Defend against augmentation and attribute attacks in face recognition models

Students	Gil Kapel 305323776, Shelly Francis 316577725
Final Project	Deep learning 046211, Winter 2023
Date	26.01.2023

Introduction

Face recognition is a highly used method in real world computer vision tasks. Although deep learning era has witnessed highly accurate face recognition models these models can be vulnerable to adversarial attacks. Adversarial examples are inputs to models that an attacker has intentionally designed to cause the model to make a mistake. These examples can cause these models to incorrectly identify an individual. For example, an attacker may use glasses, wig or other disguises to change it's appearance. This can lead to security breaches and other serious consequences. Additionally, Face recognition (FR) models that are not robust to these attacks may be subject to bias.
Our focus is to attacks on FaceNet [1], A Unified Embedding for Face Recognition and Clustering.
In this project We demonstrate how augmentation and attribute attacks can affect FaceNet's accuracy and propose a fine-tune method to overcome the attacks and improve the network's accuracy.
The chosen attacks were:

- Augmentation attacks - Color Jittering, Gaussian noise and Blur.
- Attribute attack - Change hair color of the attack objects to blond using GAN network (StarGAN [2]).

A link to our open source implementation can be found here : GitHub

Related Work

Previous works using GANs for deceiving face recognition systems suggested networks such as AdvFaces [3] that manipulate geometric features of the face in the image. There were also attempts to fool face recognition models thorough manipulation faces in the physical world: for example, Sharif et al. [4] demonstrated the possibility of physically realizable attacks to

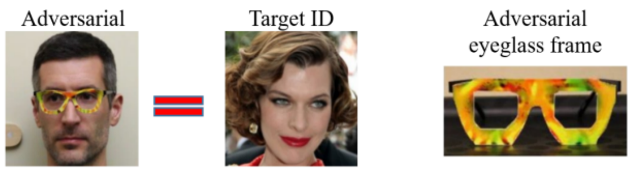

Figure 1: An example of fooling a face recognition model by wearing an adversarial eyeglass fame

impersonate an identity or evade the face recognition system. They devised an eyeglass frame for fooling the target network.

FaceNet

FaceNet [1] is a deep neural network model used for extracting features from an image of a person's face that was proposed by Google in 2015. The model takes an image of the person's face as input and outputs an embedding vector which represent the most important features of a face.

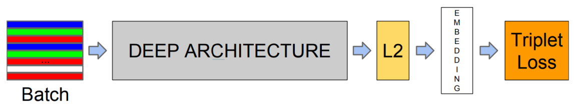

Figure 2: FaceNet model structure

FaceNet learns in the following way:

1. Randomly selects an anchor image.
2. Randomly selects another image from the anchor's class (positive example).
3. Randomly selects an image of a person different than the anchor image (negative example).

Adjusts the FaceNet network parameters so that the positive example is closer to the anchor than the negative example (triplet loss). These steps repeat until there are no more changes to be done, all the faces of the same person are close to each other and far from others.
Finally, softmax classifier is used as a final step to classify a person based on a face embedding and a finite list of identities that the model get in advance.

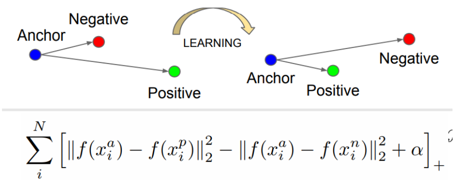
Anchor
Negative
Positive
LEARNING
Anchor
Positive
Negative
$\sum_i^N \left[\|f(x_i^a)-f(x_i^p)\|_2^2-\|f(x_i^a)-f(x_i^n)\|_2^2+\alpha\right]_+$

Figure 3: The Triplet Loss

The triples loss ensures that an image x_i^a (anchor) of a specific person is closer to all other images x_i^p (positive) of the same person than it is to any image x_i^n (negative) of any other person.

Star GAN

Star GAN is a model that transfers an image that has attributes list A to an image with attribute list B.

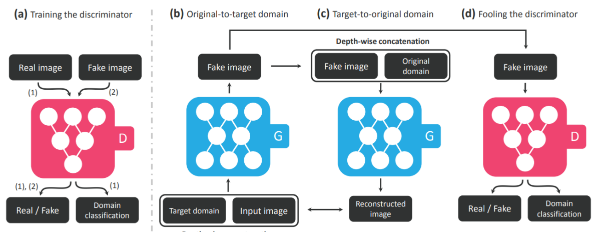

Figure 4: Overview of StarGAN

Let's explain the steps described in Figure 4:

(a) The Discriminator learns to distinguish between real and fake images, and outputs how likely the image will be from a single attribute.

(b) The Generator takes in as an input both the image and the target domain label and generates a fake image with the target domain.

(c) The generator tries to reconstruct the original image from the fake image given the original domain label and minimize the loss between the original and the reconstruction (cycle consistency loss).

(d) Finally, The generator tries to generate images that are indistinguishable from real images and are classifiable as target domain by discriminator. meaning The generator will ultimately learn to generate realistic images corresponding to the given target domain.

The loss functions used in the model training was Adversarial loss, Domain classification loss and Reconstruction loss. The Adversarial loss's purpose is to make the generated images indistinguishable from real images. Domain classification loss tries to translate an image and an attribute list c into an output image, which is properly classified with attributed defined in c. The reconstruction loss aspires to output realistic images and insure we only change the desired attributes (the difference between the original attributes a and the new attributes).
The total loss described as:
x denotes an image and c denotes an attribute target list, c' represent the input attribute of the input image.

$$\mathcal{L}_D = -\mathbb{E}_x[logD_{src}(x)] - \mathbb{E}_{x,c}[log(1 - D_{src}(G(x,c)))] + \lambda_{cls} \cdot \mathbb{E}_{x,c'}[-logD_{cls}(c'|x)]$$
$$\mathcal{L}_G = \mathbb{E}_x[logD_{src}(x)] + \mathbb{E}_{x,c}[log(1 - D_{src}(G(x,c)))] + \lambda_{cls} \cdot \mathbb{E}_{x,c'}[-logD_{cls}(c|G(x,c))] + \lambda_{rec}\mathbb{E}_{x,c,c'}[||x - G(G(x.c),c')||_1]$$

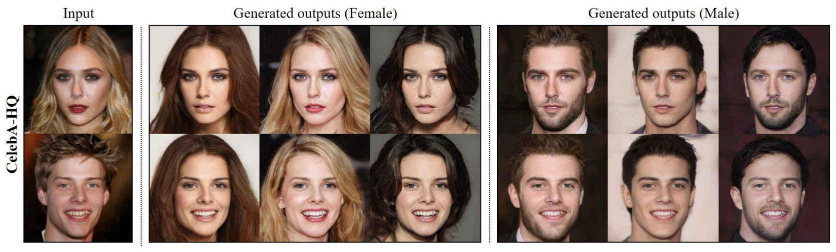

Figure 5: Diverse image synthesis results on the CelebA-HQ

Dataset - CelebA-HQ

CelebA is an open-source large-scale dataset that contains more than 200,000 images of celebrities with different attributes. The images in this dataset cover large pose variations and background clutter. CelebA-HQ is a high-quality version of CelebA that consists of 30,000 images at 256x256 resolution. Due to computational reasons, We used a subset of CelebA-HQ with more then 300 identities and 5,000 images (16 images per identity in avg).

1 Hyper-Parameter search

At first, we have tried to imitate the hyper-parameters used in CelebA-HQ-Face-Identity-and-Attributes-Recognition-PyTorch GitHub. In order to further improve our results, we decided to use Optuna python package to preform a hyper-parameter search over the following parameters: optimization algorithm, learning rate and batch size. We split the search to 2 section: Train (on the original FaceNet) and Fine-Tune on FaceNet.We had 100 trials each. the results where:

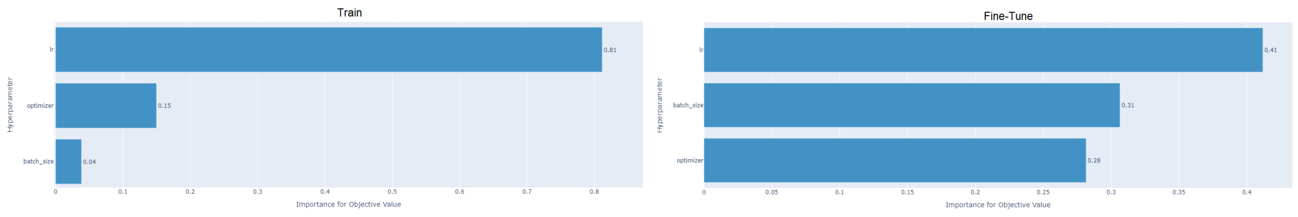

Figure 6: Train and fine-tune check - hyper parameter's importance

1. Train: optimizer: SGD ,lr: 0.018886927287921115, batch size: 64
2. Fine-tune: optimizer: SGD, lr: 0.000615545095147844, batch size: 64

experiments and results

In this project, in order to make FaceNet more robust to adversarial attacks, we used adversarial training : where the model is trained on a dataset that includes augmented and modified images. First, in order to match the dataset to the model we changed the last layer width to fit the number of identities in the subset and fine-tuned FaceNet 20 epochs while enabling the learning only to the last layer (logits layer before the softmax). parameters was the optimal train parameters from Optuna. We reached a test accuracy of 97.2% and a train accuracy of 99.76%. Denote that in the train and test transformations we normalize the images.

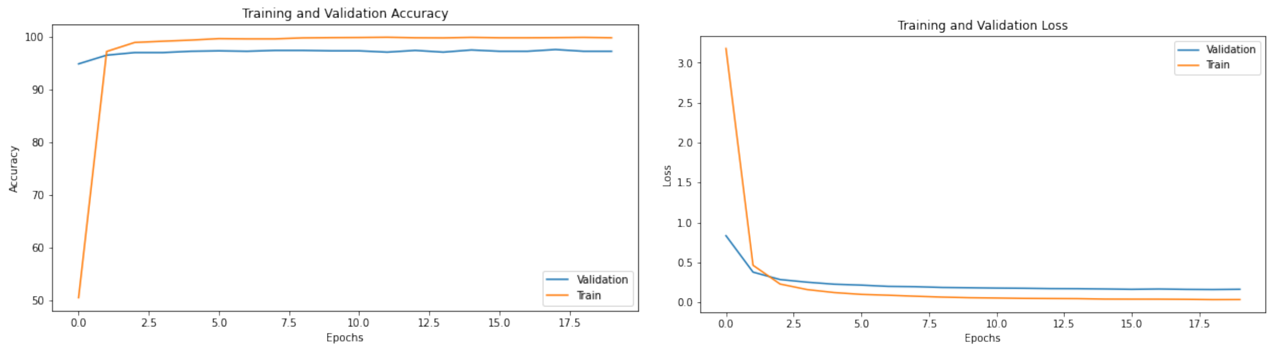

Figure 7: Fine-tune FaceNet on CelebA-HQ

Defend against augmentation attacks

First, We tested the robustness of the model to augmentations by adding noise to the images. We saw that the model dealt pretty well with noise and only an addition of Gaussian noise with mean 0 and std around 0.7 actually changed the predictions ID. Then, we tried a different approach - add multiple augmentations at the same time. The augmentations we used were (the exact value choice was random between the given values):

1. Gaussian Blur with changing kernel size (5x5 or 9x9) and changing sigma between 0.1 to 5 + Gaussian noise with std of 0.1.
2. Color Jitter with brightness between 0-0.1 and hue jitter between 0-0.9, where the factor is chosen uniformly + Gaussian noise with std of 0.1.

The model's results were:

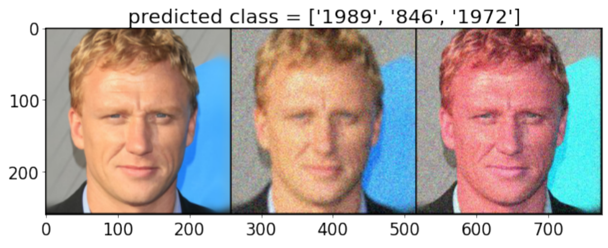

Figure 8: FaceNet results on augmentation. True ID = 1989

We added the augmented data to the train and test set and calculated the new accuracy:
FaceNet evaluation on the not augmented data: Loss: 0.16 Acc: 97.2%
FaceNet evaluation on the augmented data: Loss: 1.21 Acc: 74.32%
To improve the network's robustness to augmentation we trained the fine-tuned FaceNet 5

more epochs while still enabling the learning only to the last layer (logits layer before the softmax). The validation Loss and accuracy were not consistent and we saw on the 4th epoch a deterioration in the loss and accuracy. This can be caused due to the augmented images and the random values that were chosen to the augmentations.

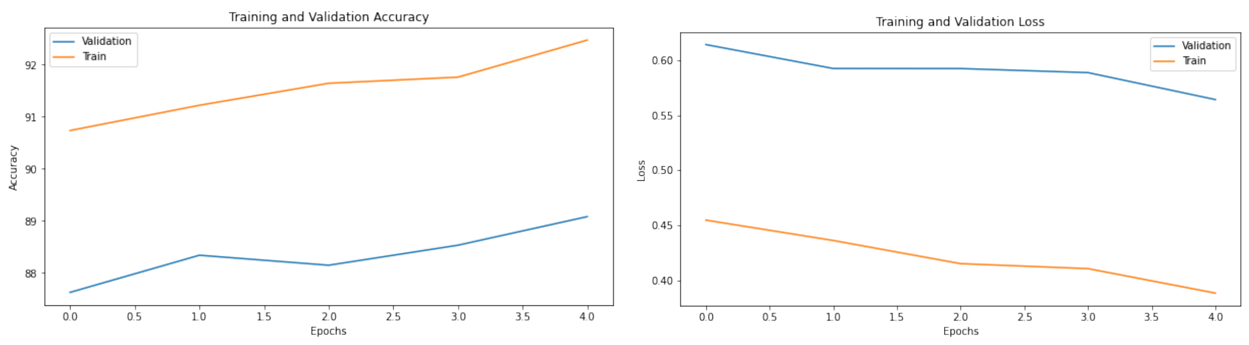

Figure 9: Fine-tune FaceNet on Augmented data

After the train, we saw that the accuracy changed to: Loss: 0.5867 Acc: 88.6968% , a 14% increase in the accuracy and decreasing the loss by more then half. The new model's results were:

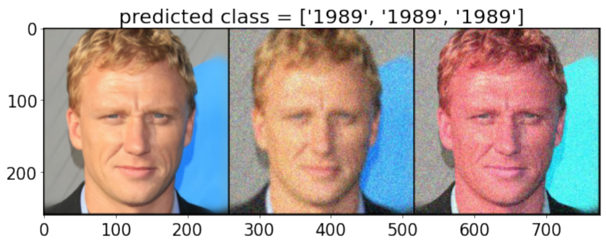

Figure 10: New FaceNet results on augmentation. True ID = 1989

In order to try a different approach, we tried to train the model from scratch using the dataset containing the augmentations. the results were similar to the results after the fine-tune with the augmentations and gave Loss: 0.4516 Acc: 90.5350% a difference of about 1.8% in accuracy. But,this method was computational expensive and took a while to train.

Defend against attribute attacks

We tested the robustness on the model to one attribute attack - hair color change (specifically to blond). In order to generate new images with blond hair we used StarGAN model. In order to evaluate the network on the same identities but with blond hair we used the same pre-trained network after the fine-tune on CelebA-HQ dataset. the new images with blond hair was added to the train and test dataset. the results were:
We got that the accuracy of the model without the defence was 44.0329%, the accuracy of the model trained on addition augmentations was 41.7833%. After we fine-tuned our model with attribute transformed data we improved our model accuracy to 77.5034%.

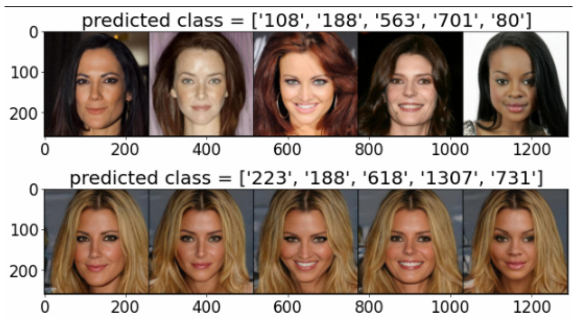
predicted class = ['108', '188', '563', '701', '80']
predicted class = ['223', '188', '618', '1307', '731']

Figure 11: FaceNet results on hair color attribute attack. True ID written in the first line of images

We noticed that our model isn't that accurate even after our defence, but we assume it happened because we tried to create attribute transformation to all of the images in the train-set without distinguish what identity is more relevant (we didn't have access to the attribute list of each identity in the dataset). Moreover, the roughly good test results on the fine-tuned CelebA can bring to a conclusion that the attribute change using StarGAN was good and close to a real world blond hair images of the identities. For improved model we train the fine-tuned CelebA 20 more epochs with the attribute attack images.

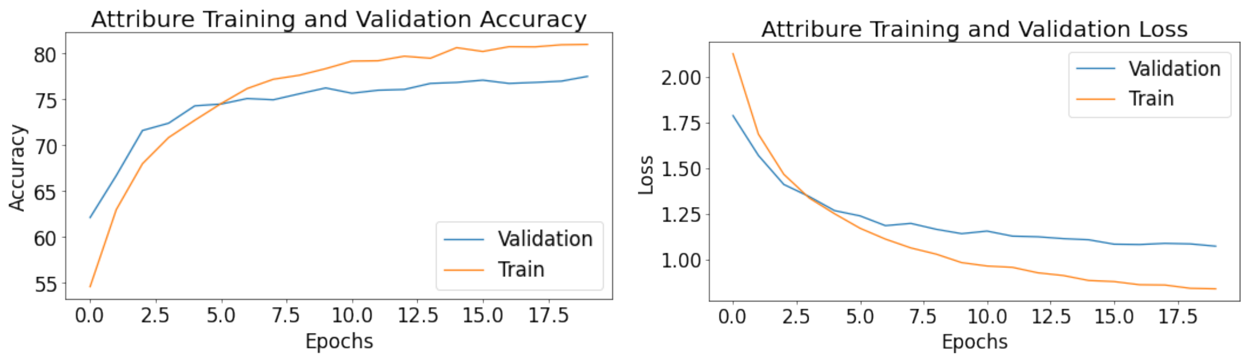

Figure 12: Fine-tune FaceNet on Augmented data

Note that this graphs are a result of our hyper-parameter search (not Optuna's) that gave a slightly better results. We used lr: 0.001 and batch size of 16. We tested the results on the same images as before and received the following results:

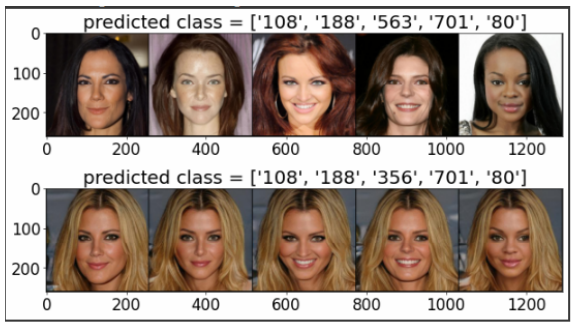
predicted class = ['108', '188', '563', '701', '80']
predicted class = ['108', '188', '356', '701', '80']

Figure 13: FaceNet results on hair color attribute attack after training. True ID written in the first line of images

Conclusion

From our results we concluded that FaceNet is not robust to all augmentation attacks. In the specific attribute attack we checked we saw that the model was quite robust, but fine-tune it on few more epochs can be useful to push the accuracy by a few extra presents. We saw that comparing the model trained from scratch with the new dataset received similar results to the 2 stage train of the model, meaning we don't need a lot of further training to reach the same results.

Future work

1. In order to make FaceNet more robust to augmentation attacks, we can train the network on a large verity of augmentations.
2. Run more tests with a range of seed numbers for more reliable results.
3. In the attribute attack we can match the CelebA-HQ subset images to the CelebA images (although they have different names) and add the information of the current attribute list to StarGAN in order to make a more reliable attributes.
4. We can add more attribute changes to the dataset and even multiple attributes changing in a single image (for example hair color + beard + glasses)
5. Build an inference algorithm that can distinguish whether the system is under adversarial attack, without specifying who is the real identity.

References

[1] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," pp. 815–823, 2015.

[2] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. 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, 2018.

[3] D. Deb, J. Zhang, and A. K. Jain, "Advfaces: Adversarial face synthesis," arXiv preprint arXiv:1908.05008, 2019.

[4] 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," p. 1528–1540, 2016. [Online]. Available: https://doi.org/10.1145/2976749.2978392