Research Article

Fingerprint Generation for DNN Training: A Case Study in Fingerprint Classification

Emre Irtem, Nesli Erdogmus

***Abstract*—Large annotated datasets are crucial for training state-of-the-art deep learning systems. However, the availability of publicly accessible fingerprint data significantly lags behind that of image datasets or text corpora, which are extensively utilized for tasks such as image understanding and natural language processing. The challenges associated with the collection and distribution of fingerprint data make synthetic data generation a viable alternative. Nonetheless, existing research primarily focuses on the large-scale evaluation of fingerprint search systems rather than examining the usability of generated fingerprint images for training purposes. This study employs a model-based method to generate synthetic fingerprints and evaluates their effectiveness in training deep neural networks for fingerprint classification. The findings indicate that augmenting the training set with synthetic fingerprint impression images enhances performance comparably to augmenting it with real fingerprint images.**

***Index Terms*—fingerprint image generation, synthetic training data, deep learning, fingerprint classification.**

I. Introduction

AUTOMATION is indispensable for fingerprint identification systems, as thousands of search requests are submitted daily to fingerprint databases, which can reach colossal sizes. Over the past several decades, researchers have developed various algorithms and systems to achieve automated fingerprint analysis and comparison. A subset of this research has concentrated on fingerprint generation, driven primarily by the necessity of large-scale datasets to evaluate the proposed algorithms under realistic conditions, given the limited availability of publicly accessible fingerprint images. This scarcity is due to fundamental challenges in fingerprint data collection, including the need for expert personnel, specialized equipment, and concerns regarding privacy. Furthermore, many commonly used datasets have been discontinued owing to recent legal restrictions aimed at protecting the privacy of biometric data [1].

Today, large-scale datasets are essential not only for performance evaluations but also for training deep neural networks. However, the use of synthetic fingerprints as training data has not been extensively investigated. To the best of our knowledge, only one recent study [2] has evaluated the impact of data augmentation through fingerprint synthesis, specifically in the context of latent fingerprint reconstruction.

In this study, we aim to analyze how synthetic fingerprint images can enhance the accuracy of deep neural networks in fingerprint classification. Fingerprint classification facilitates fingerprint matching by filtering the database based on the estimated class, thereby reducing the number of candidates and speeding up the search process. This task is chosen for our study because class information is one of the four types of ground truth provided by model-based synthetic data generators, the others being identity, frequency maps, and orientation maps [3], [4]. In contrast, learning-based generators generally do not supply these types of information and are primarily used to enlarge the gallery size for performance evaluations [5], [6]. Considering these factors, we employ the model-based SFinGe method to generate synthetic fingerprints for our approach [3].

Far as we know, the SFinGe method is implemented in two fingerprint generation tools: SFinGe [7] and Anguli [8]. The SFinGe tool is available in both demo and full versions. The demo version permits the generation of a single impression per finger and restricts the total number of generations. The full version allows for multiple impressions per finger, but the number of fingerprints that can be generated is still limited based on the purchased edition. For both versions, the generated synthetic images cannot be distributed or made publicly available on the Internet. Conversely, the Anguli tool is freely available and can be used to generate and distribute any number of synthetic fingerprint images, including multiple impressions of the same finger. However, with Anguli, the extent of degradation in the impressions is not fully controllable. Additionally, for both tools, it is not possible to add extra control parameters for existing degradation types or to introduce entirely new degradation types.

To analyze different degradation types and levels, generate multiple impressions of the same fingerprint, and distribute the synthetic fingerprint dataset for further use and reproducibility, we have implemented our own model-based tool to generate synthetic fingerprint images. The fundamental steps are inspired by the SFinGe method [3]. However, many steps have been modified in order to obtain visually more realistic synthetic fingerprints and some intermediate methods are developed from scratch to generate diverse fingerprint impressions. Unlike the approach in [3],

- A normalization step is added to the ridge pattern generation to reduce the number of required iterations.,
- ridge discontinuities and irregularities within the fingerprint are modeled using coherent noise,

Emre İrtem is with the Department of Computer Engineering, Izmir Institute of Technology, İzmir, 35430 TURKEY e-mail: emreirtem@iyte.edu.tr

Nesli Erdoğmuş is with the Department of Computer Engineering, Izmir Institute of Technology, İzmir, 35430 TURKEY e-mail: neslierdogmus@iyte.edu.tr

Manuscript received Jul. 19, 2024; accepted Jan. 23, 2025.
DOI: 10.17694/bajece.1519228

- background text and textures are modeled.

The primary contribution of this study lies not in the generation method itself but in the utilization of the generated fingerprint impression images for training set augmentation and the analysis of their impact under different conditions. While it may seem ideal to have fingerprint experts inspect and evaluate the realism of the generated impressions, we did not have access to such expertise. However, this is not the primary focus of our study. Our main objective is to enhance deep learning-based fingerprint processing tasks and, in doing so, demonstrate the fidelity of the generated data. To this end, fingerprint classification is selected as the test case because class is one of the few attributes that can be controlled in the adopted generation method. Furthermore, classification is a fundamental stage in many fingerprint recognition systems, as it can reduce the number of required comparisons and decrease the response time of matching algorithms.

Finally, a dataset of synthetically generated fingerprint impression images is made publicly available to ensure the reproducibility of the results and facilitate further analyses (GitHub repository to be disclosed). The dataset comprises 12,500 samples, with 2,500 images for each of the five NIST standard classes: Right Loop, Left Loop, Whorl, Arch, and Tented Arch (Figure 8b). Each image is accompanied by frequency and orientation maps, minutiae locations, and identity information. Additionally, the experiment codes are also open-sourced.

II. Related Work

Research on the impact of training set augmentation using synthetic fingerprint images is relatively sparse. To our knowledge, only one publication [2] has conducted an analysis along these lines. In that study, latent fingerprints are synthesized for training a reconstruction network aimed at transforming low-quality fingerprint images into ridge images through pixel-level binary classification. The enhancement in reconstruction quality was correlated with improved matching performance, as demonstrated through quantitative assessment, highlighting the efficacy of the proposed training set augmentation. In contrast, other studies referenced in this section primarily focus on generating fingerprints to expand the gallery used for testing purposes.

Each technique possesses distinct advantages and disadvantages. Model-based techniques involve decomposing the generation process into discrete sub-tasks, necessitating significant engineering skill and domain expertise. For instance, in a model-based fingerprint generation pipeline, (I) a fingerprint class is selected, (II) corresponding singular point coordinates and ridge orientations are assigned, (III) ridge frequency maps are generated, (IV) ridge patterns are synthesized using filtering techniques, and (V) fingerprint impressions are created by applying degradation processes to the synthesized pattern. However, numerous assumptions are made at each step; for example, many model-based methods assume independence between ridge orientations and minutiae locations, which may lead to unrealistic minutiae configurations [12]. Nevertheless, model-based techniques offer advantages such as not requiring training data and inherently providing various ground truth labels on generated images, which are crucial for training purposes.

In contrast, learning-based methods necessitate a substantial number of training samples to effectively learn fingerprint generation. Although they produce ”black-box” generators, limiting the direct extraction of detailed fingerprint metadata, they generally yield statistically more realistic samples compared to model-based techniques.

A. Model-based methods

In [9], a model-based technique for fingerprint generation is proposed. The process begins with the generation of a master fingerprint, which serves as the basis for producing multiple impressions. Initially, a fingerprint shape is defined, and a ridge orientation map is calculated using a mathematical model based on zero-pole patterns [10], loop, and delta coordinates. Subsequently, a ridge frequency map is generated by visually inferring from multiple real fingerprints, followed by the synthesis of a ridge pattern image using iterative Gabor filtering [11] applied to a randomly initialized image. The resulting image is then thresholded to obtain a binary master fingerprint image. To create different impressions, three types of synthetic distortions are applied to the master fingerprint. First, morphological operations are employed to simulate wet or dry skin conditions. Second, non-linear transformations are used to mimic skin elasticity. Lastly, ridge discontinuities and irregularities are modeled by adding various white blobs of different shapes and sizes to the fingerprint image.

In [3], the SFinGe approach is introduced as an advancement over its previous version [9], aiming to generate more realistic fingerprint impressions. SFinGe includes additional distortions such as random rotation and translation of the ridge pattern to simulate different finger placements in an image. It also incorporates a realistic background generated using a mathematical model based on the Karhunen-Loeve transform, which is superimposed with the fingerprint impression. Moreover, instead of uniform noise, coherent noise [20] is proposed in [21] to enhance variability in impression generation.

In [12], a non-parametric approach models ridge features of real fingerprints, which are then sampled to create synthetic features used as inputs to the SFinGe tool [3]. The realism of generated fingerprints is validated by comparing their feature densities with those of real fingerprints.

Addressing unrealistic minutiae configurations in synthetic images generated by SFinGe, [16] employs orientation maps extracted from real fingerprints to create master fingerprints. Realness tests are applied to minutiae maps to filter out non-realistic configurations, resulting in a more realistic database in terms of minutiae configurations.

In [14], a statistical model is proposed to obtain realistic fingerprint features, including singular point locations [22] and minutiae points, trained on publicly available real-fingerprint datasets. Unlike SFinGe, the ridge generation phase incorporates AF-FM filters [23].

TABLE I: Existing model and learning-based methods for fingerprint generation

Model based methods

Method	Orientation model	Ridge generation	Minutiae locations
R. Cappelli et al. [9]	Zero-Pole [10]	Gabor-Filter [11]	Random
SFinGe [3]	Zero-Pole [10]	Gabor-Filter [11]	Random
P.Johnson et al. [12]	Zero-Pole [10]	Gabor-Filter [11]	Statistically realistic [13]
Q.Zhao et al. [14]	Zero-Pole [10]	AF-FM [15]	Statistically realistic [13]
C.Imdahl et al. [16]	Zero-Pole [10]	Gabor-Filter [11]	Elimination using a realness test [17]

Learning based methods

Method	Learning model
P.Bontrager et al. [18]	Wasserstein GAN (WGAN)
K.Cao and A.K.Jain [5]	IWGAN and Autoencoder
M.Attia et al. [19]	Variational Autoencoder
V.Mistry et al. [6]	IWGAN and Autoencoder with Identity Loss

B. Learning-based methods

In [18], a method is proposed to generate synthetic fingerprints using deep neural networks aimed at attacking fingerprint-matching systems. The approach involves training a Wasserstein Generative Adversarial Network (WGAN) and evolving latent variables of the generator network using The Covariance Matrix Adaption Evolutionary Strategy. This method searches for a fingerprint that matches a large number of other fingerprints.

In [5], fingerprints are generated using Improved WGAN for which a Convolutional Autoencoder with a 512-dimensional latent vector is trained and used to initialize the generator of WGAN. This approach demonstrates improvements in fingerprint quality and diversity.

In [19], fingerprints are generated using Variational Autoencoders (VAE). The method ensures that the generated samples match the distribution of real fingerprint datasets via latent vectors. During training, an image x in the input image space X is mapped to a latent vector z in the latent vector space Z by an encoder. The training maximizes $P_Q(x|z)$, where the encoder learns the mapping of latent variable vectors (μ, σ^2). These variables are used to sample the latent vector z, which is then used by the decoder to generate an output image. Synthetic fingerprints are generated by feeding randomly generated 32-dimensional vectors into the decoder.

In [6], a method that is capable of generating more realistic fingerprints in terms of minutiae count, direction and spatial distribution is proposed. Following initialization with I-WGAN as described in [5], an identity loss is incorporated into the generator's objective during training to ensure the generation of unique identities.

III. Fingerprint Generation

A model-based generation method is adopted for the following reasons:

- **Data scarcity:** There is a limited number of publicly available fingerprint datasets, which poses a challenge for training data-intensive deep generative networks.
- **Meta-data availability:** It is crucial to obtain meta-data for the generated fingerprints. This study aims to produce synthetic data suitable for training deep neural networks for various tasks such as minutiae extraction, orientation field estimation, ridge frequency estimation, and fingerprint classification. Providing labels is essential, and model-based techniques are well-suited for this purpose.
- **Multiple impressions:** Training fingerprint matching systems requires multiple impressions of the same finger. The model-based method allows the creation of different impressions from the same master fingerprint.

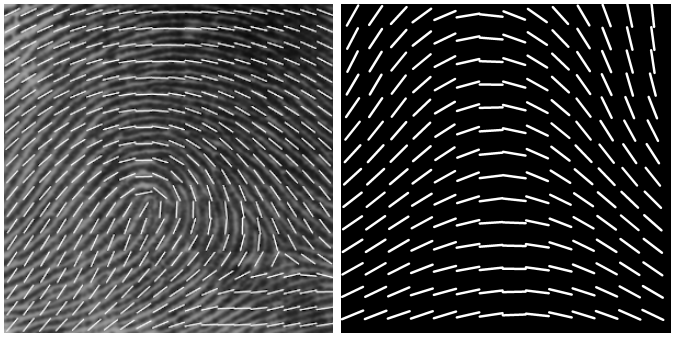
A. Master Fingerprint Generation

A master fingerprint is an image that represents the ideal impression of a ridge pattern from a "synthetic finger" [9]. It is devoid of any noise or external factors that cause variations in fingerprint images captured from the same finger. While the master fingerprint of an actual finger is not accessible, fingerprint enhancement methods aim to obtain a close approximation.

This study adopts an approach similar to SFinge [3] to generate master fingerprints. After constructing the orientation and frequency maps, a ridge pattern is developed by iteratively applying pixel-specific Gabor filters to a randomly initialized image. Unlike SFinge, this method considers variations in fingerprint area during acquisition and incorporates these variations into the impression generation process.

Orientation images

Orientation images designate principal ridge directions for each pixel on the master fingerprints to be generated. These images must adhere to strict smoothness constraints, as closely positioned ridges have similar orientations, and they must conform to a limited number of ridge pattern types. To achieve this, the zero-pole model proposed by Vizcaya and Gerhardt [24] is utilized, as employed in [3]. This model is based on the work of Sherlock and Monro [10], which computes the orientation (Θ) image in a complex plane using a complex rational function that incorporates delta (S_{d_i}) and loop (S_{l_j}) locations. (Eqn. 1)



(a) (b)

Fig. 1: Orientation maps generated by (a) zero-pole model output superimposed with an actual fingerprint image and (b) sinusoidal model output

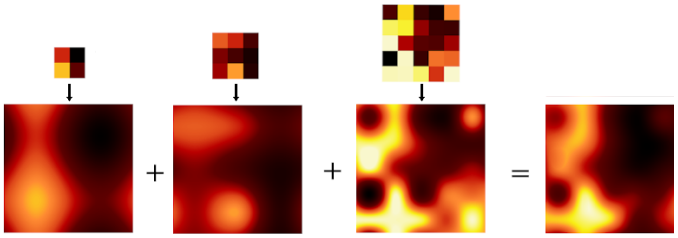


Fig. 2: Frequency image generation using coherent noise

$$\Theta = \frac{1}{2}\left[\sum_i^{n_d} \arg(z - S_{d_i})) - \sum_j^{n_l} \arg(z - S_{l_j}))\right] \quad (1)$$

However, this model does not fit very well on real fingerprints and suffers from low variability. In [24], a piece-wise linear correction function (g) that corrects the phase angle of each singularity is used. (Eqn. 2)

$$\Theta = \frac{1}{2}\left[\sum_i^{n_d} g_{d_i}(\arg(z - S_{d_i})) - \sum_j^{n_l} g_{l_i}(\arg(z - S_{l_j}))\right] \quad (2)$$

This model is capable of generating accurate orientation fields using singular point locations for four major pattern types: left loop, right loop, tented arch, and whorl classes. However, since the arch class lacks singular points, its ridge flow is simulated using a sinusoidal function. For each pixel, orientation is calculated using Eqn. 3.

$$\Theta = \beta \sin(f(x)) \quad (3)$$

The function $f(x)$ defines a uniform mapping between the x coordinates of image pixels and the interval $[\frac{\pi}{2}, -\frac{\pi}{2}]$. The parameter β adjusts the amplitude of this mapping. To enhance the variability in generated orientations, similar adjustments are made as in Vizcaya and Gerhardt's method, where arch center coordinates are used instead of loops and deltas. Sample orientation images generated using the zero-pole model [24] and the sine-based approach are illustrated in Figure 1.

Frequency images

Ridge frequency determines the density of ridges per unit length along the perpendicular direction on a fingerprint. It varies non-uniformly across the whole fingerprint area and is influenced by many factors such as distances to the existing singular points or the fingerprint borders. Additionally, these frequencies often exhibit smooth transitions between neighboring local regions.

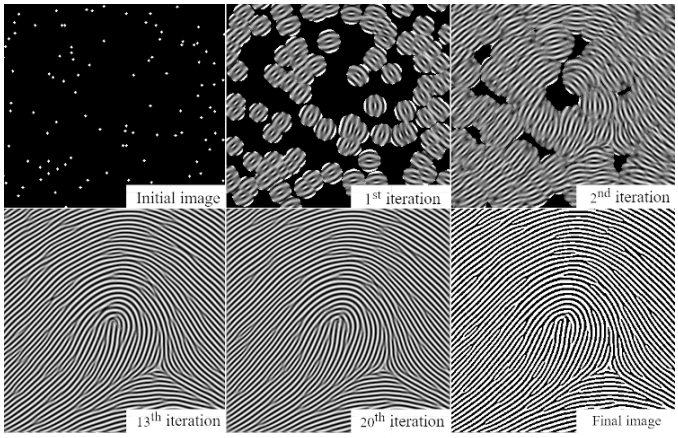


Fig. 3: A ridge pattern generated at different iterations and after thresholding

For SFinGe, frequency images are generated according to some observations on real fingerprints, such as the decrease in frequency above the northernmost loop and below the southernmost delta [25]. To increase variations in the generated fingerprint images, a less constraining approach is followed in this study, and ridge frequencies are synthesized as coherent noise maps. Three random noise images are generated at different resolutions (2x2, 3x3, and 5x5) and then scaled to the same size (400x400) using bi-cubic interpolation. The final frequency image is obtained by adding them and normalizing the outcome (Figure 2).

Ridge Patterns

Using the orientation and frequency values of each pixel, ridge patterns are iteratively formed by convolving a random initial image with Gabor filters. To optimize computation, the orientation (θ) values are discretized into 20 bins in the interval $0 \leq \theta \leq \pi$, while the frequency (f) values are discretized 100 bins in the interval $0.11 \leq f \leq 0.17$. The response image is normalized after each iteration to suppress high responses and enhance low responses. After several iterations, the resulting patterns are binarized using mean thresholding (Figure 3).

To obtain minutiae labels for the generated fingerprints, they are detected on the response images prior to thresholding. In these images, ridges and valleys manifest as extremities, while minutiae points occupy intermediate values. Leveraging this characteristic, the response image is normalized to the range $[-1, 1]$ and a probability map P, indicating the likelihood of each pixel being a minutia, is computed using Eqn. 4. Subsequently, the probability map undergoes thresholding and median filtering. The resulting image is then skeletonized, and minutiae locations are detected using maximum suppression algorithm.

$$P(X) = 1 - \left|\tanh(X)\right| \quad (4)$$

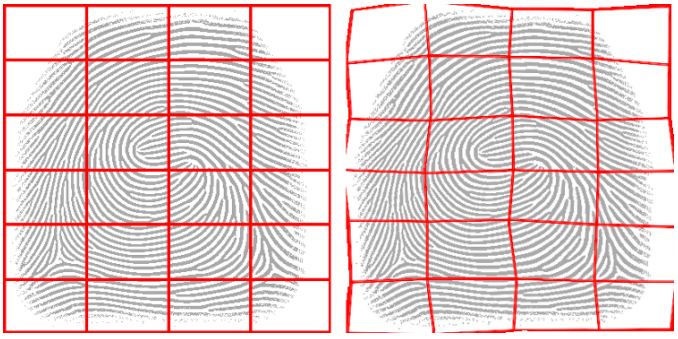


Fig. 4: Non-linear distortions simulated by Piecewise Affine Transformations

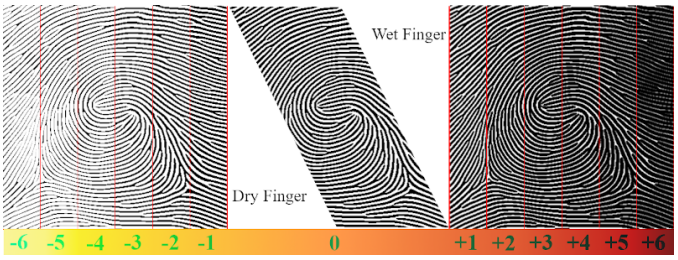


Fig. 5: Skin conditions simulated with various T

Creases

In real fingerprints, ridge lines can be interrupted by creases of varying lengths and thicknesses. These creases are modeled in this study as ellipses, where L represents the major-axis length and T denotes the minor-axis length. Each ellipse is applied to the master fingerprint image at randomly selected positions and orientations, following deformation through piece-wise affine transformations.

B. Impression Generation

Once a master fingerprint is obtained, multiple impressions can be generated by simulating various acquisition conditions. This study models impression variations introduced by six different real-world factors.

Contact area

The contact area of a fingertip on a surface varies depending on the pressure applied by the finger and the angle of the finger relative to that surface. In this study, contact areas are modeled as ellipse-like shapes using the approach proposed in [25].

Pose

Pose changes are induced by rotation (R), translation (T_x, T_y) and scaling (S) operations. The R parameter is randomly sampled from a range of $[-12°, 12°]$, T_x and T_y parameters are randomly sampled from a range of $[-30, +30]$ pixels and the S parameter is randomly generated from a range of $[0.85, 1.15]$.

Non-linear distortions

Impressions from the same finger may vary due to non-linear distortions caused by different placement and pressures of the finger against the surface. To model these distortions, a grid is overlaid on the fingerprint and deformed by applying Piecewise Affine Transformations by M pixel, where M is the movement parameter and randomly sampled for each grid segment from the interval $[-7, +7]$ (Figure 4).

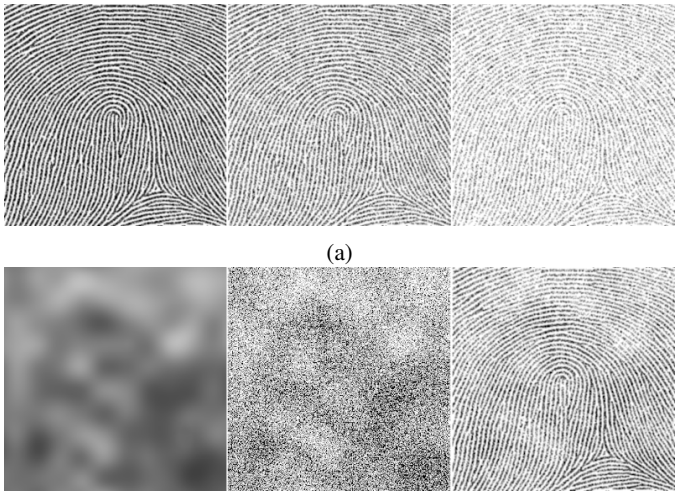


Fig. 6: Skin conditions simulated with respect to T (a) Impressions with uniform noise of probabilities of 0.25, 0.50 and 0.75 (b) Non-uniform probability map, ink density map, and the generated fingerprint

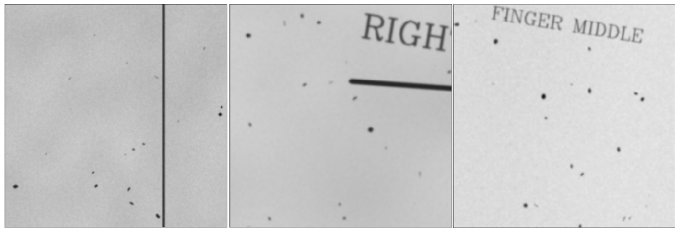


Fig. 7: Example generated backgrounds

Skin conditions

Ridge thickness in the fingerprints varies based on skin conditions; they are thinner under dry conditions and thicker with moisture. These variations are simulated using morphological operations applied to the master fingerprints. The ridge thickness is controlled by a parameter T, which determines the number of erosion or dilation operations applied to simulate dry and wet fingertips, respectively. For each impression, T is uniformly sampled from the interval $[-4, +4]$, where negative values correspond to erosion and positive values to dilation. Before applying these operations, the master fingerprints are scaled up by a factor of 4 before these operations to introduce more variability in ridge thickness (Figure 5).

Noise

Noise causes ridges discontinuities and local blurs in fingerprint images. To simulate these effects, a probability map is generated to determine whether a ridge pixel in the master fingerprint should remain or be eliminated. For instance, a uniform probability over the image creates a homogeneous ridge disappearance (Figure 6a). However, real fingerprint

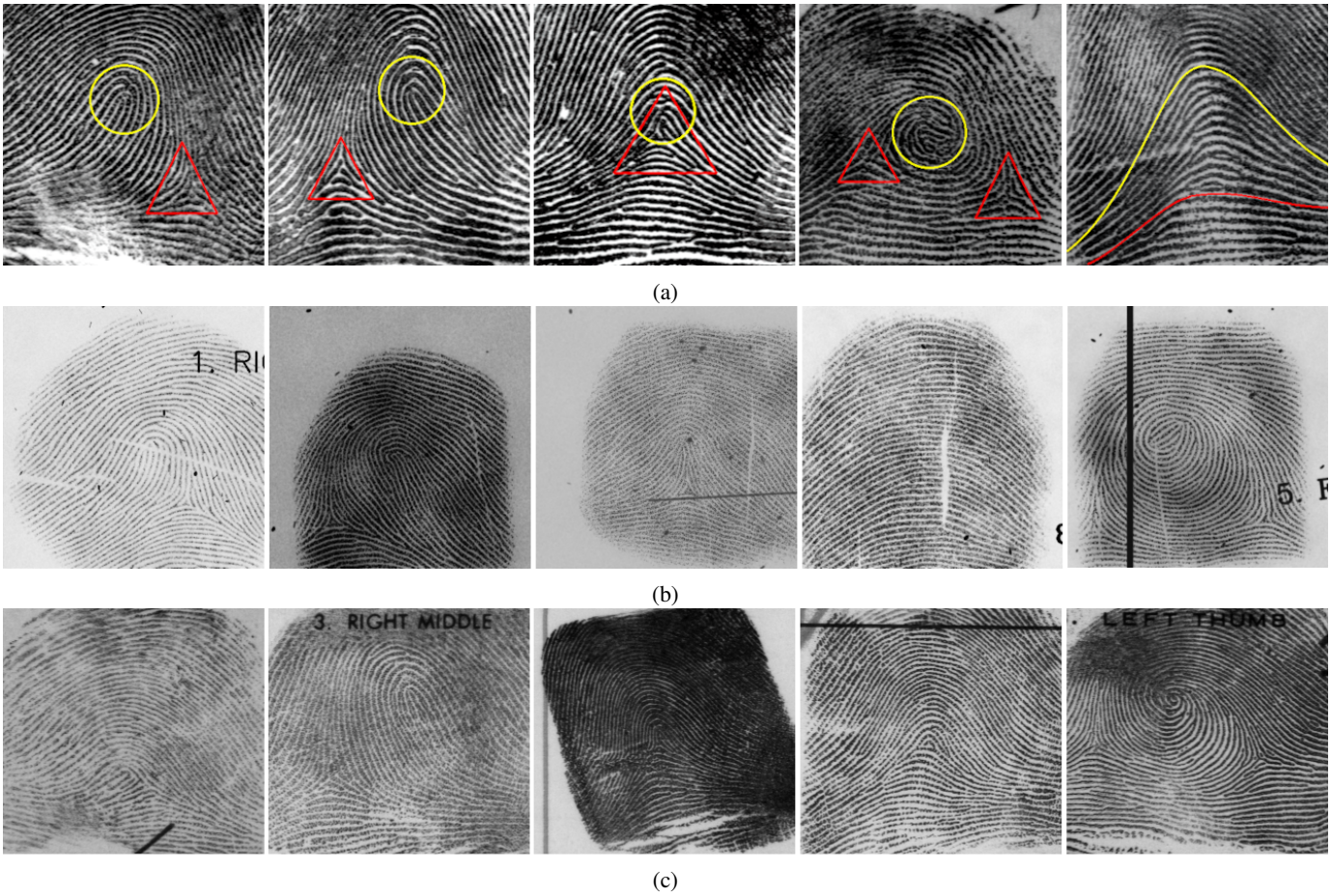


Fig. 8: (a) Fingerprint classes from NIST SD4 dataset [26]: left-loop, right-loop, tented arch, arch, and whorl classes. Loops are shown by yellow circles, and deltas are shown by red triangles. (b) Generated synthetic fingerprint examples for each class. (c) Real fingerprint examples from the NIST SD4 dataset for each class.

noise exhibits non-uniform behavior. To better reflect this, probability maps are generated similarly to coherent noise maps used for frequency images. These maps are scaled between 0 and 1 using min-max normalization (Figure 6b).

Background

Fingerprints are simulated to mimic imprinting on cards using ink and subsequent digitization by scanning, akin to the NIST SD4 dataset [26]. This dataset often includes fingerprint images with a paper background featuring additional horizontal or vertical lines, annotations, marks, and stains. To replicate this scenario, coherent noise is employed to generate paper-like backgrounds. Subsequently, lines, dots, marks, and annotations such as digits or class labels are randomly added to the background at various locations and scales (Figure 7).

IV. Experiments and Results

Fingerprints are categorized into five primary patterns based on the ridge lines: left loop, right loop, tented arch, whorl, and arch. This categorization displays a crucial role in reducing search space and subsequent search time for fingerprint matching. By comparing the queried fingerprint only with those of the same class, the accuracy of the fingerprint classification module significantly influences the overall performance of fingerprint recognition systems.

Traditional methods for fingerprint classification rely on the extraction of global (level 1) features such as ridge line flow and singular points, which are either of type core or delta, as shown in Figure 8a. Some notable studies include [27], [28], [29], [30]. More recently, deep learning algorithms, particularly convolutional neural networks (CNNs), have achieved high accuracy in fingerprint classification. These approaches use the fingerprint image directly as input and automatically learn relevant features for classification. The pioneering work by [31] introduced a stacked sparse autoencoder (SAE) neural network for learning a compact representation of fingerprint orientation fields, followed by subsequent studies [32], [33], [34], [35], [36], [37], [38]. However, the potential of these deep models is limited by the availability of large-scale, publicly accessible datasets for fingerprint classification.

For this study, synthetic fingerprint images that are known to belong one of the five classes are generated using the proposed method. An example synthetic fingerprint image for each class is given in Figure 8b.

The training efficacy of the generated dataset is evaluated

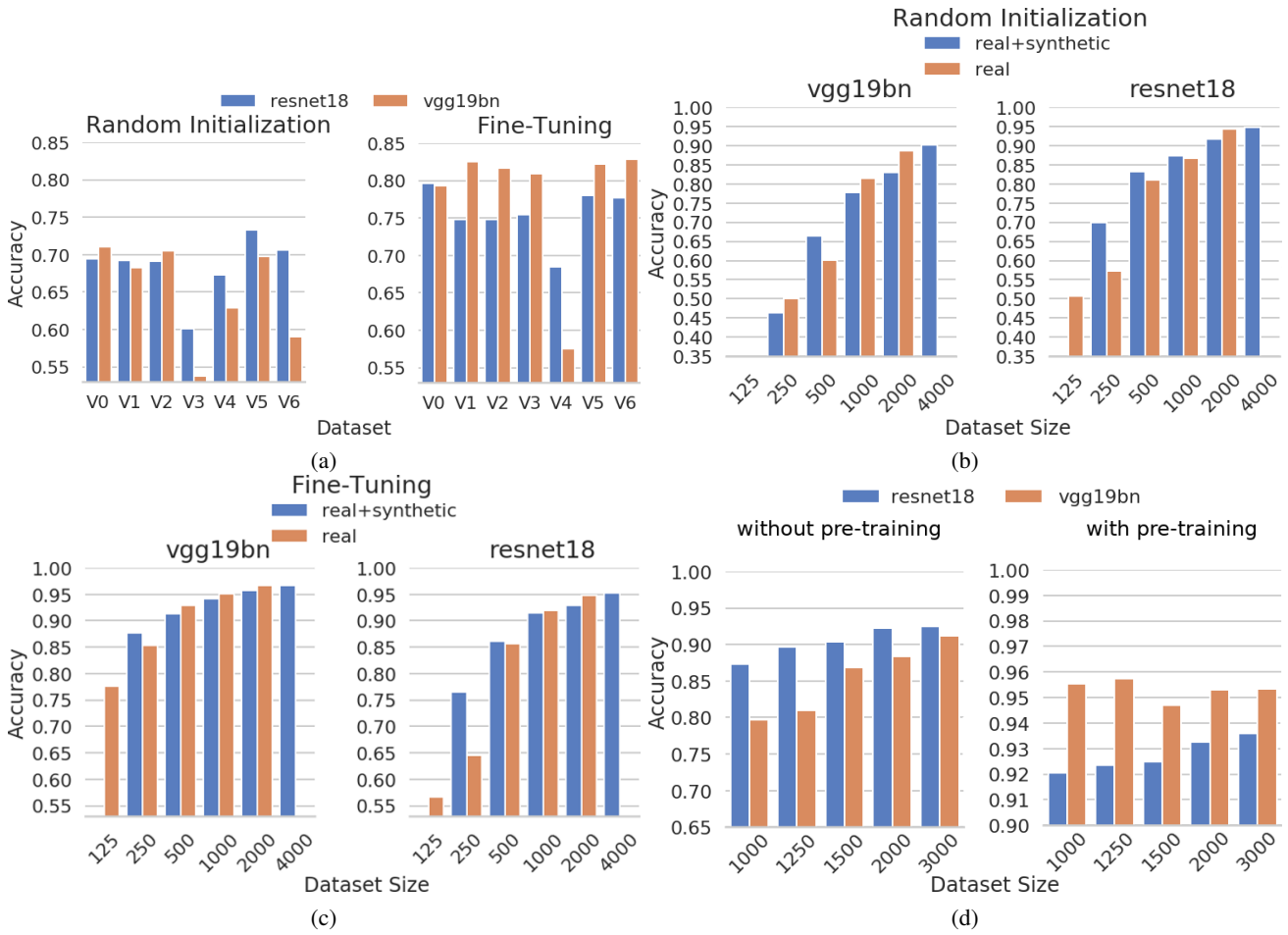


Fig. 9: (a) Classification performances only with synthetic training data (b) Classification performances in Group 1 experiments with VGG and (c) ResNet (d) Classification performances in Group 2 experiments

using the NIST SD4 dataset [26], which comprises 4000 grayscale fingerprint images sized 512x512, obtained from 2000 fingers. An example image for each fingerprint class are depicted in Figure 8c for comparison with the generated images.

The dataset is evenly divided into training and test sets with uniform distribution over five classes. The experiments are designed to achieve two main objectives: firstly, to determine whether synthetic fingerprints can effectively train fingerprint classification networks and achieve accurate results on real data, and secondly, to evaluate the impact of the proposed impression variations on performance. To this end, seven datasets are created that contain 8000 images with 1600 samples per class: Two with all variations and with (V0) and without post-processing (V6), five with one variation type excluded and with post-processing (V1-V5 for skin conditions, non-linear distortions, pose, noise, and background, respectively). In the initial experiments, only synthetic data is employed for training to evaluate the impact of impression variations on the results. A second set of experiments is conducted to assess the extent to which the synthetic fingerprints can enhance the performance when used as extra training data. All experiments are conducted using ResNet18 [39] and VGG19 [40], both with and without pre-training on the ImageNet [41].

A. *Synthetic fingerprints as the only training data*

This experiment evaluates classifier performance using exclusively synthetic data for training and analyzes the impact of post-processing and impression variations on accuracy, as shown in Figure 9a.

When no pre-training is used, pose variation (V3) and noise (V4) are observed to be rewarding in terms of performance. On the contrary, the exclusion of the background (V5) has increased the accuracy of ResNet, suggesting potential model confusion with background information rather than the fingerprint itself. VGG accuracy declines without post-processing (V6), likely due to differing intensity distributions between synthetic and real images, hindering generalization to real fingerprints.

With pre-training, performance notably improves accross most cases. Noise (V4) remains to be the most influential on the classification performance. On the other hand, pose variations (V3) lose significance. This is possibly because pre-trained models are adept at handling rotation, scale, and translation variations. However, ImageNet does not include the kind of noise observed on fingerprint images, and removing it causes a crucial loss of information.

 ISSN: 2147-284X https://dergipark.org.tr/bajece

B. Synthetic fingerprints as extra training data

Two sets of experiments are conducted using the V0 dataset to assess classification performances when synthetic fingerprints are employed as additional training data. Firstly, increasing the size of the real training set is matched with its synthetic counterpart. Secondly, the number of real samples in the training set is fixed while increasing the number of synthetic samples added.

Group 1 experiments are conducted with mixed training sets that are composed of N synthetic and N real data. Their performance is compared with the real training set of size 2N. Results for N=125, 250, 500, 1000, 2000 are given in Figure 9b and 9c. Notably, VGG without pre-training fails to converge with a training set of size 125 real samples.

With half of the training dataset consisting of synthetic, networks achieve comparable results to those trained solely on real images. More importantly, when the mixed dataset performances are compared with the real dataset performances with the same number of real images (N real+N synthetic vs. N real), the results reveal that adding synthetic data increases the performances in nearly all cases. Naturally, this improvement converges as the real dataset size increases.

Pre-trained VGG fine-tuned with a mixed dataset of size 4000 achieves the highest accuracy with 95.30%, outperforming the same model fine-tuned with all of the 2000 images in NIST SD4 training set, which achieves 94.80%.

Group 2 experiments are conducted to observe the impact of incrementally adding synthetic training data of varying sizes on classification accuracies. To this end, 1000 real fingerprint images from NIST SD4 are used as the base training set, and it is gradually augmented with synthetic samples. Results with 0, 250, 500, 1000, and 2000 additional synthetic images are given in Figure 9d.

Significant performance improvements are observed without pre-training for both models. Similar trends are noted for pre-trained ResNet. However, less pronounced effects are observed on pre-trained VGG. The leading cause for this result can be explained by the fact that fingerprint classification is not a very complex task, and the pre-trained VGG is already accurate due to the knowledge transferred from object classification on ImageNet. As such, 1000 real fingerprint images suffice to effectively fine-tune the classifier.

V. Conclusion

Motivated by the scarcity of publicly available datasets in the fingerprint domain, this study endeavors to generate realistic synthetic fingerprints and assess their efficacy in training deep learning systems. A model-based approach is employed, involving two primary stages: the generation of master fingerprints and ensuing impressions. Initially, master fingerprints representing five distinct classes are synthesized, capturing the idealized ridge patterns specific to each class. Subsequently, these master fingerprints undergo simulation of real-world variations such as skin conditions, non-linear distortions, pose variations, noise, and diverse backgrounds.

The experimental evaluation focuses on the performance of synthetic datasets in fingerprint classification task using two prominent deep neural network architectures, VGG and ResNet. Remarkably, classifiers trained exclusively on synthetic data achieve classification accuracies exceeding 80% on real test datasets. Furthermore, augmenting real training sets with synthetic samples consistently enhances classification performance, particularly beneficial when real dataset sizes are limited.

These findings underscore the potential of synthetic data generation techniques in addressing data scarcity challenges in fingerprint analysis. By effectively mimicking real-world variability, synthetic fingerprints not only facilitate robust training of deep learning models but also contribute to improving their generalization capabilities. Future research directions could explore refining synthetic fingerprint generation techniques to simulate additional real-world complexities and expanding evaluation across broader datasets and fingerprint analysis tasks.

Acknowledgment

This research is funded by The Scientific and Technological Research Council of Turkey (TÜBİTAK) via Project No. 217E092 under the 2515 COST Support Program.

References

[1] "Nist special database catalog," www.nist.gov/srd/shop/special-database-catalog, accessed: 2021-02-05.

[2] Y. Xu, Y. Wang, J. Liang, and Y. Jiang, "Augmentation data synthesis via gans: Boosting latent fingerprint reconstruction," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 2932–2936.

[3] R. Cappelli, D. Maio, and D. Maltoni, "Sfinge: an approach to synthetic fingerprint generation," in *International Workshop on Biometric Technologies (BT2004)*, 2004, pp. 147–154.

[4] A. H. Ansari, "Generation and storage of large synthetic fingerprint database," *ME Thesis, Jul*, 2011.

[5] K. Cao and A. Jain, "Fingerprint synthesis: Evaluating fingerprint search at scale," in *2018 International Conference on Biometrics (ICB)*. IEEE, 2018, pp. 31–38.

[6] V. Mistry, J. J. Engelsma, and A. K. Jain, "Fingerprint synthesis: Search with 100 million prints," in *2020 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE, 2019, pp. 1–10.

[7] "Sfinge tool, biolab, university of bologna," biolab.csr.unibo.it/research.asp, accessed: 2021-02-12.

[8] "Anguli, database systems lab, indian institute of science," dsl.cds.iisc.ac.in/projects/Anguli, accessed: 2021-02-12.

[9] R. Cappelli, D. Maio, and D. Maltoni, "Synthetic fingerprint-database generation," in *Object recognition supported by user interaction for service robots*, vol. 3. IEEE, 2002, pp. 744–747.

[10] B. G. Sherlock and D. M. Monro, "A model for interpreting fingerprint topology," *Pattern recognition*, vol. 26, no. 7, pp. 1047–1055, 1993.

[11] I. Fogel and D. Sagi, "Gabor filters as texture discriminator," *Biological cybernetics*, vol. 61, no. 2, pp. 103–113, 1989.

[12] P. Johnson, F. Hua, and S. Schuckers, "Texture modeling for synthetic fingerprint generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2013, pp. 154–159.

[13] Y. Chen and A. K. Jain, "Beyond minutiae: A fingerprint individuality model with pattern, ridge and pore features," in *International Conference on Biometrics*. Springer, 2009, pp. 523–533.

[14] Q. Zhao, A. K. Jain, N. G. Paulter, and M. Taylor, "Fingerprint image synthesis based on statistical feature models," in *2012 IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*. IEEE, 2012, pp. 23–30.

[15] K. G. Larkin and P. A. Fletcher, "A coherent framework for fingerprint analysis: are fingerprints holograms?" *Optics Express*, vol. 15, no. 14, pp. 8667–8677, 2007.

[16] C. Imdahl, S. Huckemann, and C. Gottschlich, "Towards generating realistic synthetic fingerprint images," in *2015 9th International Symposium on Image and Signal Processing and Analysis (ISPA)*. IEEE, 2015, pp. 78–82.
[17] C. Gottschlich and S. Huckemann, "Separating the real from the synthetic: minutiae histograms as fingerprints of fingerprints," *IET Biometrics*, vol. 3, no. 4, pp. 291–301, 2014.
[18] P. Bontrager, A. Roy, J. Togelius, N. Memon, and A. Ross, "Deepmasterprints: Generating masterprints for dictionary attacks via latent variable evolution," in *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. IEEE, 2018, pp. 1–9.
[19] M. Attia, M. H. Attia, J. Iskander, K. Saleh, D. Nahavandi, A. Abobakr, M. Hossny, and S. Nahavandi, "Fingerprint synthesis via latent space representation," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. IEEE, 2019, pp. 1855–1861.
[20] K. Perlin, "An image synthesizer," *ACM Siggraph Computer Graphics*, vol. 19, no. 3, pp. 287–296, 1985.
[21] R. Cappelli, D. Maio, and D. Maltoni, "An improved noise model for the generation of synthetic fingerprints," in *ICARCV 2004 8th Control, Automation, Robotics and Vision Conference, 2004.*, vol. 2. IEEE, 2004, pp. 1250–1255.
[22] L. Pang, J. Chen, F. Guo, Z. Cao, E. Liu, and H. Zhao, "Rose: real one-stage effort to detect the fingerprint singular point based on multi-scale spatial attention," *Signal, Image and Video Processing*, vol. 16, no. 3, pp. 669–676, 2022.
[23] J. Feng and A. K. Jain, "Fingerprint reconstruction: From minutiae to phase," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 2, pp. 209–223, 2011.
[24] P. R. Vizcaya and L. A. Gerhardt, "A nonlinear orientation model for global description of fingerprints," *Pattern Recognition*, vol. 29, no. 7, pp. 1221–1231, 1996.
[25] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, *Handbook of fingerprint recognition*. Springer Science & Business Media, 2009.
[26] C. I. Watson and C. L. Wilson, "Nist special database 4," *Fingerprint Database, National Institute of Standards and Technology*, vol. 17, no. 77, p. 5, 1992.
[27] K. Cao, L. Pang, J. Liang, and J. Tian, "Fingerprint classification by a hierarchical classifier," *Pattern Recognition*, vol. 46, no. 12, pp. 3186–3197, 2013.
[28] H.-W. Jung and J.-H. Lee, "Noisy and incomplete fingerprint classification using local ridge distribution models," *Pattern recognition*, vol. 48, no. 2, pp. 473–484, 2015.
[29] M. Liu, "Fingerprint classification based on adaboost learning from singularity features," *Pattern Recognition*, vol. 43, no. 3, pp. 1062–1070, 2010.
[30] R. Cappelli, D. Maio, D. Maltoni, and L. Nanni, "A two-stage fingerprint classification system," in *Proceedings of the 2003 ACM SIGMM workshop on Biometrics methods and applications*, 2003, pp. 95–99.
[31] R. Wang, C. Han, and T. Guo, "A novel fingerprint classification method based on deep learning," in *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE, 2016, pp. 931–936.
[32] J. M. Shrein, "Fingerprint classification using convolutional neural networks and ridge orientation images," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2017, pp. 1–8.
[33] W.-S. Jeon and S.-Y. Rhee, "Fingerprint pattern classification using convolution neural network," *international journal of fuzzy logic and intelligent systems*, vol. 17, no. 3, pp. 170–176, 2017.
[34] B. Pandya, G. Cosma, A. A. Alani, A. Taherkhani, V. Bharadi, and T. McGinnity, "Fingerprint classification using a deep convolutional neural network," in *2018 4th international conference on information management (ICIM)*. IEEE, 2018, pp. 86–91.
[35] P. Tertychnyi, C. Ozcinar, and G. Anbarjafari, "Low-quality fingerprint classification using deep neural network," *IET Biometrics*, vol. 7, no. 6, pp. 550–556, 2018.
[36] T. Zia, M. Ghafoor, S. A. Tariq, and I. A. Taj, "Robust fingerprint classification with bayesian convolutional networks," *IET Image Processing*, vol. 13, no. 8, pp. 1280–1288, 2019.
[37] B. Rim, J. Kim, and M. Hong, "Fingerprint classification using deep learning approach," *Multimedia Tools and Applications*, vol. 80, pp. 35 809–35 825, 2021.
[38] C. Militello, L. Rundo, S. Vitabile, and V. Conti, "Fingerprint classification based on deep learning approaches: experimental findings and comparisons," *Symmetry*, vol. 13, no. 5, p. 750, 2021.
[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
[40] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
[41] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

Emre İrtem received his B.Sc. degree from the Department of Computer Engineering at the Izmir Institute of Technology in 2017. He subsequently completed his M.S. studies in the same department in 2020. Emre is currently working as a software architect at a private bank.

Nesli Erdoğmuş is an Assistant Professor in the Department of Computer Engineering at the Izmir Institute of Technology. She earned her B.S. and M.S. degrees from the Department of Electrical and Electronics Engineering at Middle East Technical University. Dr. Erdogmus completed her Ph.D. studies at EURECOM in Sophia-Antipolis, France, and graduated from Télécom ParisTech. Following her doctoral studies, she conducted a two-year postdoctoral fellowship at Idiap Research Institute in Switzerland. Since 2016, Dr. Erdogmus has been leading her research group within her department.