GENERATING ADVERSARIAL EXAMPLES BY MAKEUP ATTACKS ON FACE RECOGNITION

Zheng-An Zhu *Yun-Zhong Lu* *Chen-Kuo Chiang*

Department of Computer Science and Information Engineering, National Chung Cheng University, Taiwan

ABSTRACT

Deep Learning models have been developed rapidly and achieved great success in computer vision and natural language processing. In this paper, we propose to generate adversarial examples to attack well-trained face recognition models by applying makeup effect to face images. It consists of two generative adversarial networks (GANs) based sub-networks, *Makeup Transfer Sub-network* and *Adversarial Attack Sub-network*. Makeup Transfer Sub-network transfers the non-makeup face images to makeup faces. Adversarial Attack Sub-networks hides attack information within makeup effect. The generated face images make the well-trained face recognition models misclassified as dodge attack or target attack. The experimental results demonstrate that our method can generate high-quality face makeup images and achieve higher error rates on various face recognition models compared to the existing attack methods.

***Index Terms*—** Adversarial example attack, generative adversarial networks, deep neural networks, face recognition

1. INTRODUCTION

Recent neural network models are proven to be powerful in many applications. This brings more and more attack methods which generate adversarial examples to decrease the recognition accuracy of deep neural models. Szegedy *et al.* [1] first discover the weaknesses in DNNs. They generate an adversarial example which is similar to the original image yet misleads DNNs to incorrect classification result. Then, FGSM algorithm [2] adds a small amount of noises to the gradient of the input image to affect the classification results of neural network models. [3] proposes the Projected Gradient Descent Method (Iterative FGSM) to improve the original FGSM results. In small region attack, [4] proposed the Adversarial Patch method to create a patch that can be applied to any image and make the target network misclassified. This algorithm is highly similar to the Expectation over Transformation [5], which creates a patch that can be applied to any position of the original image to attack the model. Another track of adversarial attack methods is to combines Generative Adversarial Networks [6] to generate adversarial examples.

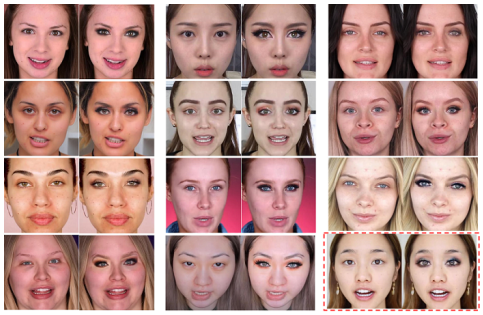

Fig. 1. Sample images of attack results. The photos with makeup effect by our method will be recognized by the well-trained face recognition model as the category of subject in the dotted-line box.

Face recognition is an important topic in image processing fields. Face recognition models can be attacked by exaggerated wearing or facial accessories to dodge the correct identity [7]. In [8], a special glasses-shaped patch is placed to a fixed position on face image to attack the face recognition models. Even though their attack rates are improved, the round-shaped or glasses-shaped patches makes the face image looks unreal. In this paper, we propose a new attack method to attack face recognition model in a more natural way, called makeup attack. By applying makeup to the eye regions that can fool the face recognition models. In Fig. 1, each pair of faces depicts the original face image and the image after applying makeup effect by our method. All face images with makeup are recognized by the well-trained face recognition model as the category of subject in the dotted-line box.

In order to generate realistic makeup effect, we propose to train generative adversarial networks (GANs) [9] to generate makeup photos. Recently, the success of style transfer based on GANs to transfer two image domains [10] inspires us to transfer non-makeup photos to makeup ones. The goal of our method is to hide the attack information in the makeup photo. Adversarial attack methods can be classified into two categories: white-box attack and black-box attack. White-box attack means that the attacker knows most of the architecture information. For black-box attack, the attacker does not know too much about architecture but still can produce examples

with perturbation noises. In this paper, we propose a novel makeup attack as white-box attack to transfer non-makeup images to makeup images where the perturbation information of the attack is hidden in the makeup areas.

The contributions of our method are listed as follows. Firstly, to attack the face recognition models, we provide a novel idea to hide the attack information by the makeup effect. This makes the face images look more natural compared to the previous patch-based methods. In addition, these adversarial noises are undetectable to human. Secondly, we propose new architecture with two new subnetworks, Makeup Transfer Sub-network and Adversarial Attack Sub-network, in a white-box attack mannar. Lastly, we collected a new high-resolution makeup and non-makeup face dataset from YouTube. The experimental results demonstrate the superior performance of our attack method.

2. PROPOSED METHOD

In this section, we describe our method to attack face recognition models. The proposed method consists of two subnetworks. *Makeup Transfer Sub-network* is proposed based on CycleGan [10] to transfer makeup photos to non-makeup photos. *Adversarial Attack Sub-network* generates adversarial examples that can attack target networks, while the discriminator ensures that photos carrying attack information still similar to real makeup photos. The system framework is depicted in Fig. 2. To minimize the amount of perturbation noises, the makeup effect is applied only to eye regions.

2.1. Makeup Transfer Sub-network

A Makeup Transfer Sub-network is proposed to transfer face images in non-makeup domain to makeup domain. Following the setting of CycleGANs[10], two generators G and F are exploited to produce makeup and non-makeup images, respectively. Specifically, G is to add makeup effect to non-makeup images. F is to remove makeup effect while still maintaining the original identity. Two discriminators D_x and D_y is utilized. D_x is used to distinguish between real non-makeup photos and generated non-makeup photos. D_y, similar to D_x, is used to distinguish between real makeup photos and generated makeup photos. The 70×70 PatchGANs[11] is used as our discriminator.

Adversarial loss is first applied to the network:

$$L_{GAN}(G, F, D_x, D_y) = -(\mathbb{E}_{x\sim P_x, y\sim P_y}[\log D_y(y) + \log D_x(x) + \log(1 - D_y(G(x))) + \log(1 - D_x(F(y)))]) \quad (1)$$

where x is the real non-makeup input, y is the real with-makeup input, and the network G generates results $G(x)$ that obfuscate discriminator D_y. Network $F : Y \rightarrow X$ generates results of non-makeup faces $F(y)$ which also obfuscator D_x.

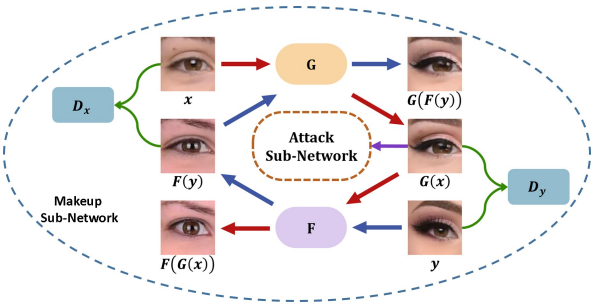

Fig. 2. System framework for makeup attack on face recognition models, including two subnetworks, Makeup Transfer Sub-network and Adversarial Attack Sub-networks.

To help GAN training and fast convergence, an additional term is added into the objective function using the gradient penalty [12], instead of the gradient clipping to keep training more stable:

$$L_{penalty}(D_x, D_y) = \mathbb{E}_{\hat{x}\sim P_{\hat{x}}, \hat{y}\sim P_{\hat{y}}}[(\|\nabla D_x(\hat{x})\|_2 - 1)^2 + (\|\nabla D_y(\hat{y})\|_2 - 1)^2] \quad (2)$$

where $\hat{x}$ is a random interpolation sample between the real non-makeup photo and the generated non-makeup image.

An identity term is proposed, if makeup photos are used as input into the network G, the output should maintain makeup style, even it looks the same as the original input image. The identity term is defined for the network G and F as:

$$L_{identity}(G, F) = \mathbb{E}_{x\sim P_x, y\sim P_y}[\,\|F(x) - x\|_1 + \|G(y) - y\|_1] \quad (3)$$

In addition, when using the output $G(x)$ of the network G as the input to the network F, the output result $F(G(x))$ of network F should be consistent with the original input x. Therefore, the cycle loss can be defined for duel paths (non-makeup to makeup and makeup to non-makeup) as:

$$L_{cycle}(G, F) = \mathbb{E}_{x\sim P_x, y\sim P_y}[\,\|F(G(x)) - x\|_1 + \|G(F(y)) - y\|_1] \quad (4)$$

To ensure the makeup effect is balanced for both eyes, an addition symmetry loss is proposed. During training, the eye images are used as input to the network in pairs (left and right eyes). L_1 loss is exploited to calculate the distance between left and right eyes after applying makeup effect and makeup removal. The symmetry loss can be defined as:

$$L_{sym}(G, F) = \mathbb{E}_{x\sim P_x}[\|G(x)_l - G(x)_r\|_1] + \mathbb{E}_{y\sim P_y}[\|F(y)_l - F(y)_r\|_1] \quad (5)$$

where $G(x)_l$ and $G(x)_r$ represents generated results of left and right eye, respectively.

A regularization loss is included to control the model output $G(x)$ with makeup with the consideration that the makeup effect should not be overly overlayed to change the subject's identity. On the contrary, since network F is to remove the makeup, we do not apply regularization loss to network F. It is defined as:

$$L_{reg}(G) = \mathbb{E}_{x\sim P_x}[\|G(x) - x\|_1] \tag{6}$$

Considering above loss terms, the objective function for Makeup Transfer Sub-network is defined as:

$$\begin{aligned} L_{Makeup} =& \lambda_G \cdot L_{GAN} + \lambda_P \cdot L_{penalty} + \lambda_I \cdot L_{identity} \\ &+ \lambda_C \cdot L_{cycle} + \lambda_S \cdot L_{sym} + \lambda_R \cdot L_{reg} \end{aligned} \tag{7}$$

where $\lambda_I, \lambda_P, \lambda_S, \lambda_G, \lambda_C, \lambda_R$ are controlling weights to balance the multiple terms in the objectives function.

2.2. Adversarial Attack Sub-network

Adversarial Attack Sub-networks is designed as GANs for generating adversarial examples. The eye regions with makeup $G(x)$ (output of Makeup Transfer Sub-network) are first blended with the original non-makeup images x by the transformation function T. Then, it serves as the input to the generator H. H aims to generate output image $H(T(x, G(x)))$ with perturbation noises that can deceive both the target network A and the discriminator D_h. Target network A is well-trained face recognition model to be attacked by the adversarial examples. Weights of model A are fixed all the time. It aims to generate adversarial examples to make the target network A misclassified. The discriminator D_h is to ensure the generated image to remain in makeup style. Therefore, real makeup face photos are also used as input to the discriminator with a mask to retain using eye regions only. The discriminator D_h is the same as the discriminator D_y. To train D_h, D_y is used as pre-train weights to initialize D_h. The network is depicted in Fig. 3.

The adversarial loss for the GAN can be defined as:

$$\begin{aligned} L_{target\_GAN}(H, D_h) = -(&\mathbb{E}_{y\sim P_y}[\log D_h(y))] \\ &+ \mathbb{E}_{x\sim P_x}[\log(1 - D_h(H(\tilde{x})))]) \end{aligned} \tag{8}$$

$$\tilde{x} = T(x, G(x)) \tag{9}$$

where y is real makeup face images.

The output of the generator H is used as input to the target network A. We hope the parameters of the generator H are updated so that the generated image of H can increase the classification probability of the target category. While maximizing the probability of target category, the maximum probability of non-target category should be reduced to boost the model discrimination. In view of this, the classification loss is defined as follows.

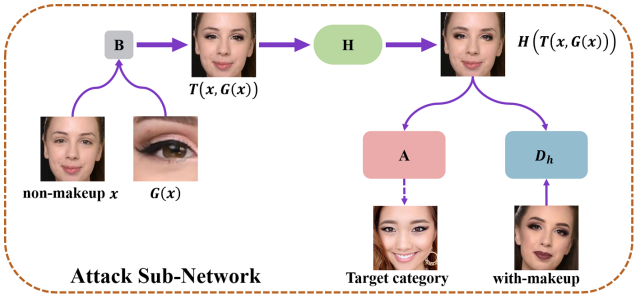

Fig. 3. Proposed Adversarial Attack Sub-networks.

$$\begin{aligned} L_{class}(H) = \mathbb{E}_{x\sim P_x}[&-\log A(H(\tilde{x}))_{target} \\ &+ \log \max_{i\neq target} A(H(\tilde{x}))_i] \end{aligned} \tag{10}$$

Lastly, to limit the amount of variation of the perturbation, we have added L_2 hinge loss for the network output:

$$L_{hinge}(H) = \mathbb{E}_{x\sim P_x}[\|H(\tilde{x})\|_2] \tag{11}$$

With the above loss terms, the objective function of Adversarial Attack Sub-networks is defined as:

$$L_{Attack} = \lambda_A \cdot L_{class} + \lambda_T \cdot L_{target\_GAN} + \lambda_H \cdot L_{hinge} \tag{12}$$

where λ_A, λ_T and λ_H are the weights to balance the multiple terms in this objectives function. Finally, we can obtain the full objective function for the proposed makeup attack model:

$$L_{Total} = L_{Makeup} + L_{Attack} \tag{13}$$

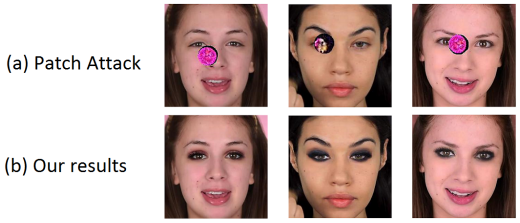

Fig. 4. Sample results of Patch Attack and our makeup attack.

3. EXPERIMENTAL RESULTS

3.1. Data Collection

Since most existing face recognition datasets do not provide high-resolution images and lack the labels of makeup/non-makeup, we collect a high resolution Makeup Face Dataset

Table 1. Face recognition accuracy of DNN-based models.

Accuracy(%)	Train	Test	Test Loss
AlexNet [13]	100.0	98.00	0.0414
SqueezeNet [14]	100.0	96.00	0.1189
VGG16 [15]	100.0	100.0	0.0029
ResNet50 [16]	100.0	100.0	0.0013
InceptionV3 [17]	100.0	100.0	0.0021
DenseNet121 [18]	100.0	100.0	0.0011
LightCNN29 [19]	100.0	98.50	0.2309

Table 2. Model error rates(%) of target attack and dodge attack by our method.

Error Rate(%)	Before	Target Attack	Dodge Attack
AlexNet [13]	2.0	98.50	100.0
SqueezeNet [14]	4.0	95.25	100.0
VGG16 [15]	0.0	100.0	100.0
ResNet50 [16]	0.0	90.00	97.50
InceptionV3 [17]	0.0	97.50	100.0
DenseNet121 [18]	0.0	95.50	100.0
LightCNN29 [19]	1.5	93.75	99.00

that is labeled by makeup and non-makeup. We searched for the channels of makeup tutorial videos on YouTube. For each channel, we screenshot 50 to 100 makeup and non-makeup images from videos. It means that each channel is a category of face class. Finally, we collected 632 face images without makeup and 745 images with-makeup. We flipped the face horizontally for data augmentation to get a total of 2754 face images of 18 subjects. To minimize the variations, we use only eye regions to hide attack information. By applying face component segmentation, we get 1020 without-makeup eyes images and 1070 with-makeup eyes images for the eye dataset of 2090 images in total.

3.2. Comparison of Makeup Attack on Face Models

We first train multiple DNN-based face recognition models. ImageNet is used as pre-train dataset and Makeup Face Dataset is used to fine-tune these models. The recognition accuracy are presented in Table 1. We can see that all these classification models achieve high face recognition accuracy.

The target attack results and the dodge attack results are shown in Table 2. By our attack method, the error rates of these face recognition models increase above 90%.

3.3. Comparison of FGSM and PGD

In Table 3, we compare the model error rates by our method to FGSM [2] and PGD [20]. Because both FGSM and PGD methods generate perturbation noises in the entire image, we apply a mask to attack eye regions only. We can note that our method increases the error rate higher than other methods.

Table 3. Model error rates(%) attacked by our method, FGSM and PGD.

Error Rate(%)	Before	FGSM [2]	PGD [20]	Ours
AlexNet [13]	2.0	2.0	1.29	98.50
SqueezeNet [14]	4.0	4.0	13.44	95.25
VGG16 [15]	0.0	0.0	85.56	100.0
ResNet50 [16]	0.0	0.0	41.34	90.00
InceptionV3 [17]	0.0	0.0	27.91	97.50
DenseNet121 [18]	0.0	0.0	31.26	95.50
LightCNN29 [19]	1.5	1.5	2.33	93.75
Average	1.07	1.07	29.02	95.78

Table 4. Model error rates(%) of our method and Patch Attack.

Error Rate(%)	Before	Patch Attack [21]	Ours
AlexNet [13]	2.0	87.58	98.50
SqueezeNet [14]	4.0	98.34	95.25
VGG16 [15]	0.0	98.76	100.0
ResNet50 [16]	0.0	98.34	90.00
InceptionV3 [17]	0.0	88.61	97.50
DenseNet121 [18]	0.0	92.34	95.50
LightCNN29 [19]	1.5	97.92	93.75
Average	1.07	94.56	95.78

3.4. Comparison of Patch Attack

In Table 4, we compare our results with Patch Attack. We set the patch size of the Patch Attack [21] method to 3% of the image, and limit the patch location only in the face area. In the experimental results, the Patch Attack method has higher error rate than our method in multiple recognition models. This is because the Patch Attack method does not limit the perturbation L_2 distance, thus introducing unnatural patches on face images, as depicted in Fig. 4.

4. CONCLUSION

A Generative Adversarial Networks based attack method is proposed by generating makeup images from non-makeup images where the attack information is hidden in the eye regions by makeup effect. This makes the attack information not to be perceived and makes the face image more natural than existing patch-based attack models. It also outperforms the conventional FGSM and PDG methods where attack information exists in the entire image. Since the makeup effect is applied only to eye regions, it can be extended to holistic face in the future.

5. REFERENCES

[1] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus, "Intriguing properties of neural networks," *CoRR*, vol. abs/1312.6199, 2013.

[2] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy, "Explaining and harnessing adversarial examples," *CoRR*, vol. abs/1412.6572, 2014.

[3] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio, "Adversarial examples in the physical world," *CoRR*, vol. abs/1607.02533, 2016.

[4] Tom B. Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer, "Adversarial patch," *CoRR*, vol. abs/1712.09665, 2017.

[5] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok, "Synthesizing robust adversarial examples," in *ICML*. 2018, vol. 80 of *JMLR Workshop and Conference Proceedings*, pp. 284–293, JMLR.org.

[6] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song, "Generating adversarial examples with adversarial networks," in *IJCAI*. 2018, pp. 3905–3911, ijcai.org.

[7] A. Harvey, "Cv dazzle: Camouflage from face detection," *Master's thesis, New York University*, 2010.

[8] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," in *ACM Conference on Computer and Communications Security*. 2016, pp. 1528–1540, ACM.

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., pp. 2672–2680. Curran Associates, Inc., 2014.

[10] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *ICCV*. 2017, pp. 2242–2251, IEEE Computer Society.

[11] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros, "Image-to-image translation with conditional adversarial networks," in *CVPR*. 2017, pp. 5967–5976, IEEE Computer Society.

[12] Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville, "Improved training of wasserstein gans," in *NIPS*, 2017, pp. 5769–5779.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., pp. 1097–1105. Curran Associates, Inc., 2012.

[14] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[15] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[17] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[18] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 2261–2269.

[19] Xiang Wu, Ran He, Zhenan Sun, and Tieniu Tan, "A light cnn for deep face representation with noisy labels," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2884–2896, 2018.

[20] Alexey Kurakin, Ian Goodfellow, and Samy Bengio, "Adversarial machine learning at scale," *arXiv preprint arXiv:1611.01236*, 2016.

[21] Danny Karmon, Daniel Zoran, and Yoav Goldberg, "Lavan: Localized and visible adversarial noise," in *ICML*. 2018, vol. 80 of *JMLR Workshop and Conference Proceedings*, pp. 2512–2520, JMLR.org.