Face Friend-Safe Adversarial Example on Face Recognition System

Hyun Kwon
School of Computing
Korea Advanced Institute of Science and Technology
Daejeon, South Korea
Email: khkh@kaist.ac.kr

Ohmin Kwon
School of Computing
Korea Advanced Institute of Science and Technology
Daejeon, South Korea
Email: ohmin59@kaist.ac.kr

Hyunsoo Yoon
School of Computing
Korea Advanced Institute of Science and Technology
Daejeon, South Korea
Email: hyoon@kaist.ac.kr

Ki-Woong Park*
Computer & Information Security
Sejong University
Seoul, South Korea
Email: woongbak@sejong.ac.kr
*Corresponding author

***Abstract*—Deep neural networks (DNNs) provide the excellent service on deep learning tasks such as image recognition, speech recognition, and pattern recognition. In the field of face recognition, researches using DNN have been carried out. However, face adversarial example is a serious threat in face recognition system. Face adversarial example adding a little of noise to the original face image can cause the misrecognition in the face recognition system. For example, an attacker intentionally modifies a face image with small distortion, which could cause the face recognition system to misidentify another person. It also shows the possibility of wrong recognition by another person when it is modulated by several points on the face. However, the concept of face friend-safe adversarial example can be useful in a military situation where friend and enemy forces are mixed. In the face recognition field, face friend-safe adversarial example may be needed that is correctly recognized by a friend face recognition system and misidentified by an enemy face recognition system. In this paper, we propose a face friend-safe adversarial example targeting the FaceNet face recognition system. The proposed scheme generates a face friend-safe adversarial example that is misrecognized by a enemy face recognition system but is correctly recognized by friend face recognition system with minimum distortion. For experiment, we used VGGFace2 and Labeled Faces in the Wild (LFW) as a dataset and Tensorflow as a machine learning library. Experimental results show that the proposed method has a 92.2% attack succes rate and 91.4% friend accuracy with only 64.22 distortion.**

***Index Terms*—Face recognition system, machine learning, deep neural network, adversarial example.**

I. Introduction

As computer technology and mass data collection become possible, services using deep learning are attracting attention. Especially, deep neural networks (DNNs) [1] provide superior performance to machine learning services such as image recognition, speech recognition, and pattern recognition. Therefore, the face recognition methods [2] [3] using DNN are introduced. In the face recognition section, there are two types of face authentication and face recognition: face authentication (this person is the same person?) and face recognition (who is this person?). An example of face recognition is a CCTV device used in the face recognition field that uses DNN to identify people. However, the face recognition system has a vulnerability in the face adversarial example. Face adversarial example of adding a little of noise to a face image can cause the face recognition system to mislead. For example, in the face recognition system [4] used for the face photograph of the ID card, the attacker intentionally modifies the face photograph so that the face recognition system mistakenly recognizes it, and the human can not detect the distortion of the modified face photograph.

However, the face friend-safe adversarial example method [5] can be useful in an army situation where friend and enemy forces are mixed. A face friend-safe adversarial example may be needed that is misrecognized by an enemy face recognition system and recognized correctly by a friend face recognition system. Kwon et al. [5] first proposed a friend-safe concept through experiments on MNIST [6] and CIFAR10 [7] image datasets. In this paper, we extend a friend-safe concept and apply it to face recognition system. We propose a face friend-safe adversarial example that is correctly recognized by the friend face recognition system and misidentified by the enemy face recognition system. The contribution of this paper is as follows.

- This study presents a face friend-safe adversarial example targeting FaceNet [4] [8] of the face recogntion system. We systematically organize the framework and principle of the proposed scheme.
- We compared the face images generated by the proposed method and the original face image. We also compared the existing adversarial example with the face friend-safe adversarial example.
- We show the performance of the proposed method that is trained on VGGFace2 [9] using the face test data, Labeled Faces in the Wild (LFW) [10].

The remainder of this paper is as follows. Section II introduces the related research, and Section III introduces the proposed method. The experiment is described and evaluted in Section IV. A discussion of the proposed scheme is presented in Section V. Finally, we draw our conclusions in Section VI.

II. Related Work

We describe the FaceNet in general and introduce methods to cause misclassification in face recognition system.

A. FaceNet

Face recognition methods [11] [12] using DNNs are trained by mediating face identities and bottleneck layers. This method has the disadvantage of learning more than 1000 images per one subject. In recent research, it has been reduced to dimensionality by using principal component analysis (PCA), but it is still easy to learn by one layer. However, unlike this approach, FaceNet [4] directly learns 128-D embedding and uses a triplet based loss function. The structure of this method consists of batch input layer and convolutional neural network (CNN) by L_2 normalization. This method is trained using triplet loss. The purpose of triplet loss is to reduce distance margins between an anchor and a positive. The triplet loss minimizes the distance between anchor and positive for the same identity and maximizes the distance as far as possible from the anchor and negative for other identity. Euclidean embedding per image is used as a DNN. Similar embedding space of face is learned directly by L_2 distance. In this method, the same face has a small difference, and if it is another face, it has a big difference. In this paper, we apply the inception-Resnet-v1 model [8], which is an easy and fast advanced model in the FaceNet model.

B. Attack methods on the face recognition system.

There are various studies on methods for causing misclassification of face recognition system. M Sharif et al. [13] proposed a specific eyeglass method that allows the face recognition system to be mistaken for another person wearing a particular eyeglass. This method can control the face area and access control in the face area. In the study of the existing face area, it is not easy for the attacker to manipulate the face due to the light condition, pose, and distance of the face change. However, this method proposes a method of wrongly recognizing the face machine by simply wearing special glasses. A Rozsa et al. [14] proposed a face adversarial example, which adds some noise to the face image to make the face recognition machine misperceive, although the human does not recognize it. In this method, fast-flipping attribute (FFA) method is proposed by applying fast gradient sign method. In this method, celebA [15] was used as a dataset, and the attack success rate was 73%, which caused the face recognition machine to misclassify the face adversarial example.

C. Adversarial example to recognize different classes

Adversarial example methods have been introduced that allow multiple models to recognize different classes by modifying one image at a time. One of these, the friend-safe method [5] [16], creates a friend-safe adversarial example that is properly recognized by a friend classifier and is not properly recognized by an enemy classifier when friends and enemies are mixed together, such as in a military situation. Because this friend-safe adversarial example has minimal distortion, a human cannot discern the difference from the original sample. This method uses MNIST and CIFAR10 datasets in image domain. Another method generates a multi-targeted adversarial example [17] meant to be recognized as a different class by each model when several models are being attacked. For example, if there are models A, B, and C, the attacker can use the method to generate an adversarial example that makes A incorrectly recognize it as right turn, makes B recognize it as left turn, and makes C recognize it as U-turn. This method is an extended version of the friend-safe adversarial example, and its performance was evaluated with the MNIST dataset. The current paper proposes a face friend-safe adversarial example, which extends the friend-safe concepts to a face recognition system.

III. Proposed Method

The purpose of this proposed method is to generate a face transformed example that is correclty recognized by the friend classifier and is misrecognized as a wrong class by the enemy classifier, while minimizing the distortion from the face original sample. This method is mathematically expressed as follows. The operation functions of a friend classifier M_{friend} and an enemy classifier M_{enemy} are denoted as $f^{\text{friend}}(x)$ and $f^{\text{enemy}}(x)$, respectively. Given the pre-trained M_{friend} and M_{enemy} and the original input $x \in X$, this is an optimization problem to generate the face adversarial example x^*:

$$\underset{x^*}{\text{argmin}}\ L(x, x^*)\ \text{ s.t. }\ f^{\text{friend}}(x^*) = y \text{ and } f^{\text{enemy}}(x^*) \neq y,$$

where $L(\cdot)$ is the distance measured between face original sample x and face transformed example x^*, and $y \in Y$ is original class.

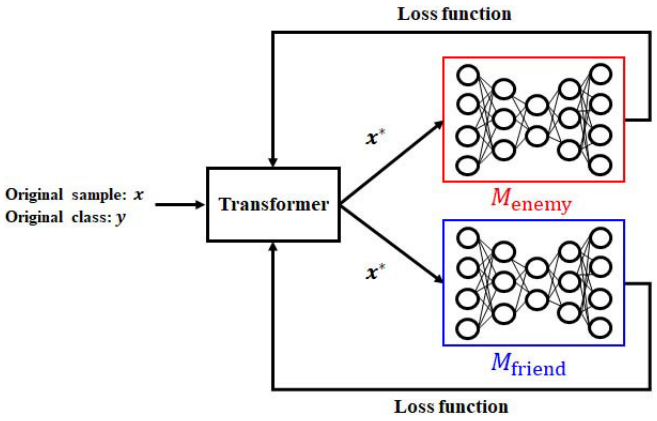

Fig. 1: Proposed architecture

To achieve this purpose, the proposed method consists of a transformer, a friend classifier M_{friend}, and an enemy classifier M_{enemy}, as shown in Fig.1. The transformer generates face adversarial example x^* by taking the original sample x and original class y as the input values.

$$x^* = x + \delta, \tag{1}$$

where δ is the noise. The classification loss of x^* by M_{friend} and M_{enemy} are returned to the transformer. M_{friend} and M_{enemy} take x^* as the input value and output the classification loss function result to the transformer. The transformer then

calculates the total loss $loss_{\mathrm{T}}$ and repeats the above procedure to generate a face adversarial example x^* while minimizing the total loss $loss_{\mathrm{T}}$. This total loss is defined as follows:

$$loss_{\mathrm{T}} = loss_{\mathrm{distortion}} + loss_{\mathrm{friend}} + loss_{\mathrm{enemy}}, \quad (2)$$

where $loss_{\mathrm{distortion}}$ is the distortion of the transformed example, $loss_{\mathrm{friend}}$ is the classification loss functions of M_{friend}, and $loss_{\mathrm{enemy}}$ is the classification loss functions of M_{enemy}. $loss_{\mathrm{distortion}}$ is the distance between the face original sample x and the face transformed example x^*:

$$loss_{\mathrm{distortion}} = \delta. \quad (3)$$

To satisfy $f^{\mathrm{friend}}(x^*) = y$, $loss_{\mathrm{friend}}$ should be minimized:

$$loss_{\mathrm{friend}} = g^{\mathrm{f}}(x^*),$$

where $g^{\mathrm{f}}(k) = \max\{Z_{\mathrm{f}}(k)_i : i \neq org\} - Z_{\mathrm{f}}(k)_{org}$, and org is the original class. $Z_{\mathrm{f}}(\cdot)$ and $Z_{\mathrm{e}}(\cdot)$ [18] [19] are the probabilities of the classes being predicted by the two discriminators, M_{friend} and M_{enemy}, respectively. $f^{\mathrm{friend}}(x^*)$ has a higher probability of predicting the original class than other classes by optimally minimizing $loss_{\mathrm{friend}}$.

To satisfy $f^{\mathrm{enemy}}(x^*) \neq y$,

$$loss_{\mathrm{enemy}} = g^{\mathrm{e}}(x^*),$$

where $g^{\mathrm{e}}(k) = Z_{\mathrm{e}}(k)_{org} - \max\{Z_{\mathrm{e}}(k)_i : i \neq org\}$, and org is the original class. $f^{\mathrm{enemy}}(x^*)$ has a lower probability of predicting the original class than other classes by optimally minimizing $loss_{\mathrm{enemy}}$. The details of the procedure for generating a face friend-safe adversarial example are given in Algorithm 1.

Algorithm 1 Face friend-safe adversarial example

Input: original sample x, original class y, number of iterations l.
Output: Face friend-safe adversarial example

$\delta \leftarrow 0$
$org \leftarrow y$
$x^* \leftarrow 0$
for l step **do**
  $x^* \leftarrow x + \delta$
  $g^{\mathrm{f}}(x^*) \leftarrow \max\{Z_{\mathrm{f}}(x^*)_i : i \neq org\} - Z_{\mathrm{f}}(x^*)_{org}$
  $g^{\mathrm{e}}(x^*) \leftarrow Z_{\mathrm{e}}(x^*)_{org} - \max\{Z_{\mathrm{e}}(x^*)_i : i \neq org\}$
  $loss_T \leftarrow \delta + g^{\mathrm{f}}(x^*) + g^{\mathrm{e}}(x^*)$
  Update δ by minimizing the gradient of $loss_T$
end for
return x^*

IV. Experiment and Evaluation

Through experiments, we show that the proposed scheme generates a face friend-safe adversarial example that is correctly classified by a friend classifier and is misclassified as wrong class by an enemy classifier, while minimizing the distortion distance from the face original sample. We used the Tensorflow library [20], widely used for machine learning, and a Xeon E5-2609 1.7-GHz server.

A. Datasets

For training data, we used VGGFace2 [9] dataset in the experiment. VGGFace2 has a 3.31 million images for the 9131 subjects, and the average number for images of each subject is 362.6. For test data, Labeled Faces in the Wild (LFW) [10] were used as datasets in the experiment. LFW face data contains 13,233 images collected from the web. It specifies the name of the person in the face image, and there are 1680 people, including two or more distinct face images.

B. Pretraining of pretrained models

Pretrained models M_{friend} and M_{enemy} are basically the structure of Inception-ResNet-v1 [8]. Their configuration and training parameters are shown in Tables III, and IV of the Appendix. The adam [21] was used as the optimizer. The initial constant of M_{friend} and M_{enemy} were 0.01 and 0.015, respectively. M_{friend} and M_{enemy} are different models with different parameters as they learned using different initial values. In the LFW test, M_{friend} and M_{enemy} correctly classified the face original samples with 99.31% and 99.24% accuracy, respectively.

C. Experimental results

When generating random 100 face adversarial examples with random LFW test datas, the adam [21] was used as the optimizer. The learninig rate was 0.01 and the constant value was 0.01. Table I shows image samples for the face friend-safe adversarial example and the face original sample. In Table I, the face friend-safe adversarial example is very similar to the face original sample, since a small noise is added to human perception. However, a friend-safe adversarial example is recognized as an wrong class rather than original class by the enemy classifier and is recognized correctly by the friend classifier.

TABLE I: Sampling of face friend-safe adversarial examples. Bob Hope, "1"; Bill Simon, "2"; Candie Kung, "3"; Ana Paula Gerar, "4"; Barry Ford, "5".

TABLE II: The average distortion, iteration, attack success rate of M_{enemy}, and friend accuracy of M_{friend} for face adversarial example. SD is standard deviation.

Description	Values
Iterations	1000
Average distortion	64.22
SD distortion	4.213
Attack success rate of M_{enemy}	92.2%
Friend accuracy of M_{friend}	91.4%

Table II shows the average distortion, the attack success rate of the enemy classifier, and the accuracy of the friend classifier for the proposed method. The attack success rate means the inconsistency rate between the face original class and the class that are mistakenly recognized by the enemy classifier. The friend accuracy means the consistency rate between the face original class and the class that are correctly recognized by the friend classifier. Distortion is the root sum of the square root of the difference between the face original sample and the face friend-safe adversarial example pixel in the L_2 distortion measure. As shown in Table II, the proposed method maintains an attack success rate of 92.2% for the enemy classifier and 91.4% accuracy for the friend classifier while maintaining a minimum of 64.22 distortion.

V. Discussion

Assumptions. The proposed method assumes a white box approach to the friend classifier and the enemy classifier. This method assumes that the attacker knows all information about the model structure, parameters, and output classification probability values for the friend classifier and the enemy classifier. The assumption of this proposed method is feasible. Even if the enemy classifier is a black box model, there is a method to attack by creating a similar model with a substitute network method.

Attack considerations. In a military situation where enemy and friend forces are involved, security is important. When applied to a face recognition system, a face recognition system that works well only in a friend system may be required. However, since face friend-safe adversarial example is an untargeted attack against enemy classifier, it is restricted by the enemy classifier to misidentify face friend-safe adversarial example as target class chosen by the attacker.

Applications. This method can be applied to ID cards for face recognition systems. It is possible to make ID cards that work only in certain friend face systems and not be recognized by other systems. Also, this paper may be extended to apply to face disguise. There is a possibility to be applied to a method of causing the face recognition system to misread by modulating a specific point on the face [22].

VI. Conclusion

In this paper, we propose a method to create face friend-safe adversarial example in face recognition system. The face friend-safe adversarial example is a face image that is correctly recognized by a friend classifier and is misidentified by an enemy classifier. Experimental results show that the proposed method has 92.2% attack success rate of the enemy classifier and 91.4% accuracy of the friend classifier with minimal distortion 64.22. In terms of human perception, the face friend-safe adversarial example is similar to the face original sample. Friend-safe concepts can be applied to audio and video applications in future studies. Also, research about defense mechanism against face-friend-safe adversarial example is also one of the interesting research topics.

Acknowledgment

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2018-0-00420 and No.2019-0-00426) and supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (NRF-2017R1C1B2003957 and 2017R1A2B4006026).

References

[1] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[2] A. Rozsa, M. Günther, E. M. Rudd, and T. E. Boult, "Facial attributes: Accuracy and adversarial robustness," *Pattern Recognition Letters*, 2017.

[3] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *International Conference on Learning Representations*, 2014.

[4] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823, 2015.

[5] H. Kwon, Y. Kim, K.-W. Park, H. Yoon, and D. Choi, "Friend-safe evasion attack: An adversarial example that is correctly recognized by a friendly classifier," *Computers & Security*, vol. 78, pp. 380–397, 2018.

[6] Y. LeCun, C. Cortes, and C. J. Burges, "Mnist handwritten digit database," *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, vol. 2, 2010.

[7] A. Krizhevsky, V. Nair, and G. Hinton, "The cifar-10 dataset," *online: http://www. cs. toronto. edu/kriz/cifar. html*, 2014.

[8] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[9] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "Vggface2: A dataset for recognising faces across pose and age," in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pp. 67–74, IEEE, 2018.

[10] E. Learned-Miller, G. B. Huang, A. RoyChowdhury, H. Li, and G. Hua, "Labeled faces in the wild: A survey," in *Advances in face detection and facial image analysis*, pp. 189–248, Springer, 2016.

[11] Y. Sun, X. Wang, and X. Tang, "Deeply learned face representations are sparse, selective, and robust," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2892–2900, 2015.

[12] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1701–1708, 2014.

[13] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1528–1540, ACM, 2016.

[14] A. Rozsa, M. Günther, E. M. Rudd, and T. E. Boult, "Facial attributes: Accuracy and adversarial robustness," *Pattern Recognition Letters*, 2017.

[15] Z. Liu, P. Luo, X. Wang, and X. Tang, "Large-scale celebfaces attributes (celeba) dataset," *Retrieved August*, vol. 15, p. 2018, 2018.

[16] H. Kwon, Y. Kim, H. Yoon, and D. Choi, "One-pixel adversarial example that is safe for friendly deep neural networks," in *International Workshop on Information Security Applications*, pp. 42–54, Springer, 2018.

[17] H. Kwon, Y. Kim, K.-W. Park, H. Yoon, and D. Choi, "Multi-targeted adversarial example in evasion attack on deep neural network," *IEEE Access*, vol. 6, pp. 46084–46096, 2018.

[18] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Security and Privacy (SP), 2017 IEEE Symposium on*, pp. 39–57, IEEE, 2017.

[19] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*, pp. 372–387, IEEE, 2016.

[20] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "Tensorflow: A system for large-scale machine learning.," in *OSDI*, vol. 16, pp. 265–283, 2016.

[21] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *The International Conference on Learning Representations (ICLR)*, 2015.
[22] H. Kwon, H. Yoon, and D. Choi, "Restricted evasion attack: Generation of restricted-area adversarial example," *IEEE Access*, vol. 7, 2019.

APPENDIX

TABLE III: A pretrained model parameters.

Parameter	**Values**
Learning rate	0.1
Momentum	0.9
Weight decay	0.005
Dropout	0.2
Epochs	150

TABLE IV: A pretrained model architecture for Inception-ResNet-v1 [8]. Conv is the convultional neural network. V means the use of "valid" padding, otherwise "same" padding is used.

Layer type	**Output shape**
Input	[299, 299, 3]
3 ×3 Conv (32 stride 2 V)	[149, 149, 32]
3 ×3 Conv (32 V)	[147, 147, 32]
3 ×3 Conv (64)	[147, 147, 64]
3 ×3 maxpool (stride 2 V)	[73, 73, 64]
1 ×1 Conv (80)	[73, 73, 80]
3 ×3 Conv (192 V)	[71, 71, 192]
3 ×3 Conv (256 stride 2 V)	[35, 35, 256]
5 × Inception-resnet1-A	[35, 35, 256]
Reduction-A	[17, 17, 896]
10 × Inception-resnet1-B	[17, 17, 896]
Reduction-B	[8, 8, 1792]
5 × Inception-resnet1-C	[8, 8, 1792]
Average pooling	[1792]
Dropout (keep 0.8)	[1792]
Softmax	[1000]