Vikriti-ID: A Novel Approach For Real Looking Fingerprint Data-set Generation

Rishabh Shukla
Indian Institute of Technology Jammu
Jammu, India
rishabh.shukla@iitjammu.ac.in

Aditya Sinha
Indian Institute of Technology Jammu
Jammu, India
2021umt0192@iitjammu.ac.in

Vansh Singh
Indian Institute of Technology Jammu
Jammu, India
2021ume0234@iitjammu.ac.in

Harkeerat Kaur
Indian Institute of Technology Jammu
Jammu, India
harkeerat.kaur@iitjammu.ac.in

Abstract

Fingerprint recognition research faces significant challenges due to the limited availability of extensive and publicly available fingerprint databases. Existing databases lack a sufficient number of identities and fingerprint impressions, which hinders progress in areas such as Fingerprint-based access control. To address this challenge, we present Vikriti-ID, a synthetic fingerprint generator capable of generating unique fingerprints with multiple impressions. Using Vikriti-ID, we generated a large database containing 500000 unique fingerprints, each with 10 associated impressions. We then demonstrate the effectiveness of the database generated by Vikriti-ID by evaluating it for imposter-genuine score distribution and EER score. Apart from this we also trained a deep network to check the usability of data. We trained the network inspired from [13], on both Vikriti-ID generated data as well as public data. This generated data achieved an Equal Error Rate(EER) of 0.16%, AUC of 0.89%. This improvement is possible due to the limitations of existing publicly available data sets, which struggle in numbers or multiple impressions.

1. Introduction

In the past few decades, automatic fingerprint recognition systems have been widely used in various fields such as mobile recognition, payments, immigration, and access control [18]. Despite great progress in FVC, achieving a low false positive rate of 0.626% with a false positive rate of 0.06% in FVC, persistent challenges remain. The lack of publicly available fingerprint databases is a significant obstacle to solving this problem. An extensive database of multiple fingerprints with multiple impressions is important for the training and parameter estimation of algorithms.

For example, in the study [13], some of the 27000 fingerprints from the 30,000 synthetic fingerprints were taken to train the convolution network. This network is designed to generate new proxy fingerprints. The work will improve its performance if trained on multi impression dataset, which is not available in sufficient numbers. Despite their potential, some efforts [5], [24], [25], [15] struggles due to the lack of fingerprint data comparable to [8]. It is beneficial to avoid facial information provided by the Internet [18].

The lack of fingerprint training data not only hampers algorithm development but also hinders evaluation for large-scale studies involving millions or billions of games. This is especially important when considering the integration of fingerprint tracking algorithms into real-world systems such as India's Aadhaar and the FBI's NGI system [18].

Although there are several fingerprint databases available (FVC database, LivDet database, NIST 302 special database), they are limited. They have little identity, few effects on identity (usually 5-10), and data availability issues due to privacy regulations [18].

Tables 1 and 2 show the differences between fingerprint and face data. In contrast to free images (Table 2), the study [26] obtained 80 million facial images from the web to measure automatic face matchers [18]. To address the lack of data, research has explored synthetic fingerprint image generation algorithms. Synthetic fingerprints, which are unrelated to real people, avoid privacy regulations such as Institutional Review Board (IRB) approval [18]. However, existing methods [6], [28], [12], [19], [2], [21], [9], [4], [20], [27] does not serve well for algorithm training and evaluation;

- **Unrealism:** The synthetic fingerprint is very different from the real domain, creating a domain gap.
- **Limited variation:** Many generative adversarial networks (gan)-based methods [4], [20], [27], [3] struggle

to capture multiple impressions on the finger without changing the class.

To overcome this limitation, we present Vikriti-ID. It uses several GAN transfer modules and forces to generate realistic fingerprints for multiple real-looking impressions. Our quality assessment demonstrates Vikriti-ID's commitment to real fingerprints. Quantitative evidence includes minute distributions and matching scores from the MCC matcher.

After demonstrating the realism of Vikriti-ID synthetic fingerprints, we show how the synthetic data is generated. Vikriti-ID can be used to train deep networks to improve the performance of the models. In particular, we demonstrate by starting a deep network With a database of 500,000 Vikriti-ID fingerprints ($10000 Unique - ID * 5 classes$, 10 impressions per finger) then fine-tuning on the publicly available FVC and SOCOFing datasets Obtaining the Equal Error Rate (EER) of 0.1% which is more than 0.4% which is obtained if train the model only with publicly available datasets. By pointing out the ability to train networks on synthetic fingerprints and then do well in real fingerprints in some additional databases, we show that our synthetic fingerprints show inter class and intra class variability. Contribution of this study as follows:

- A synthetic fingerprint generator capable of creating more authentic fingerprints than modern methods. We demonstrate this through comprehensive qualitative and several quantitative measures.
- Existing synthetic fingerprint generators do not adequately model the required group interclass and intraclass variation. Where it is tested that our generated database has all the required properties related to interclass and intraclass variations.
- Our model is capable of generating multiple impressions of a particular class.

Finally, we generated a database of 50k synthetic fingerprint identities with 10 impressions generated from Vikriti-ID which sum up a total of 500k fingerprint images dataset. Tests show that our synthetic fingerprint generator does not "leak" identity information from its training database. This allows us to share our synthetic fingerprints securely to explore new avenues and interested researchers previously held back due to the lack of fingerprint database.

2. Related Work

During the past few decades, as a response to the lack of publicly available fingerprint databases, many studies have been conducted to produce authentic synthetic fingerprints. These methods can be broadly divided into two categories:

- Handmade or Engineered Methods [6], [28], [12]
- Learning-Based Methods [19], [2], [21], [9], [4], [20], [27], [3].

Although these approaches have undoubtedly made important contributions and made remarkable progress in creating a real fingerprint database, they are also characterized by certain limitations. In terms of quality, most synthetic fingerprint developers today struggle to produce fingerprints that are visually indistinguishable from real fingerprints. This difference appears when we compare the real fingerprints and the various synthetic fingerprints. The difference between real and synthetic fingerprints highlights the benefits of synthetic fingerprints for deep network training and evaluating fingerprint recognition systems.

In addition, most of the "handmade" methods are limited by assumptions or constraints imposed by the chosen model. For example: The model used to generate the excitation field (Pole Zero [23]) Mountain wave structure (AM/FM model [14] or Gabor Filter [10]) Minute points assume independence, resulting in unrealistic pattern. Recent advances in fingerprint synthesis aim to reduce the limitations of some "handmade" methods by using Generative Adversarial Networks (GAN) to learn how to convert random signals into synthetic fingerprints without introducing some of the aforementioned assumptions. This significantly improves the authenticity of fingerprints. However, it also introduces new limitations:

- Most GAN-based approaches focus on generating small fingerprints instead of full fingerprints to improve the stability of GAN training.
- Most GAN-based methods only produce unique fingerprints. [27] is limited to producing fingerprints, not fully rounded fingerprints, although it can produce several impressions for each finger. Existing GAN methods can not produce a large number of full fingerprint effects for individuals within classes.
- Due to the lack of training data, some GAN-based methods produce synthetic fingerprints that are more skewed than real fingerprints compared to the "handcrafted" approach.

GAN is often used without adding fingerprint-specific domain knowledge, which can improve the authenticity of synthetic fingerprints. On the other hand, [19], [2], [21], [9], [4], [20], [27], [3] works the same way as the learning-based synthesis method. Vikriti-ID uses a combination of VAE and GANs to create real-looking synthetic fingerprints that closely resemble real fingerprints 6. However, Vikriti-ID made significant improvements to the existing learning-based synthesis pipeline to address its limitations. First, Vikriti-ID integrates domain knowledge during synthesis in a way not seen in traditional GAN-based methods. Instead of directly mapping random signals to fingerprints through a single GAN, Vikriti-ID divides the synthesis into several steps, each of which focuses on modeling changes between classes. Vikriti-ID uses VAE to generate an intermediate image which acts as a starting point for the proposed

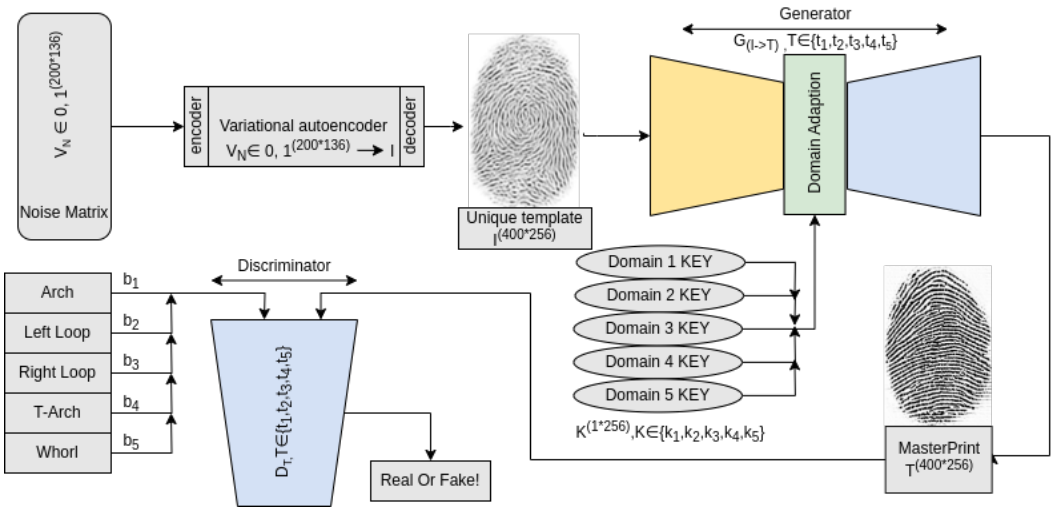

Figure 1. Training process of Vikriti-ID(Taking input as an random matrix and generating intermediate image by variational auto-encoder, which passes to generator model to generate unique identity with the help of discriminator.)

pipeline. Finally, using Vikriti-ID GAN, it simulates the impact of real real-looking fingerprint to create a unique ID. Finally, Vikriti-ID Impression generator is used to generate multiple real-looking impressions of this generated Unique-ID. Although there are related works that divide the synthesis process into steps, there are significant differences:

1. Other methods do not focus on class-wise fingerprint generation whereas Vikriti-ID generates samples with class-specific keys.
2. Vikriti-ID able to generate multiple impressions from a unique-ID.
3. It can be used directly to feed the data to a training model rather than storing it on disk. Which reduces extra space requirements. One more work using both synthetic fingerprint generation as well fingerprint template security is proposed by [13]. By using GANs to synthesize fingerprints, we take advantage of their ability to generate more reliable fingerprints than traditional manual approaches.

3. Methodology

The proposed work uses a set of steps to generate real-looking fingerprints. The steps can be summarized as follows:

- Variational autoencoders (VAEs) are used for intermediate image generation from a noise matrix $M_N \in 0, 1^{200*136*1}$.
- Generation of a unique fingerprint identity $T^{400*256}$ with the proposed Vikriti-ID.
- The last step employs the module of an impression generator (IG) to generate various modules of the unique fingerprint identity.

Let the generated unique fingerprint identity be called $T^{400*256}$ and noise matrix denoted as $M_N \in 0, 1^{200*136*1}$ and $I^{200*136*1}$ be the Intermediate image generated by VAE.The key used for class conversion is K^{1*256}. In Fig.3 all the five classes can be seen. The generative approach can be divided into three parts.

3.1. Intermediate Image Generation($M_N \in 0, 1^{200*136*1} \rightarrow I^{200*136*1}$):

In this step, we design and train variational autoencoders $VAE_{M_N \in 0,1^{200*136*1}}$ to take the input noise matrix $M_N \in 0, 1^{200*136*1}$ and generate an intermediate image $I^{200*136*1}$. This transformation majorly imparts semantic characteristics of a fingerprint template. Here we use a variational autoencoder ($VAE_{M_N \rightarrow I}$) to generate the intermediate image from the noise matrix $M_N \in 0, 1^{200*136*1}$. This transformation ensures that the final transformed template looks like some sample fingerprint.

3.2. Generating unique fingerprint identity from intermediate image($I^{200*136*1} \rightarrow T^{400*256}$):

This step uses the intermediate image ($I^{200*136*1}$) generated in the previous step and transforms it in latent representations (LV_I). LV_I is projected to new randomized mapping in the target class domain according to the class-specific key K. The randomized latent representation(LV_I') is then transformed using a trained generator model to generate a unique fingerprint identity $T^{400*256}$.

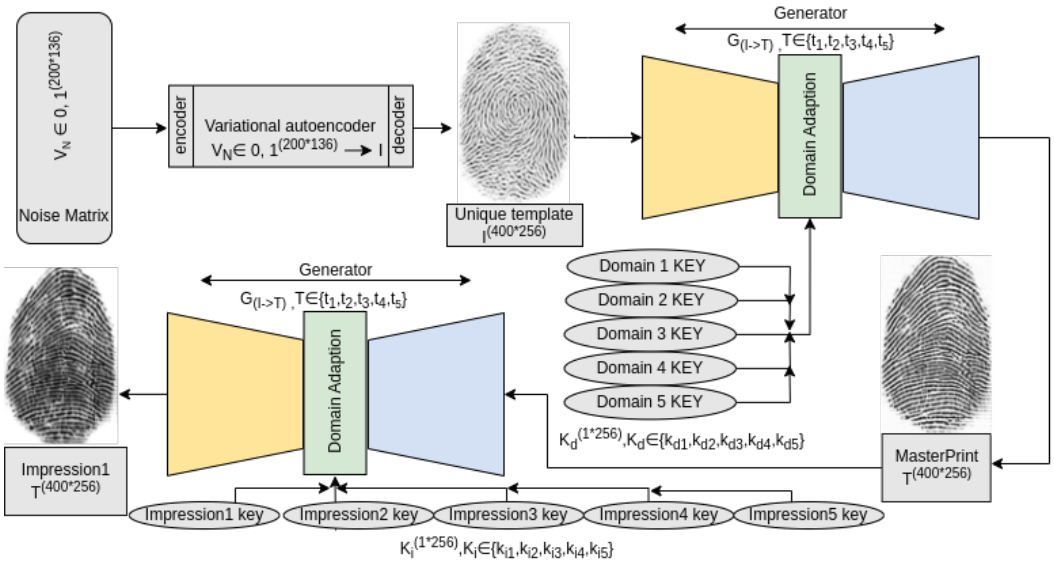

Figure 2. Illustration of Vikrit-ID generating impressions from the random noise matrix.

Figure 3. Types of fingerprints

3.3. Generating multiple impressions from unique fingerprint identity ($T_I^{400*256} \in I_1, I_1, I_1 - -I_N$):

Here we utilize an impression generator module $G_{T_I^{400*256} \in I_1, I_1, I_1 - - I_N}$ to generate the multiple impressions of the generated unique-ID (in the previous step). This transformation ensures that the impression must have variation between them but also store characteristics of intra and inter-user.

The above three steps are discussed in detail as follows.

4. Proposed Key-based Generative Approach

4.1. Variational autoencoder for intermediate image generation

This work uses variational autoencoder (VAEs) for intermediate template generation of fingerprint samples. VAEs are powerful models that can be used to generate new samples by learning underlying class distributions and allow domain translations between a different source domain to target domains [29]. Let $M_N \in 0, 1^{200*136*1}$ and $I^{200*136*1}$ be two different domain, where $M_N \neq I$. The VAE model is trained for image domain translation. The data set used for training contains fingerprint images of 1000 users having 4 impressions each. Which makes a total of 4000 (1000×4) image dataset. The dataset used was generated using Anguli software [1]. This VAE model learns the distribution of fingerprints and maps the input noise matrix $M_N \in 0, 1^{200*136*1}$ to intermediate image I. Let $VAE_{(M_{N \in 0,1^{200*136*1}} \rightarrow I^{200*136*1})}$ denote the variational autoencoder model trained to transform random samples to image $I^{200*136*1}$. Let $M_N \in 0, 1^{200*136*1}$ denote the input random sample. The input image $I^{200*136*1}$ is passed through $VAE_{(M_N \in \{0,1\}^{200 \times 136 \times 1)}} \rightarrow I^{200 \times 136 \times 1}$ to generate $I^{200 \times 136 \times 1}$. The loss function for this network can be defined as,

$$L = L\text{recon} + L\text{reg}$$

where Lrecon is the reconstruction loss and Lreg is the regularization loss. Regularisation loss can be expressed in terms of Kulback-leibler-divergence,

$$L\text{reg} = \|y - \hat{y}\|^2 + KL[(\mu_x, \sigma_x), N(\mu_y, \sigma_y)]$$

The first term in the equation, $\|y - \hat{y}\|^2$, is simply the squared L2 norm of the difference between the true output y and the predicted output $\hat{y}$. The second term in the equation, $KL[N(\mu_x, \sigma_x), N(\mu_y, \sigma_y)]$, is the KL divergence between two Gaussian distributions. The first Gaussian distribution is characterized by its mean μ_x and standard deviation σ_x, which represent the prior distribution over the input features x. The second Gaussian distribution is characterized by its mean μ_y and standard deviation σ_y, which represent the distribution over the predicted outputs $\hat{y}$.

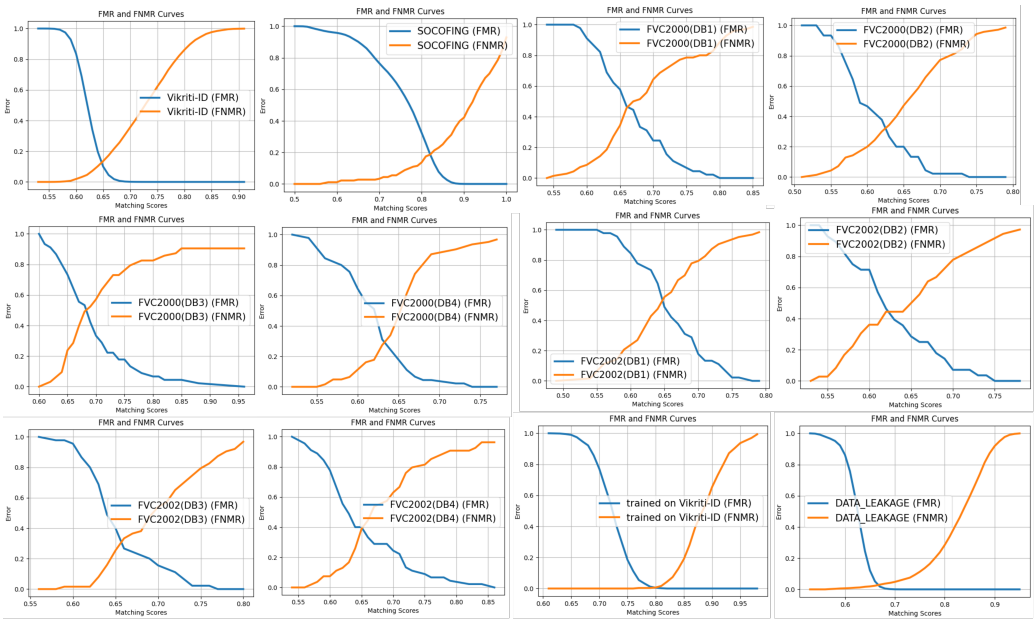

Figure 4. FMR-FMNR graph on Vikriti-ID, SOCOFING, FVC, and deep convolution model trained on Vikriti-ID generated data.

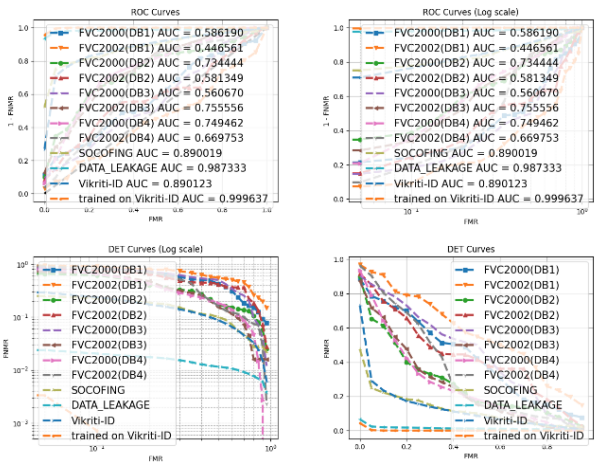

Figure 5. ROC and DET curves(LOG and normal) on Vikriti-ID, SOCOFING, FVC, and deep convolution model trained on Vikriti-ID generated data.

4.2. Generating unique fingerprint identity from intermediate image

Once we get the intermediate image $I^{200*136*1}$, the next trick is to assign randomized mapping of this intermediate image $I^{200*136*1}$ to a new template $T^{400*256}$. We train deep U-Net-based generative network. The idea of this model is inspired by the idea of star Gan [7] and conditional GAN [11]. Given an input image $I^{400*256}$, passing it through generative module $G_{I\to T}$ we get a lower-dimensional encoded representation, denoted as $I_L^{(1600,256)}$. The generative module $G_{I\to T}$ takes the input as intermediate image $I^{200*136*1}$ to minimize the perceptual loss defined as:-

$$\mathcal{L}_{\text{perceptual}} = \sum_i \|\phi_i(\mathbf{I}) - \phi_i(\mathbf{I}_{\text{target}})\|_2^2$$

where: $\mathcal{L}_{\text{perceptual}}$ represents the perceptual loss, ϕ_i represents the feature map extracted from layer (i) of the generative network, $\|\cdot\|_2$ denotes the L2 norm.

While we are training the network $G_{I\to T}$, we are extracting the lower dimension representations $I_L^{(1600,256)}$ from the intermediate image $I^{200*136*1}$ by passing it through the first half layers of the generative model($G_{I\to T}$). We play with this main latent matrix to map it to a new representation of a specific class domain by projecting it on some orthonormal random matrices defined as keys. Let K be the random matrix used as a projection key specific to a class. To obtain a new representation of the intermediate image $I^{200*136*1}$, the main lower dimension representation $I_L^{(1600,256)}$ is projected on a random matrix K, i.e., $I_L^{'(1600,256)} = K.I_L^{(1600,256)}$. This lower-dimensional representation will be transformed into a unique identity.

4.3. Real looking impression generator

The identity $T^{400*256}$ has certain characteristics reminiscent of a synthetic fingerprint, while it does not have any impressions. In order to create a realistic appearance, the distinct identity $T^{400*256}$ is given to the Impression generator $I^G_{T\to T_N\in(T_2,T_3....T_n)}$, as seen in Figure 2. The final impression is computed as Vikriti-ID $T_N = V^{ID}_{M_N\in 0,1^{200*136*1}\to T_N\in(T_2,T_3....T_n)}$.

The whole process can be illustrated as:

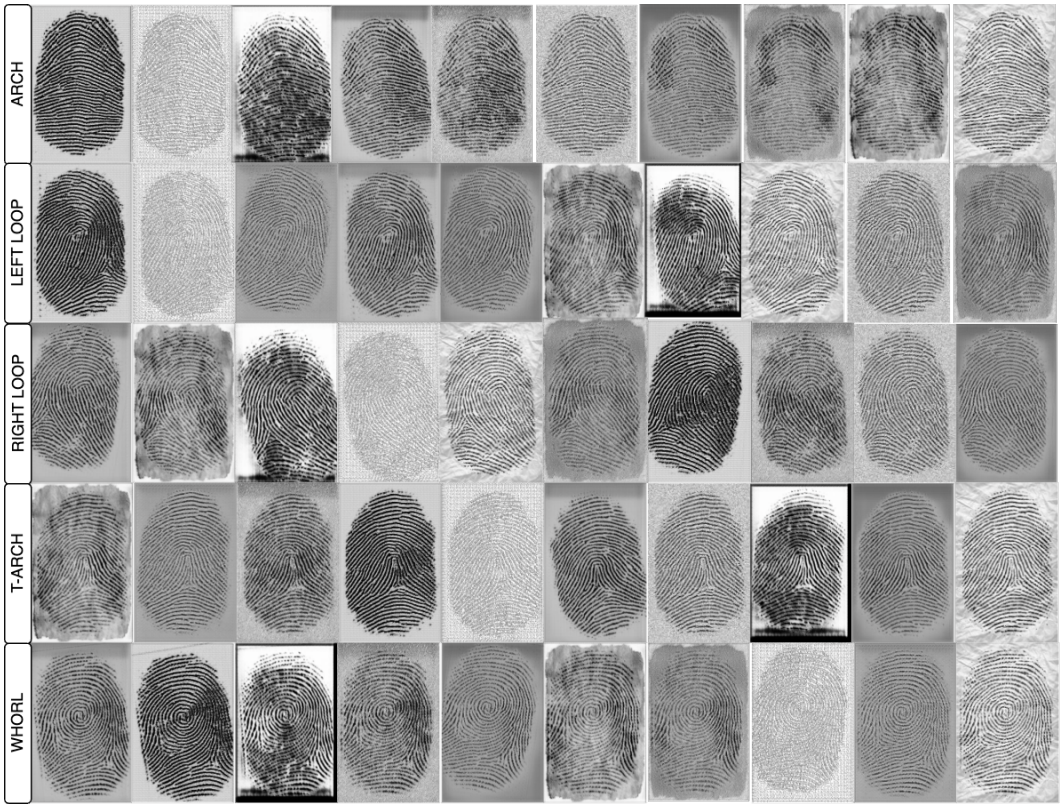

Figure 6. Sample impressions generated using Vikrit-ID. Top to bottom(arch,left loop,right loop,tented arch,whorl)

Algorithm 1 $Gen\_Real\_Impr(M_N, I, T, K)$

Require: Trained $VAE_{(M_N \in 0,1^{200*136*1} \rightarrow I^{200*136*1})}$

1: $I^{200*136*1} \leftarrow VAE_{M_N \rightarrow I}(I)$ *// generate intermediate image* $I^{200*136*1}$

2: $I_L^{'(1600,256)} \leftarrow G_{I \rightarrow T}$ *// extract latent matrix of* B_T

3: $I_L^{'(1600,256)} \leftarrow I_L^{'(1600,256)}.K$ *// random projection of LV on key K respective to their class*

4: $G_{I \rightarrow T}$ *// Generate class specific unique ID from projected data*

5: $T_N \leftarrow I^G_{T \rightarrow T_N \in (T_2, T_3 .... T_n)}$ *//generate final impressions*

6: **return** T_N

Figure 2 illustrates the process described in the above three subsections. The step-wise process also consists in Algorithm 1. The network consisting of VAE and GAN network is designed to process the input and output variables with the following dimensions, I and $M_N \in \mathcal{R}^{200 \times 136 \times 1}$, I_L and $I'_L \in \mathcal{R}^{1600 \times 256}$, $T \in \mathcal{R}^{400 \times 256}$. Figure 6 illustrates some impressions generated using the network proposed. The class-specific semantics and natural looks of the generated impression can be clearly observed.

5. Experimental Results and Analysis

To test the efficiency of the proposed work we ensure that the generated impression samples must perform equally well and better as compared to other available publicly available datasets. To test the performance of the generated dataset we generated 10000 Unique IDs of each class having 10 samples for each unique ID. This gives a total of 10000 subjects (5×10) and a total of 500000 fingerprint image samples.

5.1. Quantitative Performance

Given 500000 samples of five classes, 200 subjects per class are selected randomly with 4 samples per subject. This gives a total of 1000 subjects (5×200) and a total of 4000 fingerprint image samples. Let this data subset be called as '$gen\_database$'. The evaluation takes into consideration two scenarios: - intra-user scenario and inter-user scenario, which are described below. For matching the fingerprints we have used MCC matcher.

a) Intra user Scenario: In this case, 3 impressions $T_N T_2, T_2, T_3$ of every unique ID T in $gen\_database$ compared with each other to compute the matching scores between them. For this we have created a dataset holds total 100 unique IDs T. Its generated impression samples $T_N T_1, T_2, T_3$ in a random manner in all classes. So we

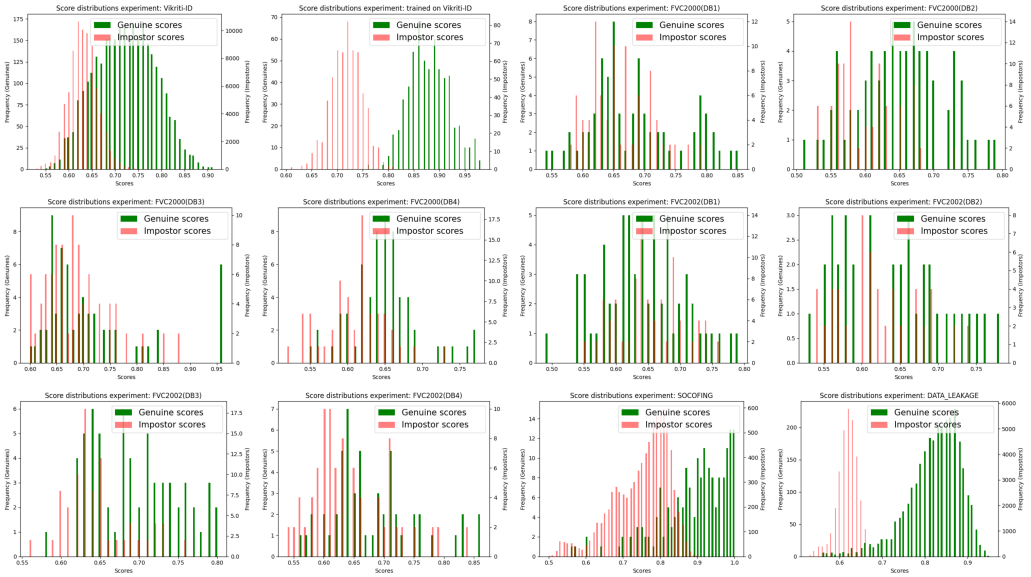

Figure 7. Distribution of data on Vikriti-ID, deep convolution model trained on Vikriti-ID generated data, FVC, and SOCOFING.

Table 1. Comparison of performance between Vikriti-ID and other publicly available datasets using MCC matcher.

Metric	Database	EER%	AUC	remarks
Comparision	Vikriti-ID	0.16%	0.9	Performing good as compared to other datasets.
	SOCOFING [22]	0.17%	0.89	Nearly equal to Vikriti-ID.
	FVC 2000(DB1) [16]	0.46	0.58	Good Performance
	FVC 2000(DB2) [16]	0.30%	0.73	High AUC as compared to other FVC
	FVC 2000(DB3) [16]	0.47%	0.56	Average Performance
	FVC 2000(DB4) [16]	0.29%	0.74	High AUC as compared to other FVC
	FVC 2002(DB1) [17]	0.52%	0.44	Average Performance
	FVC 2002(DB2) [17]	0.41%	0.58	Average Performance
	FVC 2002(DB3) [17]	0.3%	0.75	Highest AUC as compared to other FVC
	FVC 2002(DB4) [17]	0.38%	0.66	Performing well.
Performance	Data leakage	0.02%	0.98	There is no leakage from the training dataset in generated samples.
	Trained on generated dataset	0.005%	0.99	Performing well compared to other datasets.

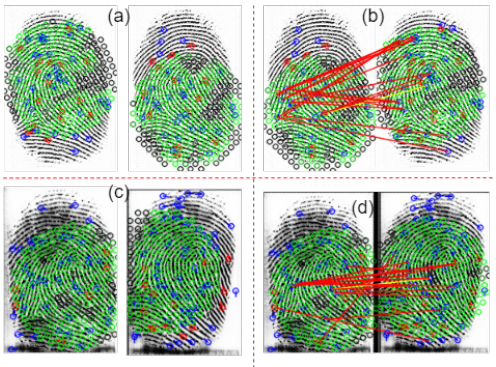

Figure 8. Minutiae points in generated image with their Local structures in (a),(c), matching using MCC matcher in (b),(d).

tested it by matching same user impressions with each other, for instance, user i has four impressions so we match user i's impresion 1 (i_1) with user i's impresion 2 (i_2). This way for each subject we generated a new impression, thereby giving us a dataset of 300 samples ($100(\#impressions) \times 3(\#samples)$) called as '$gen\_database\_intra$'.

b) Inter User Scenario: This scenario studies the case where any unique user wants to authenticate himself then it should not matched with any other ID. Here, every unique ID i in $gen\_database$ is matched with all other IDs in the database $gen\_database$. This dataset can be denoted as $gen\_database\_inter$. The EER, FAR,FRR,and AUC curves scores are reported in fig4, 5 and 7.

The generated samples in dataset must preserve intra-user and intra-user distinctive characteristics. Within each class, impressions belonging to the same subject must match (genuine matches), whereas impressions belonging to different subjects must not match (imposter matches). The matching scores are computed using Minutia Cylinder-Code (MCC). MCC is one of the most popular, widely used, and accurate fingerprint descriptors. MCC associates a local structure to each minutia which encodes spatial and directional relationships between the minutia and its (fixed-radius) neighborhood and can be conveniently represented as a cylinder whose base and height are related to the spatial and directional information, respectively. Here we will consider only the base of the cylinders, which is rotated according to the minutia direction and discretized into a fixed number of cells. The local structures r_1 and $r2$ and their respective minutia points are computed for two fingerprint images are shown in Fig. 8 where the similarity between two local structures $r1$ and $r2$ can be simply computed as

$$similarity(r_1, r_2) = 1 - \frac{\|r_1 - r_2\|}{\|r_1\| + \|r_2\|}$$

The performance results of this work in terms of FAR (False accept rate) and FRR (False reject rate) are reported for original data samples and generated data samples in both inter and intra user scenarios using the above-mentioned techniques in Table 1. It can be observed that the matching performance is nearly the same and better than the publicly available datasets.

The FAR-FRR curves, Genuine-Imposter score distribution graphs, and ROC and DET curves are also provided in Fig. 4, 5 and 7 for the various scenarios. A good score separation is observed Fig.7 in support of the obtained results.

5.2. Imposter Distribution

In the subsequent analysis, we have conducted computations on several distributions of impostor scores. The purpose of this analysis is two-fold: firstly, to identify instances of identity leaking from the authentic fingerprint training database, and secondly, to discover such instances within the synthetic fingerprints that have been created. The objectives of this study are to:
1) Investigate the effectiveness of Vikriti-ID in generating fingerprint IDs.
2) Assess the level of uniqueness shown by the created fingerprint identities.

5.3. Leakage of data

We have calculated any leakage in the produced fingerprints. For this we calculated match score between 50000 from the training dataset to the 50000 unique IDs of generated dataset $gen\_database\_intra$, which will result in 2.5 billion total scores. To reduce the time and computation load we randomly selected 1000 training fingerprint samples and matched them with 1000 unique IDs of actual generated database $gen\_database\_intra$ IDs. The total number of comparisons is One million here. Which greatly cut the time needed to execute 1 million calculations matches. In this experiment, we got the EER as reported in table 1 is 0.01%. Figure4, 5 and 7. showing the graphs for this.

5.4. Comparison with other real-looking data

As the main motivation of this is to solve the problem of the unavailability of real-life fingerprint dataset, we tested the generated fingerprint dataset $gen\_database$ with some publicly available dataset such as SOCOfing and FVC. For this, we have calculated EER,FAR,FRR for all of these datasets. The EER for socofing and FVC200 is 0.17% and 0.4% respectively. Whereas the EER for impression generated from Vikriti-ID $gen\_database\_intra$ is 0.16%. All these scores are reported in table1. Graphs plotted on this comparison can be seen in fig4, 5 and 7.

5.5. Usability

This section explains the Usability of the unique IDs and their respective generated from Vikriti-ID. For this, we trained a deep network based on [13]. The training data is divided into 2 parts.
1)generated data $gen\_database\_intra$ is used for training.
2)FVC2002 is used for fine-tuning.
Here we used $Gen\_Real\_Impr(M_N, I, T, K)$ to generate the dataset for the first part. The EER,FAR,and ROC curves can be seen in fig4, 5 and 7. Also, the EER,FAR,and ROC curves after training the deep model on FVC2002 can be seen in fig4, 5 and 7.

6. Conclusion and Future Work

In conclusion, our research tackled the challenge of limited fingerprint databases by introducing Vikriti-ID, a synthetic fingerprint generator capable of creating unique fingerprints with multiple impressions. Our database, containing 500,00 fingerprints with 10 impressions each, showcased its effectiveness through imposter-genuine score distribution and EER evaluations. Our deep network model, trained on Vikriti-ID data and publicly available datasets, achieved an impressive EER of 0.16% and AUC of 0.9. Looking ahead, future work could focus on refining Vikriti-ID's fingerprint generation techniques using GANs and exploring multi-modal fusion for enhanced accuracy. Real-world testing, addressing adversarial attacks, and assessing generalization to different domains will be pivotal. Additionally, the continuous update of the Vikriti-ID database, considering privacy implications, and sharing it with the research community will support the advancement of fingerprint recognition technology.

References

[1] Afzalul Haque Ansari. Generation and storage of large synthetic fingerprint database. *ME Thesis, Jul*, 2011. 4

[2] Mohamed Attia, MennattAllah H Attia, Julie Iskander, Khaled Saleh, Darius Nahavandi, Ahmed Abobakr, Mohammed Hossny, and Saeid Nahavandi. Fingerprint synthesis via latent space representation. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 1855–1861. IEEE, 2019. 1, 2

[3] Keivan Bahmani, Richard Plesh, Peter Johnson, Stephanie Schuckers, and Timothy Swyka. High fidelity fingerprint generation: Quality, uniqueness, and privacy. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 3018–3022. IEEE, 2021. 1, 2

[4] Kai Cao and Anil Jain. Fingerprint synthesis: Evaluating fingerprint search at scale. In *2018 International Conference on Biometrics (ICB)*, pages 31–38. IEEE, 2018. 1, 2

[5] Kai Cao and Anil K Jain. Fingerprint indexing and matching: An integrated approach. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pages 437–445. IEEE, 2017. 1

[6] Raffaele Cappelli, Dario Maio, and Davide Maltoni. Synthetic fingerprint-database generation. In *2002 International Conference on Pattern Recognition*, volume 3, pages 744–747. IEEE, 2002. 1, 2

[7] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 5

[8] Joshua J Engelsma, Kai Cao, and Anil K Jain. Learning a fixed-length fingerprint representation. *IEEE transactions on pattern analysis and machine intelligence*, 43(6):1981–1997, 2019. 1

[9] Masud An-Nur Islam Fahim and Ho Yub Jung. A lightweight gan network for large scale fingerprint generation. *IEEE Access*, 8:92918–92928, 2020. 1, 2

[10] Itzhak Fogel and Dov Sagi. Gabor filters as texture discriminator. *Biological cybernetics*, 61(2):103–113, 1989. 2

[11] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 5

[12] Peter Johnson, Fang Hua, and Stephanie Schuckers. Texture modeling for synthetic fingerprint generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 154–159, 2013. 1, 2

[13] Harkeerat Kaur, Rishabh Shukla, Isao Echizen, and Pritee Khanna. Secure and privacy preserving proxy biometric identities. In *International Conference on Advanced Information Networking and Applications*, pages 487–500. Springer, 2023. 1, 3, 8

[14] Kieran G Larkin and Peter A Fletcher. A coherent framework for fingerprint analysis: are fingerprints holograms? *Optics express*, 15(14):8667–8677, 2007. 2

[15] Ruilin Li, Dehua Song, Yuhang Liu, and Jufu Feng. Learning global fingerprint features by training a fully convolutional network with local patches. In *2019 International Conference on Biometrics (ICB)*, pages 1–8. IEEE, 2019. 1

[16] Dario Maio, Davide Maltoni, Raffaele Cappelli, James L. Wayman, and Anil K. Jain. Fvc2000: Fingerprint verification competition. *IEEE transactions on pattern analysis and machine intelligence*, 24(3):402–412, 2002. 7

[17] Dario Maio, Davide Maltoni, Raffaele Cappelli, James L Wayman, and Anil K Jain. Fvc2002: Second fingerprint verification competition. In *2002 International conference on pattern recognition*, volume 3, pages 811–814. IEEE, 2002. 7

[18] Davide Maltoni, Dario Maio, Anil K Jain, Salil Prabhakar, et al. *Handbook of fingerprint recognition*, volume 2. Springer, 2009. 1

[19] Shervin Minaee and Amirali Abdolrashidi. Finger-gan: Generating realistic fingerprint images using connectivity imposed gan. *arXiv preprint arXiv:1812.10482*, 2018. 1, 2

[20] Vishesh Mistry, Joshua J Engelsma, and Anil K Jain. Fingerprint synthesis: Search with 100 million prints. In *2020 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–10. IEEE, 2020. 1, 2

[21] M Sadegh Riazi, Seyed M Chavoshian, and Farinaz Koushanfar. Synfi: Automatic synthetic fingerprint generation. *arXiv preprint arXiv:2002.08900*, 2020. 1, 2

[22] Yahaya Isah Shehu, Ariel Ruiz-Garcia, Vasile Palade, and Anne James. Sokoto coventry fingerprint dataset. *arXiv preprint arXiv:1807.10609*, 2018. 7

[23] Barry G Sherlock and Donald M Monro. A model for interpreting fingerprint topology. *Pattern recognition*, 26(7):1047–1055, 1993. 2

[24] Dehua Song and Jufu Feng. Fingerprint indexing based on pyramid deep convolutional feature. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pages 200–207. IEEE, 2017. 1

[25] Dehua Song, Yao Tang, and Jufu Feng. Aggregating minutia-centred deep convolutional features for fingerprint indexing. *Pattern Recognition*, 88:397–408, 2019. 1

[26] Dayong Wang, Charles Otto, and Anil K Jain. Face search at scale. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1122–1136, 2016. 1

[27] André Brasil Vieira Wyzykowski, Mauricio Pamplona Segundo, and Rubisley de Paula Lemes. Level three synthetic fingerprint generation. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 9250–9257. IEEE, 2021. 1, 2

[28] Qijun Zhao, Anil K Jain, Nicholas G Paulter, and Melissa Taylor. Fingerprint image synthesis based on statistical feature models. In *2012 IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 23–30. IEEE, 2012. 1, 2

[29] Yang Zhao and Changyou Chen. Unpaired image-to-image translation via latent energy transport. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16418–16427, 2021. 4