foreground;
    else
      Do not update any pixels of the region;
    end
    Region = Region - 1;
  end
  $N = N - 1$;
end

Algorithm 1: Algorithm for finger-selfie segmentation using a fine-tuned VGG SegNet architecture followed by a layer of 32×32 block-wise smoothening.

2.4.2 Implementation Details

To train the VGG SegNet + 32×32 block-wise smoothening network, finger-selfies of size $224 \times 224 \times 3$ are used along with their corresponding ground-truth annotation of size $224 \times 224 \times 1$. As illustrated in Fig. 2.7, VGG SegNet consists of an encoder and a decoder network. The output dimension of encoder network is

$14 \times 14 \times 512$. This multi-channel output is fed to the decoder network, which in turn gives an output of dimension $112 \times 112 \times 2$. The output of the decoder network serves as input to the Softmax layer, whose task is to provide a binary prediction for each pixel. The white pixel in the binary predicted mask represents the finger region, whereas the black pixel represents the background. Similar to VGG SegNet, FCN 8 is also provided finger-selfies and its corresponding ground-truth annotation.

The VGG SegNet and FCN 8 architectures are fine-tuned using an augmented training set. The augmented training data is created by increasing the original training set with mirror flipped, intensity changed, blurred, and rotated finger-selfies. Rotation of finger-selfies is performed at three different angles: 90°, 180°, and 270°. After image augmentation, the size of the training set increases to 27600 images. The corresponding finger location annotation is generated for these augmented images from the original ground-truth annotation. Using the augmented training dataset, the deep architectures are fine-tuned for 100 epochs.

2.4.3 Performance Evaluation Metrics

To evaluate the performance of segmentation algorithm, the following metrics are used:

- Segmentation accuracy (SA):

$$SA = \frac{CPB}{TB} \tag{2.1}$$

  where CPB is a count of the correctly predicted blocks while TB is the total number of blocks.
- Foreground segmentation accuracy (FSA):

$$FSA = \frac{CPFB}{TFB} \tag{2.2}$$

  FSA is the normalized foreground segmentation accuracy, where CPFB represents the number of correctly predicted foreground blocks, normalized with respect to the total count of foreground annotated blocks (TFB).
- Background Segmentation Accuracy (BSA):

$$BSA = \frac{CPBB}{TBB} \tag{2.3}$$

  BSA is the normalized background segmentation accuracy, where CPBB portrays the number of correctly predicted background blocks normalized with respect to the total count of background annotated blocks (TBB).

Figure 2.8 demonstrates a visual elucidation of FSA and BSA using the segmentation algorithm.

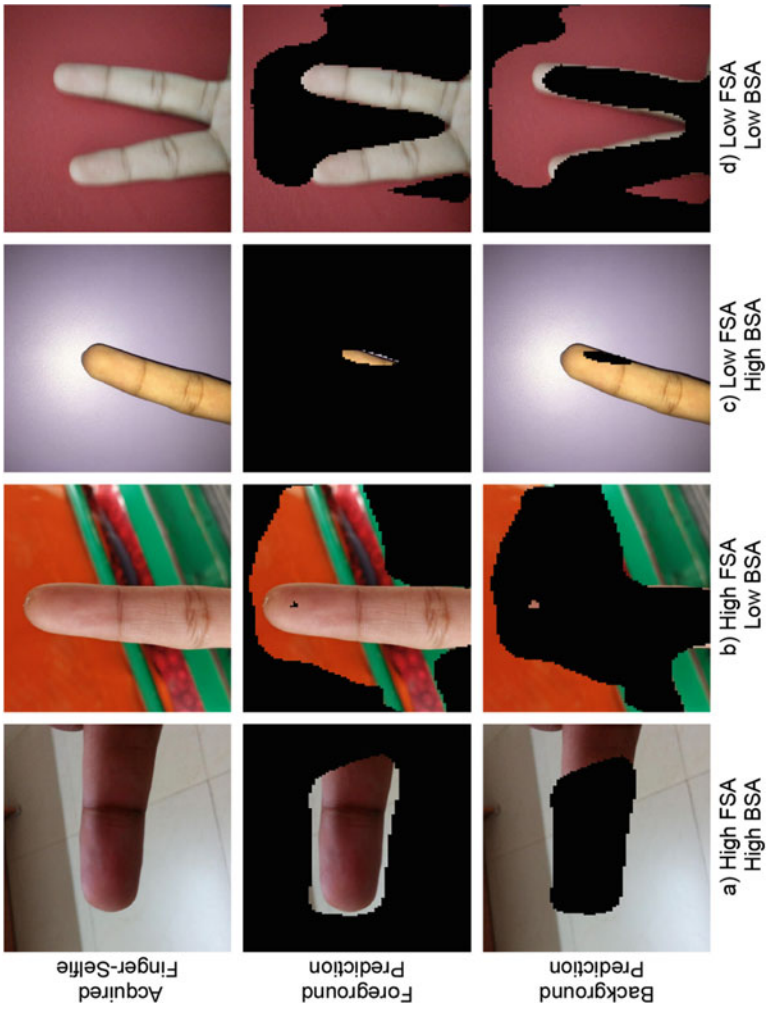

Fig. 2.8 Interpretation of FSA and BSA while segmenting finger-selfies

2.4.4 Segmentation Performance

Table 2.3 reports the segmentation performance of the algorithm in terms of FSA, BSA, and SA. VGG SegNet, along with 32 × 32 block-wise smoothening, provides the best foreground segmentation accuracy and performs well in terms of BSA and SA as well. Tables 2.4 and 2.5 illustrate a comparison of various segmentation techniques with the VGG SegNet+block-wise smoothening algorithm. Figure 2.9 shows a few samples where the segmentation framework can segment finger-selfie correctly, whereas Fig. 2.10 shows some failure cases of the segmentation algorithm.

In the proposed UNFIT database, background pixels constitute 86.21% pixels compared to 13.79% foreground pixels. While FSA is lower than BSA in Table 2.3, the reported segmentation accuracy (SA) is biased toward BSA for all fingers. This is due to higher number of background pixels in the UNFIT database as compared to foreground finger region pixels.

2.4.4.1 Effect of 32 × 32 Block-Wise Smoothening

Table 2.4 shows a comparison of the proposed architecture with VGG SegNet. For VGG SegNet, it can be observed that BSA outperforms FSA for all the fingers. The

Table 2.3 Segmentation performance of the VGG SegNet + 32 × 32 block-wise smoothening finger-selfie segmentation algorithm

Algorithm	Segmentation metric	Finger			
		All together (%)	Index (%)	Middle (%)	Multiple (%)
VGG SegNet + 32 × 32 block-wise smoothening	SA	89.04	89.89	90.62	86.61
	BSA	92.71	93.16	93.06	91.91
	FSA	71.22	70.28	74.49	68.90

Table 2.4 Comparison of the segmentation framework with VGG SegNet: illustrating the effect of 32 × 32 block-wise smoothening

Algorithm	Segmentation Metric	Finger			
		All together (%)	Index (%)	Middle (%)	Multiple (%)
VGG SegNet	SA	90.08	91.01	91.77	87.45
	BSA	94.69	95.04	94.89	94.15
	FSA	66.75	65.98	70.16	64.10
VGG SegNet + 32 × 32 block-wise smoothening	SA	89.04	89.89	90.62	86.61
	BSA	92.71	93.16	93.06	91.91
	FSA	**71.22**	**70.28**	**74.49**	**68.90**

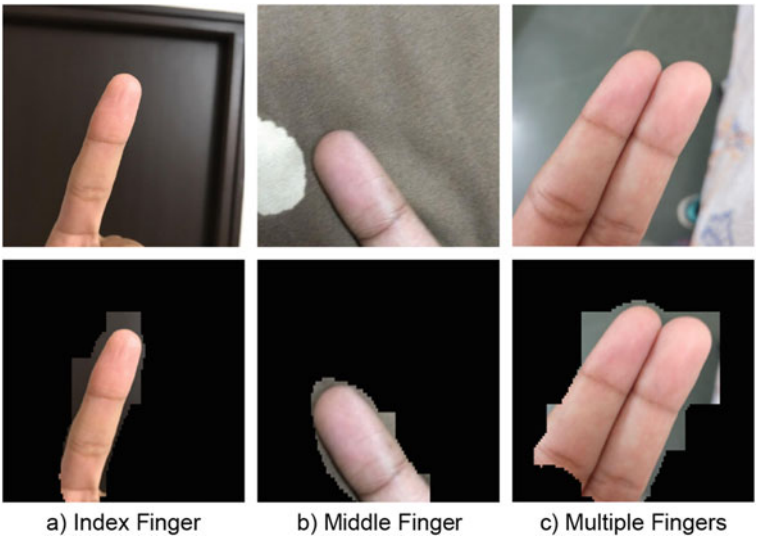

Fig. 2.9 Illustration of the successful cases of the segmentation framework

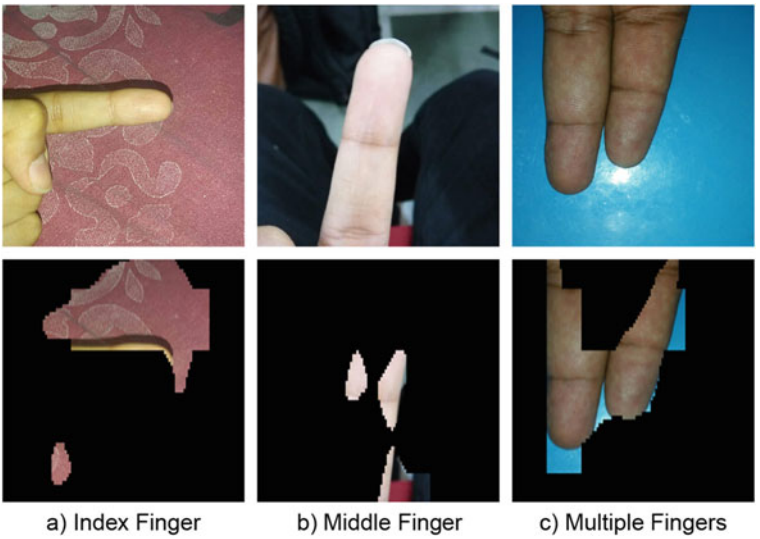

Fig. 2.10 Illustration of the failure cases of the segmentation framework

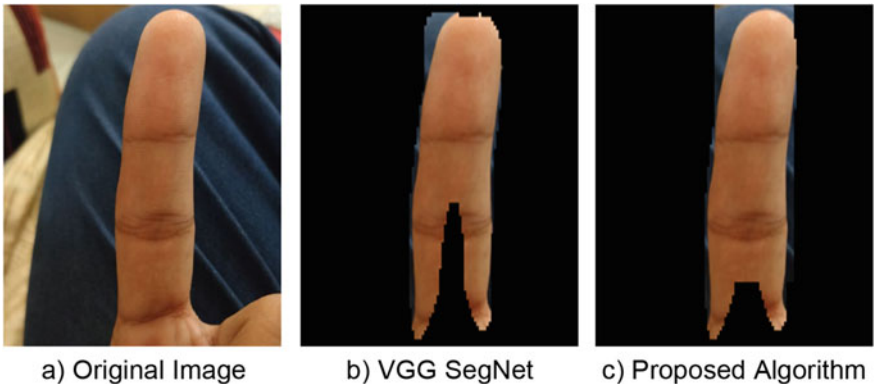

Fig. 2.11 Significance of 32 × 32 smoothening over VGG SegNet architecture

reason for higher BSA is the tight bound over the located finger-selfie obtained by the trained VGG SegNet. A drawback of a tight bound over the located finger-selfie is that few foreground finger regions are termed as background while most background regions are predicted as background. Thus, for VGG SegNet, BSA is higher than FSA due to erroneous classification of foreground pixels on the boundary of the located finger-selfie.

As observed by the segmentation performance of VGG SegNet in Table 2.4, FSA remains lower due to misclassification of foreground pixels located on the boundary of the located finger-selfie. Loosening the predicted boundary by VGG SegNet will increase foreground pixels, in turn increasing FSA. Thus, a 32 × 32 block-wise smoothening layer is added in the VGG SegNet architecture and it aids in increasing the FSA from 66.75 to 71.22%. While there is a trade-off for reduced SA and BSA by 1.04 and 1.98%, respectively, the distinctive ridge-valley details present in foreground region in finger-selfies are not compromised. An illustration of the effect of smoothening over VGG SegNet is shown in Fig. 2.11.

2.4.4.2 Comparison of VGG SegNet with FCN 8

Similar to VGG SegNet, a FCN 8 architecture is also fine-tuned. Inferring from the positive effect of 32 × 32 block-wise smoothening on FSA, FCN 8 architecture also includes a 32 × 32 block-wise smoothening. The FCN 8 trains a fully convolutional encoder–decoder network, and it uses an AdaDelta optimizer and a cross-entropy loss function.

Table 2.5 shows a comparison of segmentation performance of FCN-8-based segmentation with VGG SegNet-based segmentation algorithm. However, with highest FSA and overall segmentation accuracy, the VGG SegNet + block-wise smoothening model outperforms under both the scenarios. One of the major reasons for better performance of VGG SegNet-based approach is the lesser number of trainable

Table 2.5 Comparison of segmentation performance of the finger-selfie segmentation framework with FCN 8

Algorithm	Segmentation metric	Finger			
		All together (%)	Index (%)	Middle (%)	Multiple (%)
FCN 8	SA	88.55	89.45	90.19	86.01
	BSA	93.92	94.22	94.09	93.45
	FSA	61.46	60.11	63.66	60.62
FCN 8 + 32 × 32 block-wise smoothening	SA	87.56	88.37	89.16	85.16
	BSA	92.04	92.41	92.43	91.27
	FSA	65.81	64.19	67.97	65.26
VGG SegNet + 32 × 32 block-wise smoothening	SA	89.04	89.89	90.62	86.61
	BSA	92.71	93.16	93.06	91.91
	FSA	71.22	70.28	74.49	68.90

parameters [33]. Using the max-pooling indices from respective encoding layers, the decoder in VGG SegNet performs sparse upsampling. This procedure reduces computation time as well as increases generalizability of the model. On the contrary, FCN 8 learns parameters for upsampling too. Hence, despite data augmentation, the training data may not be enough to train additional parameters, which justifies VGG SegNet outperforming FCN 8.

2.4.4.3 Comparison with Skin Color-Based Segmentation

Inspired from existing studies [12, 16, 18, 23], the VGG SegNet + 32 × 32 block-wise smoothening model is also compared with various skin color-based segmentation algorithms. The results are presented in Fig. 2.12. The foremost comparison is performed with a thresholding color channel-based skin color segmentation algorithm [34, 35]. The finger-selfie image, available in RGB color space, is converted to HSV and YCbCr color space. The information in Hue, Cb, and Cr color space is used to find probable skin color regions using pre-defined thresholds. While the VGG SegNet + 32×32 block-wise smoothening method provides FSA of 71.22%, skin color-based segmentation provides FSA of 58%. Segmentation algorithm proposed by Sankaran et al. [12] also fails to perform well. Due to image augmentation by varying intensities, our fine-tuned model becomes robust toward illumination variations and flash usage in finger-selfies. However, because of too bright or too dull skin regions in certain cases, the standard skin color algorithms fail due to fixed thresholds.

Additionally, a comparison is shown of skin color segmentation with a deep architecture. Firstly, the salient region is cropped out using skin color-based segmentation. The salient region is fed as input to the architecture: VGG SegNet + 32 × 32

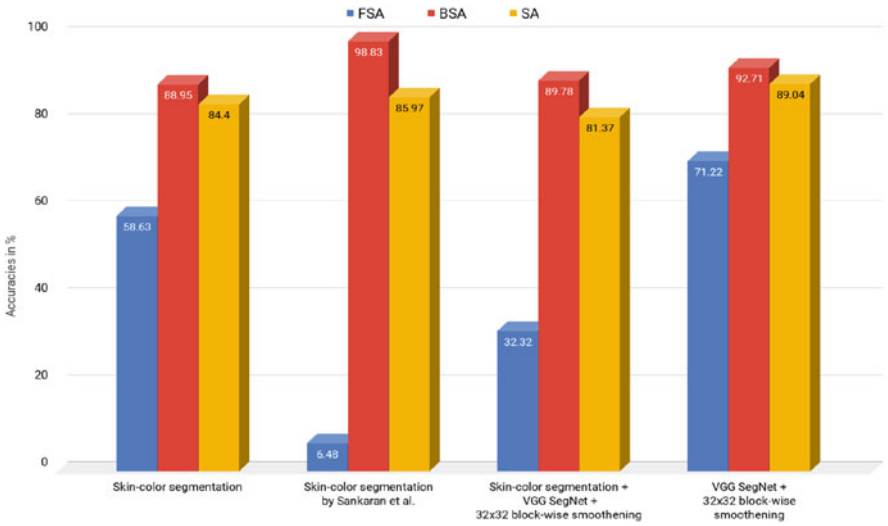

Fig. 2.12 Comparison of segmentation accuracies obtained with the skin color-based techniques and the VGG SegNet with block-wise smoothening algorithm

smoothening. However, both SA and FSA are reduced. The results are shown in Fig. 2.12. These results indicate that in an unconstrained scenario, skin color-based segmentation is likely to fail.

2.5 Finger-Selfie Recognition

In 2013, Li et al. [22] highlighted that minutiae-based techniques for feature extraction and matching would fail for finger-selfies. Sankaran et al. [12] showcased a similar inference, highlighting that minutiae-based techniques fail for semi-constrained scenarios. Hence, the authors used ScatNet for their experiments. While ScatNet worked for the semi-constrained scenario, the representation would fail to encode discriminatory information under deformations and rotational variations present in the UNFIT database. As a result, we too utilized two non-minutiae-based algorithms for feature extraction, namely CompCode and ResNet50. The details are mentioned in the subsection below.

2.5.1 Feature Representations

Non-Deep learning: Competitive Coding (CompCode) [13, 36] is a popular non-minutiae-based feature representation, commonly deployed for fingerprint and palm-

print recognition. Quite recently, CompCode and its variant were exploited for utilizing ridge-valley details present in palmprints for person recognition [36]. With ridge-valley pattern forming a unique structure, filters that encode orientation information can provide an efficient feature representation. CompCode features are extracted by convolution of the real part of the Gabor filter G_r over the image I. The Gabor filters G_r have J different orientations, each of which varies from previous by $\frac{\pi}{J}$. Along with orientation variations, Gabor filters differ in frequency W as well. Hence, the total number of filters convolved to obtain the feature representation are $J \times W$. The response of the filter, convolved over the segmented finger-selfie I, is given as:

$$R = I(x, y) * \psi_R(x, y, \omega_i, \theta_j) \tag{2.4}$$

Here, ψ_R is the real part of the Gabor filter ψ, while ω_i and θ_j are frequency and orientation of the Gabor filter. Note that the segmented output is upscaled to a fixed size of 400×400 before applying Gabor filters to obtain the representation.

Deep learning-based approach: The segmented finger-selfie is served as input to the ResNet50 architecture [14]. ResNets have shown their application to general object recognition with deeper networks. To counter the effect of vanishing gradient and overfitting, ResNets have shortcut connections among different convolutional layers. Intuitively, along with the feedforward mapping $F(x)$ from the previous layer C_l, the input to the next convolution layer C_{l+1} also includes an identity mapping x from some previous layer C_{l-k}. Hence, the input to convolutional layer C_{l+1} can be written as:

$$F(x, \{W_i\}) + x \tag{2.5}$$

where W_i signifies transformation through multiple convolutional layers. In the ResNet50 architecture, the function $F(x)$ involves two stacked convolutional layers. This implies that the input x is taken from the activated output of layer C_{l-2}, and $W_i x$ is a transformation of x over two convolutional layers.

The segmented RGB image is provided to the network at a fixed size of 224 × 224. In our experiments, the ResNet50 architecture is initialized using the weights of the model trained on the ImageNet database. With the Softmax classification layer removed, the network provides a feature vector of dimension 2048 × 1, which is treated as the feature representation of the finger-selfie. The intermediate layers of

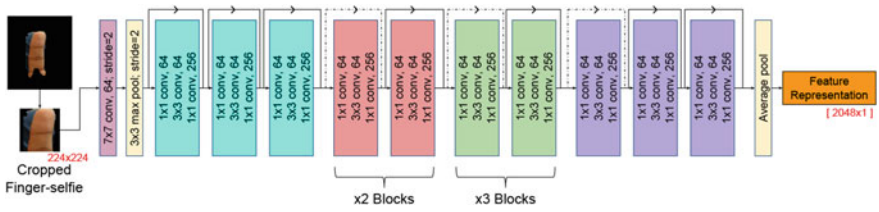

Fig. 2.13 Procedure to obtain feature representation using ResNet50 architecture

ResNet50 look for different shapes and strokes. Hence, the final feature representation encodes curves, vertical and horizontal lines, and other shapes, which are equivalent to ridge orientations, finger shape, and phalanx lines. The procedure to obtain the feature representation is illustrated in Fig. 2.13.

2.5.2 *Finger-Selfie Recognition Performance*

After extracting features from finger-selfie images, the next step is to match the query feature templates with the gallery templates. The CompCode features are matched with gallery templates using Hamming distance to obtain a distance score. Similarly, representation obtained from ResNet50 architecture is matched with gallery templates using cosine similarity.

On the testing set of 57 subjects, receiver operating characteristic (ROC) curve is used to report the verification performance. The ROC curve is shown in Fig. 2.14. Table 2.6 shows the confusion matrix when feature representation from CompCode and ResNet50 are matched using Hamming distance and cosine similarity, respectively. In spite of the potency of CompCode for palmprint and fingerprint recognition, we observe an EER of 41.41% for finger-selfie matching. On the other hand, the cosine similarity of ResNet50-based representation yields a better performance with EER as 35.32%.

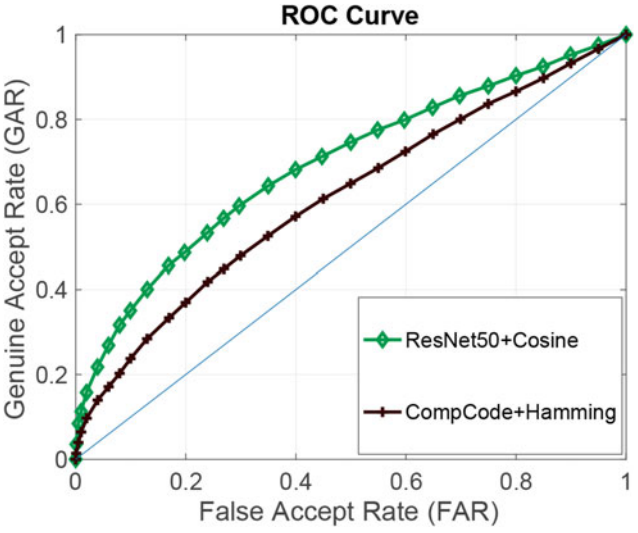

Fig. 2.14 Receiver operating characteristic (ROC) curve for the VGG SegNet + 32 × 32 segmentation pipeline. Representation from ResNet50 architecture is matched using cosine similarity, and CompCode features are matched using Hamming distance metric on the test set of UNFIT database

Table 2.6 Confusion matrix when feature representation from CompCode and ResNet50 are matched using Hamming distance and cosine similarity, respectively. From a total of 731,025 pairs (855 probe representations matched with 855 gallery representations), there are 4275 genuine and 726,750 imposter pairs. Values are reported at 10% FAR

		Prediction			
		CompCode+Hamming		ResNet50+Cosine	
		Match	Non-Match	Match	Non-Match
Ground truth	Match	1016	3259	1517	2758
	Non-match	72,480	654,270	73,335	653,415

The finger-selfie dataset, namely the UNFIT database, has numerous variations. The variations occur due to an unconstrained environment. The ResNet50 model is pre-trained on ImageNet database, where objects are of different shapes and sizes. These learned weights can handle variations in finger-selfies pertaining to scale and orientation of finger-selfie. Also, ResNet50 feature representation for segmented finger-selfies is matched using cosine similarity. Since cosine similarity is an angular similarity of two vectors, variations introduced in the magnitude of representations due to illumination variations do not effect cosine similarity. Hence, the recognition model becomes robust toward illumination variations. Thus, the overall performance of ResNet50 + Cosine similarity is better than CompCode + Hamming distance-based recognition.

While such results are motivating that deep architectures have a better potential for finger-selfie recognition, there is still a long way to go for recognition of finger-selfies in an unconstrained scenario. With the proposed UNFIT database, we expect that the research community will be driven toward building better segmentation, enhancement, quality assessment, and feature representation modules for finger-selfie-based recognition.

2.6 Conclusion

This chapter presents a review of existing research on finger-selfies and later introduces finger-selfie in an unconstrained environment. The proposed UNconstrained FIngerphoTo (UNFIT) database incorporates various challenges such as rotation, translation, orientation, position, scale, multiple fingers, illumination, background, and resolution which arise due to the differing environments in which the finger-selfies are acquired. This database includes the manual annotations and experimental protocol, using which segmentation and verification results are benchmarked. A VGG SegNet-based segmentation approach is presented along with baseline results, followed by matching algorithms using CompCode and ResNet50 representations. We assert that the proposed database can take forward the research in this domain and the segmentation pipeline can segment and perform authentication using finger-

selfies despite the challenges posed in the database. Future work can include quality assessment for detection of poor-quality finger-selfies and use of minutiae in conjunction with deep learning features for improved recognition performance.

Acknowledgements The authors would like to thank all the volunteers for their enthusiastic participation in the collection of the UNFIT database. Aakarsh Malhotra is partly supported through the Visvesvaraya Ph.D. Scheme of Ministry of Electronics & Information Technology, Government of India, being implemented by Digital India Corporation. Mayank Vatsa and Richa Singh are partly supported by the Infosys Center for Artificial Intelligence, IIIT Delhi, India. Mayank Vatsa is also supported by Swarnajayanti Fellowship from Government of India.

References

1. Taekyoung K, Jin H (2015) Analysis and improvement of a PIN-entry method resilient to shoulder-surfing and recording attacks. IEEE Trans Inf Forensics Secur 10(2):278–292
2. Staff M, Fleishman G (2018) iPhone X. https://www.macworld.com/article/3225406/iphone-ipad/face-id-iphone-x-faq.html. Accessed on 11 Feb 2018
3. Bajaj K, Bhagat HR (2018) Your phone's fingerprint scanner can do much more than just unlock your phone. Here's how . https://economictimes.indiatimes.com/magazines/panache/your-phones-fingerprint-scanner-can-do-much-more-than-just-unlock-your-phone-heres-how/articleshow/57766012.cms. Accessed on 02 May 2018
4. Tim Ahonen. Phone book 2012: Statistics and facts on the mobile phone industry, 2012. http://www.tomiahonen.com/ebook/phonebook.html. Accessed on 02 May 2018
5. Richter F (2018) Smartphones cause photography boom. https://www.statista.com/chart/10913/number-of-photos-taken-worldwide/. Accessed on 20 June 2018
6. Kumar A, Zhou Y (2011) Contactless fingerprint identification using level zero features. In: IEEE conference on computer vision and pattern recognition workshops, pp 114–119
7. Wood C (2018) WhatsApp photo drug dealer caught by 'groundbreaking' work. http://www.bbc.com/news/uk-wales-43711477. Accessed on 25 May 2018
8. Hern A (2018) Hacker fakes German minister's fingerprints using photos of her hands. https://www.theguardian.com/technology/2014/dec/30/hacker-fakes-german-ministers-fingerprints-using-photos-of-her-hands. Accessed on 25 May 2018
9. Chopra S, Malhotra A, Vatsa M, Singh R (2018) Unconstrained fingerphoto database. In: IEEE conference on computer vision and pattern recognition workshops, pp 517–525
10. Badrinarayanan V, Kendall A, Cipolla R (2015) SegNet: a deep convolutional encoder-decoder architecture for image segmentation. In arXiv:1511.00561v3
11. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: IEEE conference computer vis pattern tecognit, pp 3431–3440
12. Sankaran A, Malhotra A, Mittal A, Vatsa M, Singh R (2015) On smartphone camera based fingerphoto authentication. In: IEEE international conference on biometrics theory, applications and systems, pp 1–7
13. Kong AW-K, Zhang D (2004) Competitive coding scheme for palmprint verification. In: IAPR international conference on pattern recognition vol 1, pp 520–523
14. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: IEEE conference on computer vision and pattern recognition, pp 770–778
15. Song Y, Lee C, Kim J (2004) A new scheme for touchless fingerprint recognition system. In: IEEE international symposium on intelligent signal processing and communication systems, pp 524–527
16. Lee C, Lee S, Kim J, Kim S-J (2006) Preprocessing of a fingerprint image captured with a mobile camera. In: IAPR international conference on biometrics. Springer, pp 348–355

17. Lee D, Choi K, Choi H, Kim J (2008) Recognizable-image selection for fingerprint recognition with a mobile-device camera. IEEE Trans Syst, Man, Cybern, Part B (Cybern) 38(1):233–243
18. Piuri V, Scotti F (2008) Fingerprint biometrics via low-cost sensors and webcams. In: IEEE international conference on biometrics: theory, applications and systems, pp 1–6
19. Hiew BY, Teoh ABJ, Yin OS (2010) A secure digital camera based fingerprint verification system. J Vis Commun Image Represent 21(3):219–231
20. Derawi MO, Yang B, Busch C (2011) Fingerprint recognition with embedded cameras on mobile phones. In: International conference on security and privacy in mobile information and communication systems. Springer, pp 136–147
21. Yang B, Li G, Busch C (2013) Qualifying fingerprint samples captured by smartphone cameras. In: IEEE international conference on image processing, pp 4161–4165
22. Li G, Yang B, Olsen MA, Busch C (2013) Quality assessment for fingerprints collected by smartphone cameras. In: IEEE conference on computer vision and pattern recognition workshops, pp 146–153
23. Raghavendra R, Busch C, Yang B (2013) Scaling-robust fingerprint verification with smartphone camera in real-life scenarios. In: IEEE international conference on biometrics: theory, applications and systems, pp 1–8
24. Stein C, Bouatou V, Busch C (2013) Video-based fingerphoto recognition with anti-spoofing techniques with smartphone cameras. In: IEEE international conference of the biometrics special interest group, pp 1–12
25. Tiwari K, Gupta P (2015) A touch-less fingerphoto recognition system for mobile hand-held devices. In: IAPR international conference on biometrics, pp 151–156
26. Taneja A, Tayal A, Malhotra A, Sankaran A, Vatsa M, Singh R (2016) Fingerphoto spoofing in mobile devices: a preliminary study. In: IEEE international conference on biometrics theory, applications and systems pp 1–7
27. Lin C, Kumar A (2018) Matching contactless and contact-based conventional fingerprint images for biometrics identification. IEEE Trans Image Process 27(4):2008–2021
28. Malhotra A, Sankaran A, Vatsa M, Singh R (2018) Learning representations for unconstrained fingerprint recognition. Deep Learn Biom 197–226
29. Malhotra A, Sankaran A, Mittal A, Vatsa M, Singh R (2017) Fingerphoto authentication using smartphone camera captured under varying environmental conditions. Human Recognition in Unconstrained Environments: Using Computer Vision, Pattern Recognition and Machine Learning Methods for Biometrics, pp 119–144
30. Dollár P (2018) Piotr's computer vision matlab tooflbox (PMT). https://github.com/pdollar/toolbox. Accessed on 22 Feb 2018
31. Girshick R (2015) Fast R-CNN. In: IEEE international conference on computer vision, pp 1440–1448
32. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask R-CNN. In: IEEE international conference on computer vision, pp 2980–2988
33. Semantic Segmentation Models for Autonomous Vehicles. https://blog.playment.io/semantic-segmentation-models-autonomous-vehicles/. Accessed on 26 June 2018
34. Kakumanu P, Makrogiannis S, Bourbakis N (2007) A survey of skin-color modeling and detection methods. Pattern Recognit 40(3):1106-1122
35. Kolkur S, Kalbande D, Shimpi P, Bapat CP, Jatakia J (2016) Human skin detection using RGB, HSV and YCbCr color models. In: International Conference on Communication and Signal Processing
36. Zheng Q, Kumar A, Pan G (2016) Suspecting less and doing better: new insights on palmprint identification for faster and more accurate matching. IEEE Trans Inf Forensics Secur 11(3):633–641