Directed Adversarial Attacks on Fingerprints using Attributions

Steven Fernandes[1], Sunny Raj[1], Eddy Ortiz[2], Iustina Vintila[2], Sumit Kumar Jha[1]
[1]Computer Science Department, University of Central Florida, Orlando, FL, USA
[2]Solution Acceleration and Innovation Department, Royal Bank of Canada
{steven, sraj, jha}@cs.ucf.edu {eddy.ortiz, iustina.vintila}@rbc.com

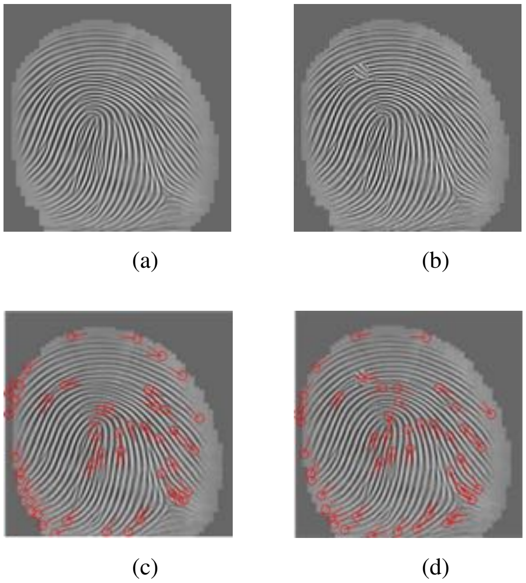
Abstract

Fingerprint recognition systems verify the identity of individuals and provide access to secure information in various commercial applications. However, with advancements in artificial intelligence, fingerprint-based security methods are vulnerable to attack. Such a breach has the potential to compromise confidential, private and valuable information. In this paper, we attack a state-of-the-art fingerprint recognition system based on transfer learning. Our approach uses attribution analysis to identify the fingerprint region crucial to correct classification, and then perturbs the fingerprint using error masks derived from a neural network to generate an adversarial fingerprint.

Image quality assessment metrics applied to calculate the difference between the original and perturbed fingerprints include average difference, maximum difference, normalized absolute error, and peak signal to noise ratio. On the ATVS fingerprint dataset, the differences between these values in the original and corresponding perturbed fingerprint images are negligible. Further, the VeriFinger SDK is used to detect the minutiae and perform matching between the original and perturbed fingerprints. The matching score is above 250, which reinforces the fact that there is virtually no loss between the original and perturbed fingerprints.

1. Introduction

Fingerprints are unique, highly-detailed graphical patterns formed by the ridges and valleys on the surface of human fingers. Given that fingerprints cannot be readily altered, their use as an alternative to password-based or PIN-enabled security has become more visible in applications such as mobile phones. Fingerprints are used in a wide variety of applications, including criminal investigations [11], access controls [26], and other commercial use [28].



(a) (b) (c) (d)

Figure 1: (a) Original enhanced fingerprint obtained from FingerNet (b) Perturbed adversarial fingerprint (c) Minutiae detected in original fingerprint (d) Minutiae detected in perturbed fingerprint using the VeriFinger SDK. Images (a)-(d) are from the ATVS fingerprint dataset.

With the recent advances in artificial intelligence and deep learning, it is possible to circumvent fingerprint recognition systems. In this paper we present DeepAttack, where fingerprints from the ATVS fingerprint dataset [6] are enhanced using FingerNet [24] as shown in Fig. 1(a). The enhanced fingerprints are used to train transfer learning-based fingerprint recognition systems. This approach has a high classification accuracy, even on small datasets. It was a transfer learning-based technique that won first prize at the fingerprint liveness detection competition [16], with an accuracy of 95.51%. These algorithms are also used to detect liveness in latent fingerprints [15].

978-1-7281-3640-0/19/$31.00 ©2019 IEEE

Attributions [30] are used to find the discriminative image regions that are considered by the convolutional neural network (CNN) while making the predictions. A pixel coordinate window of size 14 x 14, encapsulating the region having maximum intensity, is selected from the discriminative image region that is highlighted by the attributions. The corresponding 14 x 14 pixel coordinate window is obtained from the original fingerprint data and saved in a matrix. The error codes are generated using [17] which considers the first convolutional layer. The pixel values obtained from the error codes, each corresponding to the 14 x 14 window from the class activation map, are normalized by dividing each pixel value by 255. A kernel is then generated by subtracting the resulting values from the existing matrix. The kernel is subsequently slid along the original fingerprint in order to generate a set of perturbed fingerprints. The perturbed fingerprint images are used to launch an attack against the transfer learning-based fingerprint recognition system using attributions.

This is a recursive process where an unsuccessful attack will instruct the system to further perturb the fingerprints. This is performed by again subtracting the normalized error code from the original fingerprints. This process is repeated until either the attack on the fingerprint image is successful or 8000 perturbations have been performed. Beyond this threshold, it is more likely that the quality of the perturbed image will suffer degradation, ultimately leading to a lower similarity.

The first test for similarity between the original and perturbed fingerprints is undertaken by calculating several pixel difference-based image quality assessment (IQA) parameters [7]. These parameters are the average difference (AD), maximum difference (MD), normalized absolute error (NAE), and peak signal to noise ratio (PSNR). Together, these are the first set of metrics for determining similarity.

The VeriFinger SDK [1] is used as a second method for determining similarity between images. The process detects the minutiae and performs matching between the original and perturbed fingerprint images. One-to-many matching is performed in order to compare the original and perturbed versions. This yields a score, where a higher score is indicative of greater similarity between the images.

Figure 1 (a) shows the output obtained from FingerNet. This image is perturbed using DeepAttack, and the output obtained from this process is shown in Figure 1 (b). The VeriFinger SDK, which is NIST [12] MINEX compliant, is used to obtain the minutiae from fingerprint images. The VeriFinger output based on the FingerNet image is shown in Fig. 1 (c), whereas the minutiae obtained from the perturbed image is shown in Fig. 1 (d).

To the best of our knowledge, this is the first attempt to attack a transfer learning-based fingerprint recognition system.

2. Related Work

Deep learning is used extensively in the tasks of image classification [22], object recognition [27], image reconstruction [20], image stitching [3], and others. It is also applied for the purposes of fingerprint enhancement [14], and minutiae-based matching [2]. Researchers have employed deep learning models for fingerprint liveness detection [29]. Nogueira et al. [5] used a CNN and local binary patterns [10] for liveness detection. It was extended by them [16] using AlexNet [13], and pre-trained on ImageNet [18]. Nogueira et al. [16] showed that the pre-trained model developed to detect objects can be fine-tuned for liveness detection of fingerprints [18]. The refined model produces better results when compared to a model that is initialized using random weights.

Recently, Marasco et al. [15] utilized VGG19 for detecting liveness in latent fingerprints. The feature extraction was performed using 16 convolutional layers, with the final 3 layers forming the classifier. The model was fine-tuned by selecting the last 3 layers of the pre-trained model; this helps to avoid overfitting while training the CNN. The Biometrika FX2000 is the only common fingerprint scanner used by both Nogueira et al. [16] and Marasco et al. [15] for experimental evaluation. The Biometrika FX2000 uses a flat optical sensor and it is FBI-compliant with 128-bit encryption. The fingerprints have 400 x 560 pixels, with a resolution of 512 pixels per inch (PPI). Hence, we have used the ATVS fingerprint dataset, which contains fingerprints obtained from the Biometrika FX2000 scanner.

Apart from VGG19, we have implemented transfer learning using 11 other image recognition models on the ATVS fingerprint dataset. They are MobileNetV2 [19], MobileNet [8], NASNetMobile [31], DenseNet121 [9], DenseNet169 [9], DenseNet201 [9], Xception [4], InceptionV3 [23], ResNet50 [25], NASNetLarge [31], and VGG16 [21]. Every one of the 12 models was pre-trained on ImageNet. The final two, fully-connected (FC) layers are trained on the fingerprint images. Among them, MobileNetV2 is the latest image recognition model, but VGG19 is one of the most commonly used in current state-of-the-art fingerprint liveness detection systems [15]. For this reason, we have attacked the VGG19-based fingerprint recognition system using DeepAttack on the ATVS fingerprint dataset.

Image quality assessment (IQA) parameters are considered by Galbally et al. to identify whether a given fingerprint is genuine or fake. For a given fingerprint image I_f of dimensions (X x Y), a smoothed version I_s is obtained by applying it to a low-pass Gaussian kernel with $\sigma = 0.5$ and a window size of 3 x 3. Pixel-based differences, including AD, MD, NAE, and PSNR are calculated between I_f and I_s for a given fingerprint image, as well as for the real fingerprint (ground truth) image.

3. Proposed Work

The proposed DeepAttack system consists of six major steps: enhancing fingerprints using FingerNet [24], training the fingerprint recognition system using a transfer learning-based approach, finding the discriminative image regions using attributions [30], generating the corresponding error code [17], generating perturbed fingerprints, calculating the quality between the original and perturbed fingerprints using IQA parameters, and finally comparing the minutiae using the VeriFinger SDK . The process flowchart for DeepAttack is shown in Fig. 2

3.1. FingerNet

FingerNet is used for generating an enhanced map and extracting the corresponding minutiae. FingerNet uses nonlinear and non-parametric activation layers on each pixel for normalization, with a gradient-based estimator that is used to compute the ridge orientation. FingerNet utilizes three handcrafted kernels along with a shallow ConvNet to estimate the orientation. FingerNet makes use of a complex Gabor filter that is obtained from local ridge orientation θ_{r} and local ridge frequency ω_{r}. Convolution operations are performed on local fingerprint blocks. The enhanced fingerprint map E_{block} is described as follows:

For every pixel (x_f, y_f) in block I_{block};

$$\begin{aligned} E_{block}(x_f, y_f) &= (I_{block} \times g_{\omega_r\, \theta_r})(x_f, y_f) \\ &= A_{mp}(x_f, y_f) \cdot \exp i\phi_{ph}(x_f, y_f) \end{aligned} \tag{1}$$

where $A_{mp}(x_f, y_f)$ and $\mathrm{i}\phi_{ph}(x_{f0}, y_{f0})$ are the amplitudes and phase angles of the enhanced fingerprint maps, and $\phi_{ph}(x_{f0}, y_{f0})$ is the final enhanced fingerprint map. This result is supplied to the transfer learning-based fingerprint recognition system described in Section 3.2.

3.2. Transfer Learning

We have adopted the VGG19 network for fingerprint recognition on the ATVS fingerprint dataset. During training, the input to the VGG19 architecture is a 224 x 224 pixel image. Filters of size 3 x 3 are used during convolution, and the stride value is set to 1. Padding of 1 pixel is also added for the 3 x 3 convolution layers. The dimensionality is reduced using max-pooling, resulting in a 2 x 2 pixel window with stride 2. The VGG19 network is pre-trained using the ImageNet dataset, and the initial weights of the network are fixed. However, a new FC layer is added with a rectified linear unit (ReLU) as the activation function, and a new softmax layer is also added. These last two newly-added layers are used to train the VGG19 network on the fingerprint images from the ATVS fingerprint dataset.

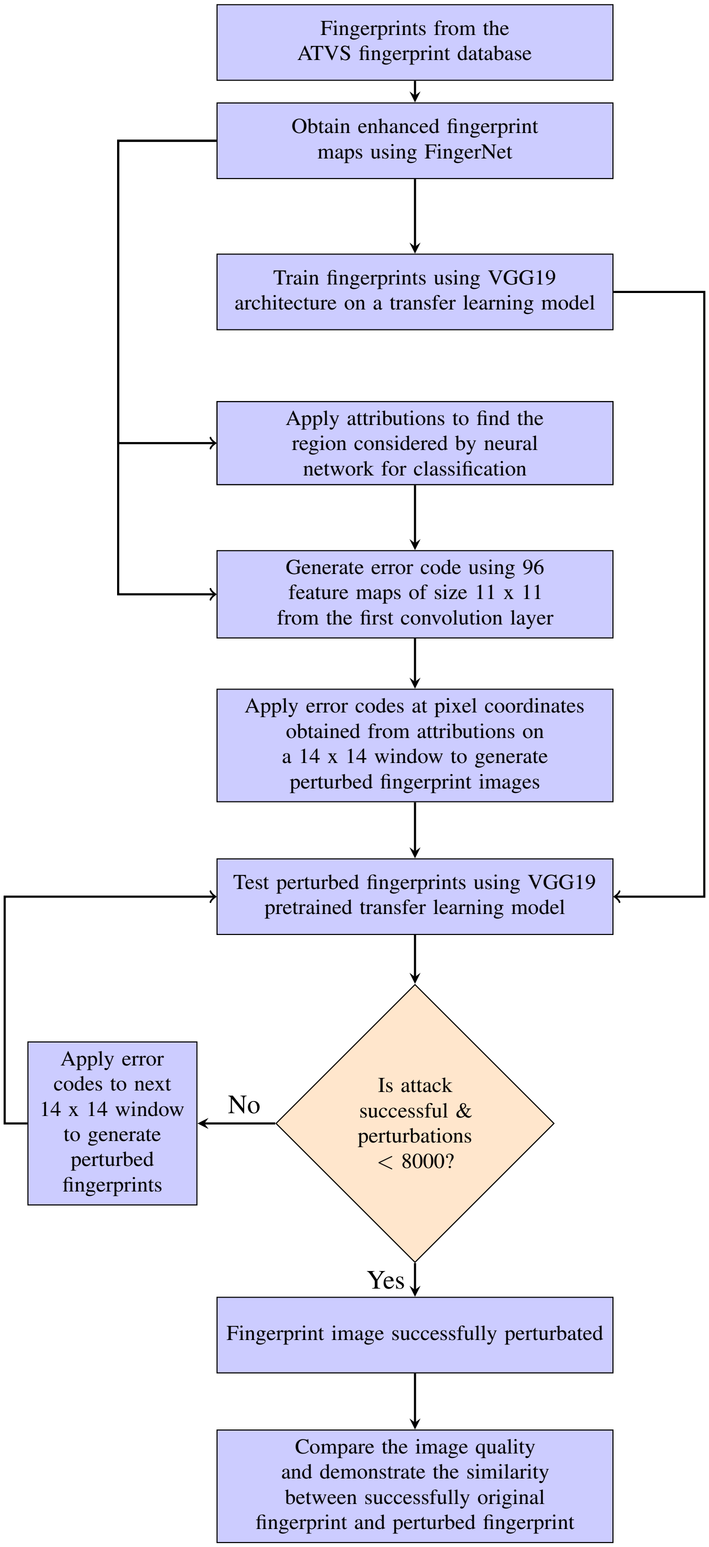

Fingerprints from the ATVS fingerprint database
Obtain enhanced fingerprint maps using FingerNet
Train fingerprints using VGG19 architecture on a transfer learning model
Apply attributions to find the region considered by neural network for classification
Generate error code using 96 feature maps of size 11 x 11 from the first convolution layer
Apply error codes at pixel coordinates obtained from attributions on a 14 x 14 window to generate perturbed fingerprint images
Test perturbed fingerprints using VGG19 pretrained transfer learning model
Is attack successful & perturbations < 8000?
No
Apply error codes to next 14 x 14 window to generate perturbed fingerprints
Yes
Fingerprint image successfully perturbated
Compare the image quality and demonstrate the similarity between successfully original fingerprint and perturbed fingerprint

Figure 2: Flowchart of our proposed DeepAttack system.

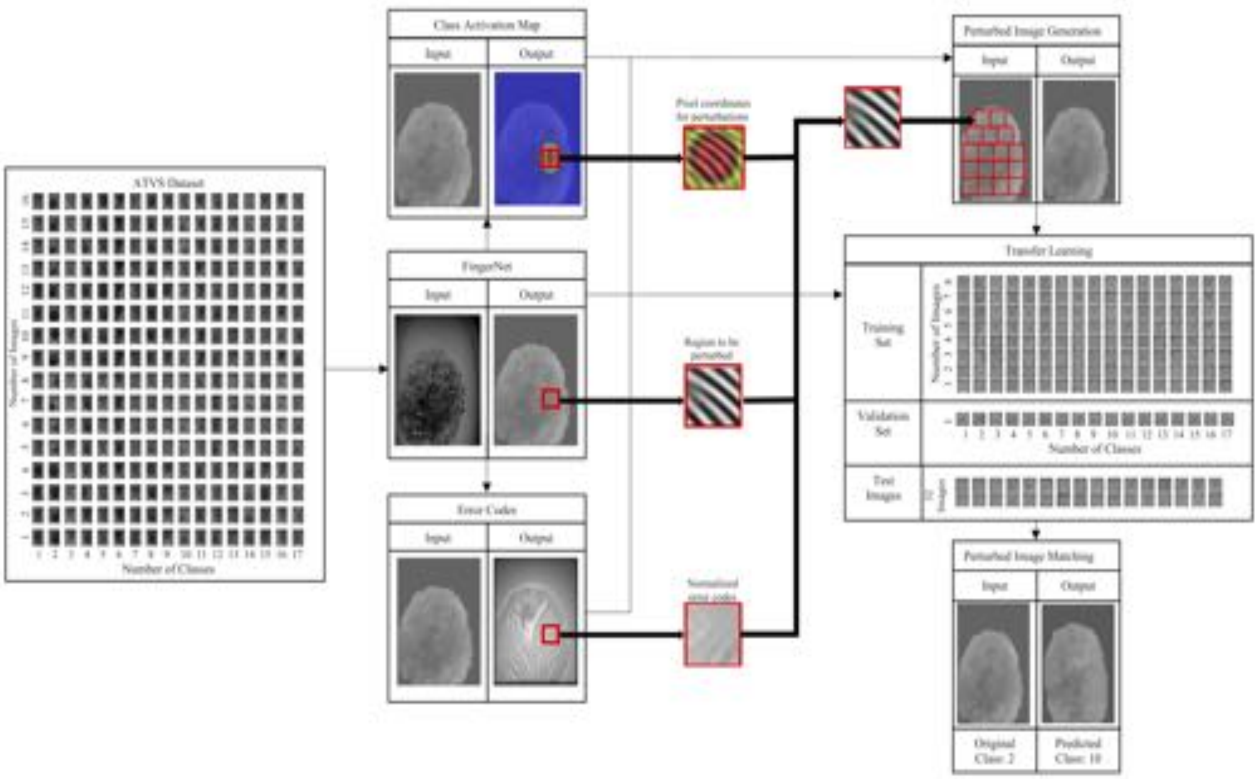


Figure 3: Block diagram of DeepAttack on ATVS fingerprint dataset.

3.3. Attributions

For the FingerNet enhanced image *(x_f, y_f)*, the activation of k units in the last convolutional layer is given by a_k*(x_f, y_f)* at spatial location *(x_f, y_f)*. For each unit *k*, global average pooling is performed and the result obtained A^k is $\sum_{x_f,\, y_f} a_k(x_f,\, y_f)$. For a given class *z*, the output to the softmax P_z is given by $\frac{exp(S_z)}{\sum_z exp(S_z)}$. S_z is the input to the softmax activation function, given by $\sum_k w_k^z A_k$ where w_k^z represents the weights corresponding to class *z* for unit *k*. The importance of the activation function A_k for class z is indicated by the weights w_k^z. A 14 x 14 pixel coordinate window is created to surround the maximum intensity region obtained from attributions.

3.4. Error Code Generation

Error codes are used in the process of generating the perturbed fingerprints. They are built using the features derived from the first layer of CNN [17]. Ninety-six convolutional kernels of size 11 x 11 are used in the first layer. The error model is resized to 224 x 224 pixels and pixel values are extracted from the coordinates obtained during the attributions, as discussed in Section 3.3.

3.5. Perturbed Fingerprint Generation

The pixel values are extracted from the original images and error codes corresponding to the 14 x 14 window obtained from attributions. Each of these pixels are normalized by dividing the value by 255. The result obtained after division is subtracted from the original fingerprints to generate the kernel. This kernel is slid across the entire image to create the set of perturbed images. The perturbed fingerprint is then used to launch an attack against the VGG19-based transfer learning fingerprint recognition system. If the attack is unsuccessful, the perturbation process is repeated. The maximum number of perturbations is set to 8000 to avoid degradation of the image quality.

3.6. Fingerprint Image Quality Assessment and Minutiae Detection

The IQA parameters presented by Galbally et al., namely AD, MD, NAE and PSNR, are used to determine the quality of the original and perturbed images. The differences between them is calculated for instances in the ATVS fingerprint dataset. The VeriFinger SDK is used to detect minutiae and perform one-to-one matching between the original fingerprints and perturbed fingerprints. The block diagram for DeepAttack is shown in Fig. 3.

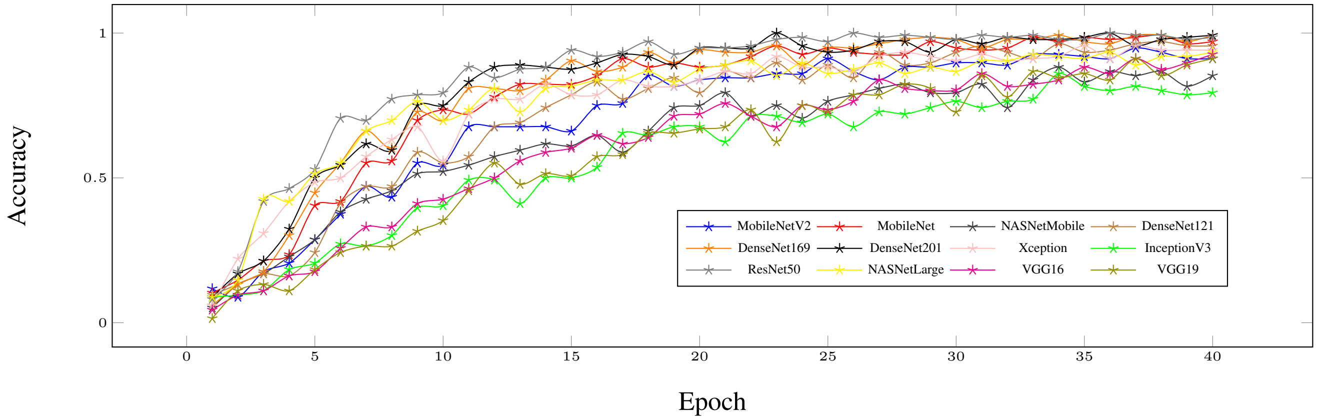


Figure 4: Training accuracy for 12 state-of-the-art image recognition models on the ATVS fingerprint dataset.

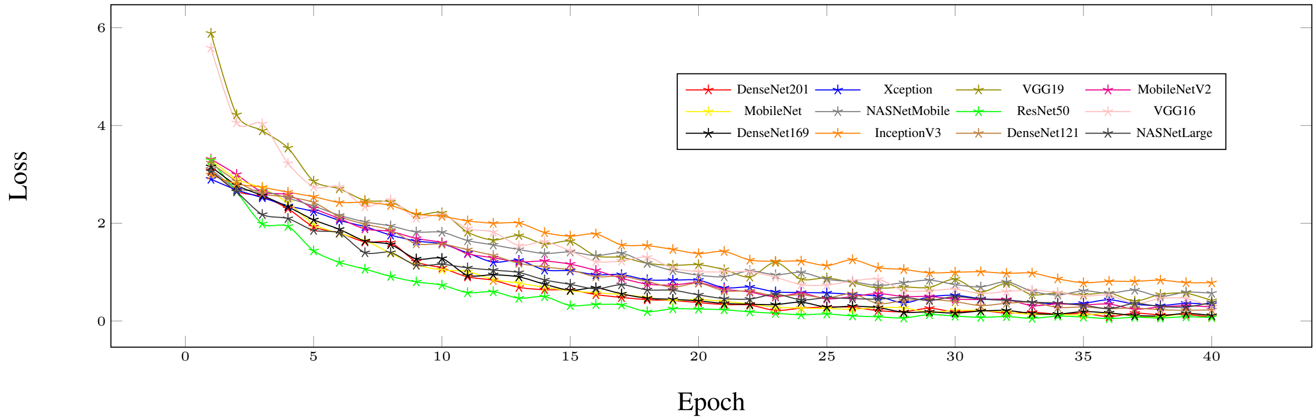


Figure 5: Training loss for 12 state-of-the-art image recognition models on the ATVS fingerprint dataset.

4. Experimental Results

We have selected 272 fingerprints from the ATVS fingerprint dataset. These include data from 17 subjects, 4 fingers (right and left first/second finger) with 4 samples per finger. The fingerprint images are enhanced using FingerNet.

4.1. Enhancing Fingerprints using FingerNet

FingerNet is pre-trained using 8000 pairs of fingerprints, using a private dataset named CISL24218 with 512 x 512 pixels and 500 PPI. In our process, all of the 272 fingerprints from the ATVS fingerprint dataset are supplied to FingerNet for enhancement. The enhanced images are resized to 224 x 224 pixels and split into sets for training, validation, and testing. The training set contains 136 fingerprint images; this includes 2 fingers (right and left first/second finger), 4 samples per finger for 17 subjects. The validation set contains 17 fingerprints; this includes 1 finger (right second finger), 1 sample per finger for 17 subjects. Thirty-two randomly selected fingerprints of the left first finger are tested using transfer learning.

4.2. Transfer Learning for Classifying Fingerprints

The training data in our experiments is augmented by random vertical and horizontal flipping, according to the following parameters: rotation = 20.0, zoom = 0.2, shear = 0.2, and dropout = 1e-3. For the ATVS fingerprint dataset, the network is trained for 40 epochs, with a batch size of 17. The model is compiled using an Adam optimizer, with categorical cross entropy as the loss function and accuracy as the metric. The training and validation accuracies and losses for each of the 12 image recognition models are pre-trained on ImageNet. The last two FC layers are trained on the ATVS fingerprint dataset. The twelve image recognition models are MobileNetV2, MobileNet, NASNetMobile, DenseNet121, DenseNet169, DenseNet201, Xception, InceptionV3, ResNet50, NASNetLarge, VGG16, and VGG19. The training accuracies and training losses on the ATVS fingerprint dataset for these models are shown in Fig. 4 and Fig. 5, respectively. The weights used to train each of the models, along with the results obtained for each one, are available at: `http://bit.ly/2UNHu3n`.

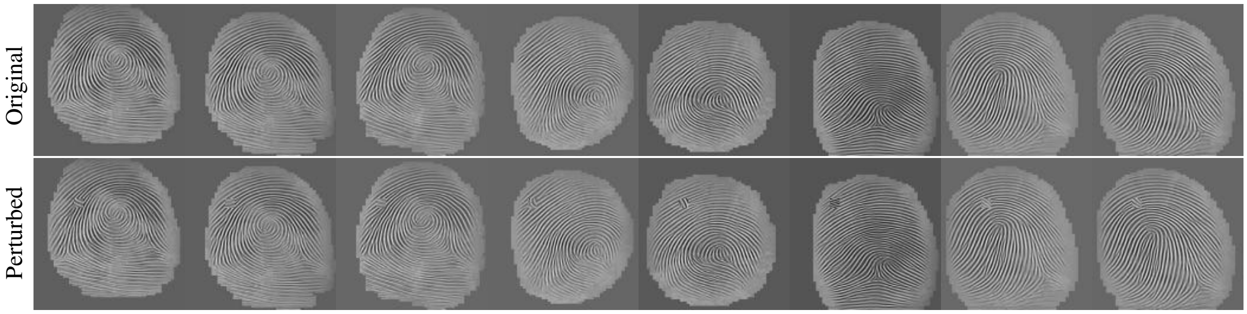


Figure 6: Top: Original FingerNet enhanced fingerprints. Bottom: Perturbed fingerprints from DeepAttack

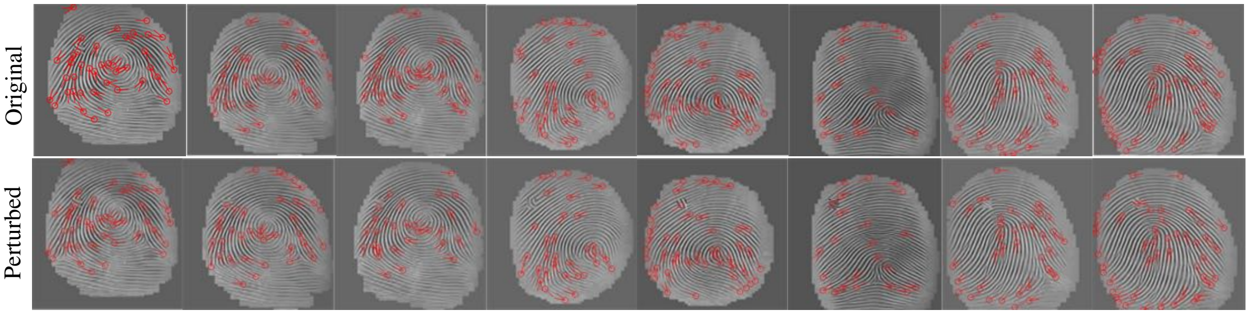


Figure 7: Top: Minutiae of original fingerprints. Bottom: Minutiae of perturbed fingerprints

4.3. Obtaining Attributions

The class activation map highlights the discriminative fingerprint region used by the CNN. A 14 x 14 pixel coordinate window is created to surround the discriminative fingerprint region.

4.4. Error Model Generation

Error codes are build using 96 convolutional kernels, sized 11 x 11, from the first convolutional layer. The pixel coordinates obtained in Section 4.3 are considered, and the corresponding pixel values in both the original image and error model are obtained. These are then saved into two separate matrices. The values of the error matrix are normalized by dividing by 255. The pixel coordinates of the original image are subtracted by the normalized error matrix to obtain a kernel with a size of 14 x 14. This kernel is then slid across the entire image to create a set of perturbed images.

4.5. Applying Perturbed Fingerprints to Transfer Learning-based Fingerprint Recognition

Perturbed fingerprint images are used to attack the VGG19-based fingerprint recognition system. In this research, all of the test images were subject to attack. The total number of perturbations is 26,720. The execution time was approximately 0.4 seconds per perturbation on the NVIDIA Tesla V100, having 96 cores and 16GB RAM.

IQA parameters	Largest difference between original & perturbed images
Average difference (AD)	0.004822
Maximum difference (MD)	0.538818
Normalized absolute error (NAE)	0.003267
Peak signal to noise ratio (PSNR)	0.39276

Table 1: Largest difference between original and perturbed fingerprints on ATVS fingerprint dataset

4.6. Fingerprint Image Quality Assessment and Minutiae Detection

The original enhanced fingerprint, obtained from FingerNet, and the perturbed images that were generated using DeepAttack, are shown in Fig. 6. IQA parameters AD, MD, NAE, and PSNR are calculated for both sets [7]. The largest difference between each is tabulated in Table 1. The original and perturbed fingerprints are resized to 128 x 128 pixels, with a resolution of 300 dpi, and supplied to the VeriFinger SDK. Figure 7 shows the minutiae detected from both original and perturbed fingerprints. One-to-many matching is performed between them. For the ATVS fingerprint dataset, a matching score greater than 250 was obtained for all of the perturbed fingerprints when compared to the originals in the same class. This score is indicative of a high degree of similarity between them.

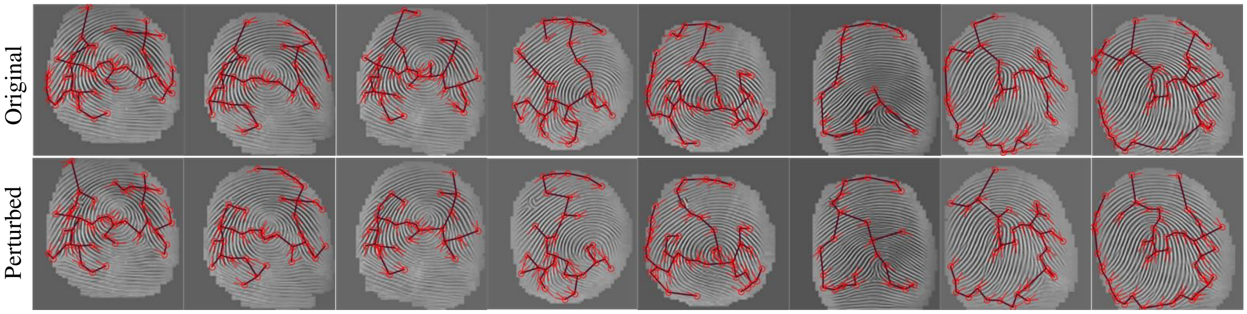


Figure 8: Top: One-to-one matching of original fingerprints. Bottom: One-to-one matching of perturbed fingerprints

5. Conclusion

In this paper, we attack the VGG19-based fingerprint recognition system. This system is used in current, state-of-the-art fingerprint liveness detection. Our approach involves perturbing the discriminative fingerprint region that is used by the CNN for classification. A successful attack indicates that the perturbed fingerprint image has been misclassified. The quality of the perturbed image is analyzed by calculating the IQA values AD, MD, NAE, and PSNR. For the ATVS fingerprint dataset, the results obtained are 0.004822, 0.538818, 0.003267, and 0.39276, respectively. The differences in IQA values between the original and corresponding perturbed fingerprint images are negligible, indicating that there is no significant loss in terms of image quality using DeepAttack.

A second comparative analysis is performed using the VeriFinger SDK. This operates on the minutiae that are extracted using VeriFinger. A one-to-many matching score greater than 250 is obtained when the perturbed fingerprints are compared with the originals. This score is indicative of a high degree of similarity between them. The DeepAttack process is summarized as follows:

(i) Generate perturbed fingerprint images by subtracting the normalized error codes from the original fingerprints. This is performed using pixel coordinates that correspond to a 14 x 14 window contained within the region of maximum intensity, and obtained from the attributions.

(ii) Launch a successful attack on the transfer learning approach, which is also used in current state-of-the-art fingerprint liveness detection systems.

(iii) Compare the image quality of the perturbed fingerprints with original fingerprints by computing the IQA parameters: AD, MD, NAE, and PSNR.

(iv) Detect the minutiae and perform matching between the original and perturbed fingerprints using the VeriFinger SDK.

From the perspective of a human visual recognition system, the perturbed version looks virtually identical to the original. The one-to-one matching results obtained using VeriFinger is shown in Fig. 8. The weights used to train the 12 state-of-the-art image recognition models on the ATVS fingerprint dataset, along with the results obtained for each model, and the results obtained for FingerNet, attributions, error codes, IQA values for AD, MD, NAE, PSNR are available at: `http://bit.ly/2UNHu3n`.

6. Future Work

The results generated by DeepAttack are significant, and the authors agree that this topic warrants further study. Our plan for future work includes:

(i) Attacking fingerprints captured without cooperation from the subject. The ability to circumvent a state-of-the-art fingerprint identification system, without cooperation from the subject, represents a very serious breach in security. It implies that access can be gained without the subject's direct knowledge.

(ii) Attacking a fingerprint recognition system using a 3D-printed replication of a latent fingerprint. Success in this experiment would demand significant attention from the security community because the subject does not have to be present during access.

Combined, successfully completing these two experiments would render the current system of fingerprint recognition-based security practically obsolete. It is our intention to determine whether another, more secure alternative to password-based security is required.

Acknowledgements. We acknowledge support from NSF Awards #1822976 and #1422257, an award from the Florida Cybersecurity Center, Royal Bank of Canada, and the Air Force Young Investigator Award to Sumit Jha.

References

[1] VeriFinger. http://www.neurotechnology.com/verifinger.html.

[2] and T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg. Matchnet: Unifying feature and metric learning for patch-based matching. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3279–3286, June 2015.

[3] M. Brown and D. G. Lowe. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, 74(1):59–73, Aug 2007.

[4] F. Chollet. Xception: Deep learning with depthwise separable convolutions. *CoRR*, abs/1610.02357, 2016.

[5] R. Frassetto Nogueira, R. de Alencar Lotufo, and R. Campos Machado. Evaluating software-based fingerprint liveness detection using convolutional networks and local binary patterns. In *2014 IEEE Workshop on Biometric Measurements and Systems for Security and Medical Applications (BIOMS) Proceedings*, pages 22–29, Oct 2014.

[6] J. Galbally, F. Alonso-Fernandez, J. Fierrez, and J. Ortega-Garcia. A high performance fingerprint liveness detection method based on quality related features. *Future Generation Computer Systems*, 28(1):311 – 321, 2012.

[7] J. Galbally, S. Marcel, and J. Fierrez. Image quality assessment for fake biometric detection: Application to iris, fingerprint, and face recognition. *IEEE Transactions on Image Processing*, 23(2):710–724, Feb 2014.

[8] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017.

[9] G. Huang, Z. Liu, and K. Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016.

[10] X. Jia, X. Yang, K. Cao, Y. Zang, N. Zhang, R. Dai, X. Zhu, and J. Tian. Multi-scale local binary pattern with filters for spoof fingerprint detection. *Information Sciences*, 268:91 – 102, 2014. New Sensing and Processing Technologies for Hand-based Biometrics Authentication.

[11] A. K Jain and J. Feng. Latent fingerprint matching. *IEEE transactions on pattern analysis and machine intelligence*, 33:88–100, 01 2011.

[12] K. Ko. User's guide to nist biometric image software (nbis). Technical report, 2007.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[14] J. Li, J. Feng, and C.-C. J. Kuo. Deep convolutional neural network for latent fingerprint enhancement. *Signal Processing: Image Communication*, 60:52 – 63, 2018.

[15] E. Marasco, S. Cando, and L. Tang. Can liveness be automatically detected from latent fingerprints? In *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pages 93–99. IEEE, 2019.

[16] R. F. Nogueira, R. de Alencar Lotufo, and R. C. Machado. Fingerprint liveness detection using convolutional neural networks. *IEEE Transactions on Information Forensics and Security*, 11(6):1206–1213, June 2016.

[17] S. Raj, S. K. Jha, L. L. Pullum, and A. Ramanathan. Statistical hypothesis testing using cnn features for synthesis of adversarial counterexamples to human and object detection vision systems. 5 2017.

[18] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[19] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. *CoRR*, abs/1801.04381, 2018.

[20] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 519–528, June 2006.

[21] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[22] S. A. Sudiro, M. Paindavoine, and T. M. Kusuma. Simple fingerprint minutiae extraction algorithm using crossing number on valley structure. In *2007 IEEE Workshop on Automatic Identification Advanced Technologies*, pages 41–44, June 2007.

[23] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.

[24] Y. Tang, F. Gao, J. Feng, and Y. Liu. Fingernet: An unified deep network for fingerprint minutiae extraction. *CoRR*, abs/1709.02228, 2017.

[25] S. Xie, R. B. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. *CoRR*, abs/1611.05431, 2016.

[26] D. Yambay, L. Ghiani, P. Denti, G. L. Marcialis, F. Roli, and S. Schuckers. Livdet 2011 fingerprint liveness detection competition 2011. In *2012 5th IAPR International Conference on Biometrics (ICB)*, pages 208–215, March 2012.

[27] L. Yan, Y. Wang, T. Song, and Z. Yin. An incremental intelligent object recognition system based on deep learning. In *2017 Chinese Automation Congress (CAC)*, pages 7135–7138, Oct 2017.

[28] S. Yoon and A. K. Jain. Longitudinal study of fingerprint recognition. *Proceedings of the National Academy of Sciences*, 112(28):8555–8560, 2015.

[29] C. Yuan, Z. Xia, L. Jiang, Y. Cao, Q. M. Jonathan Wu, and X. Sun. Fingerprint liveness detection using an improved cnn with image scale equalization. *IEEE Access*, 7:26953–26966, 2019.

[30] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. *CoRR*, abs/1512.04150, 2015.

[31] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le. Learning transferable architectures for scalable image recognition. *CoRR*, abs/1707.07012, 2017.