Final Project: Developing Filters to Trick Facial Recognition Software with GANs

Jialiang Liang, Chris Quion, Cole Weinhauer

INTRODUCTION

For our project, we are developing a system that creates filters that can be overlaid over human faces that cause facial recognition systems to be unable to detect the face. To do this, we will be using a generative adversarial network (GAN). GANs are 2 deep neural net systems, the generator and the discriminator, that are pitted against each other. These were first introduced by Ian Goodfellow and others in 2014, where this method was tested on the MNIST Dataset, the Toronto Face Database, and the CIFAR-10 Dataset [1]. GANs will be discussed more in the Approach Section. To do this, we will use the Labeled Faces in the Wild (LFW) Dataset, a dataset of over 13,000 face images with labels of the person's name. Also, we will use a subset of the Caltech 101 Dataset, which contains 40-800 images for each classification.

For our approach, we will be using a "targeted attack" technique, that focuses on both ensuring that the filter and face will be classified as not a face while minimizing the actual effects of the filter to the human eye. To do this, we will develop a system that can 1) take in a human face label and face images, 2) read them and find a separate label with a good chance of being a viable label, and 3) develop a filter that when applied to the face and passed to the discriminator, will be read as not a face. To test this, we will be getting various face images, developing filters for them, and seeing how well they fare against our discriminator.

PROBLEM STATEMENT AND PURPOSE

In order to construct our desired images, the face images overlaid with the filter created by the generator, the filters will be constructed such that when applied to the face image, the result will have a high confidence of being classified as not a human face. The discriminator's job is to decide whether an image is a human face or not. The desired outcome is to get filters that minimize the difference between the filter and the image, while still having the discriminator return a not a face result when it encounters the filter overlaid on the face.

One reason for us pursuing our project is because of the large relevance and importance of facial recognition in today's society. Facial recognition is used in many fields, including cyber security, sentiment analysis, criminology, and business to name some [2]. Thus, developing both a good discriminator and generator could help in many fields in many ways. For example, good discriminators could help real-time cameras detect possible threats quicker or it could help develop security systems that limit successful attempts to subvert the system. Similarly, developing a good generator could help to expose some problems with current facial recognition security systems. It could also be helpful for those who wish to be more secure in posting picture of themselves online, where scrapers could easily gather one's profile.

Another reason for this project's creation is to work more with GANs. As mentioned previously, GANs are a relatively new type of machine learning, only created 4 years ago. However, in that short time they have been very successful in generating samples of high dimension objects, like text and images. This is because of GANs ability to project data and find similarities in such high dimensions. Similarly, GANs, because of how they look at data, have applications in almost every fields. GANs can be used in everything from generating medical images for helping people diagnose diseases, to helping to create images with synthesized textures, and to even help generate astrophysical galaxy images past the current point of human ability (and all of these will be discussed below).

RELATED WORK

One way that GANs are being used is with real time texture synthesis. This method, created by Chuan Li and Michael Wand, uses a Markovian Generative Adversarial Network (MGAN) combined with using “neural patches”, a sampling of a synthesized image from the generator and the real image it is based off of. Training was done on one hundred random images from ImageNet and with a single texture. Following training, synthesis was fast and allowed for variants of input style. Results showed that this type of GAN was able to more effectively map textures in images, especially images that were considered more complex (more colors, textures) [3].

Another way GANs are being used is in the medical field. One such way is improvements to medical image synthesis, specifically in the field of Computed Tomography (CT) Imaging, which has wide arrays of uses in both the diagnostic and therapeutic fields. One such GAN was developed by Dong Nie and others, which includes a generator which tries to estimate a CT given some input and a discriminator which distinguishes between real CT and the generated CT, which is done in both systems by minimizing the binary cross entropy. This method was tested against 3 others on 2 separate datasets: one of 22 pelvic CT and MR Images and one of 16 subjects MR and CT images acquired from the Alzheimer’s Disease Neuroimaging Initiative database. This method was shown to outperform all 3 in both mean absolute error and peak signal-to-noise ratio [4]. Similarly, Thomas Schlegl and others used GANs to help with anomaly detection. Here, they used a CNN discriminator and generator optimized around the Nash equilibrium of costs, rather than a specific loss function. This was tested on Optical Coherence Tomography (OCT) volumes of the retina, where the model was able to effectively label all anomalies, and even outperformed other GAN based methods used in the field currently, like the AnoGAN, a deep convolutional GAN [5].

Finally, GANS are being applied to educational fields. Kevin Schawinski and others developed a GAN that can be used to recover features in astrophysical images of galaxies. The generator is fed images of galaxies with degradation on them and the images without the degradation and then the GAN learns to remove the degradation by minimizing the differences in the images. This recovered image is then passed to the discriminator with the real image where it decides if these images are recovered or original. It was found that the generator could help recover features past the deconvolution point, which is the current practice for image recover in the astrophysics field [6]. Another such example is using GANs to develop jet images, which are 2D representations of energy depositions from particles interacting. The GAN generates realistic synthetic particle reactions that have the desired low-dimension properties of jet images. It was found both qualitatively and quantitatively effective at generating these images [7].

For our approach, we will be using GANs in the facial recognition field. Our approach is different from these: in all the above, the desired outcome is around generating images, so much more is on the generator. However, in our project, most of the emphasis is on better classification, and thus, the discriminator is more important. The generator is only integral in making better filters, not in guaranteeing that a filter works. Similarly, in most of the works above, we have the discriminator choose between real and generated images, where as in our project, we will be combining these and having the discriminator check if this is a face or not. Ultimately, the other experiments use GANs to learn more about the data they are generating from, where as we will be using GANs to see what does not characterize our data.

DATASET

For our project, we use 2 datasets. The first is the LFW Dataset, which is over 13,000 images collected from the internet. Each image is labeled with the person's name, and 1680 people have two separate pictures within the dataset. We specifically will be using the deep funneling set of images, which is a process that reduces variability from factors like posse or shadowing. Among the four augmented LFW datasets, this one is tested as performing the best when it comes to facial recognition algorithms. The database was both created and updated by University of Massachusetts, Amherst Researchers, specifically, Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller [8].

The other dataset we will use is the widely-used ImageNet, which contains more than a thousand classes of various objects. We will be using this dataset in order to be able to push classification of faces towards classification of a desired class. For instance, we will be overlay filters on faces such that the discriminator will classify it as a goldfish [9].

APPROACH

Our approach to create these filters is to develop a GAN, as we mentioned above. The idea is we are going to do something similar to a "targeted attack", which is a technique used in GANs to get the discriminator to classify an image as a certain label [10]. Thus, we will optimize around the following equation

$$C = \frac{1}{2}||Y_{goal} - Y(x)||_2^2 + \lambda||X - X_{target}||_2^2$$

Where Y_{goal} is the goal label for our filter and face (so our target class), $Y(x)$ is the distribution of random noise that is acting as our overlaid filter, X is the combination of the face and the filter, and X_{target} is the face image that we based our filter around at the beginning of the process. Thus, we are trying to do two things: 1) maximize the confidence of our random noise being classified as our targeted classification, and 2) minimize the difference between our face and filter and the original face. Below is an example of what we are trying to accomplish.

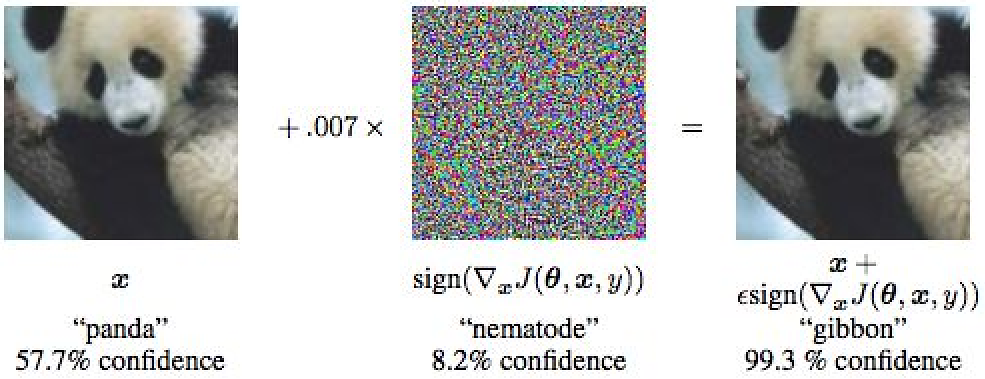

+ .007 ×
=
$\boldsymbol{x}$
"panda"
57.7% confidence
$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"nematode"
8.2% confidence
$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"gibbon"
99.3 % confidence

Figure 1: We see that by combining our image (classified as a panda with 57% confidence) with a random distribution filter (classified as a nematode with low confidence), that we can get unexpected classifications, like the one above (image and filter classified as a gibbon with very high confidence)

To do this, we need to develop a fooling system that can actually attack a certain classification. For this, we will be using the foolbox library, which helps to create adversarial examples against neural net systems. To do this, we will be calling upon the Foolbox library, specifically the LBFGSattack library. This uses limited-memory BFGS to minimize the differences between the face and the face and filter. This is done by optimizing the

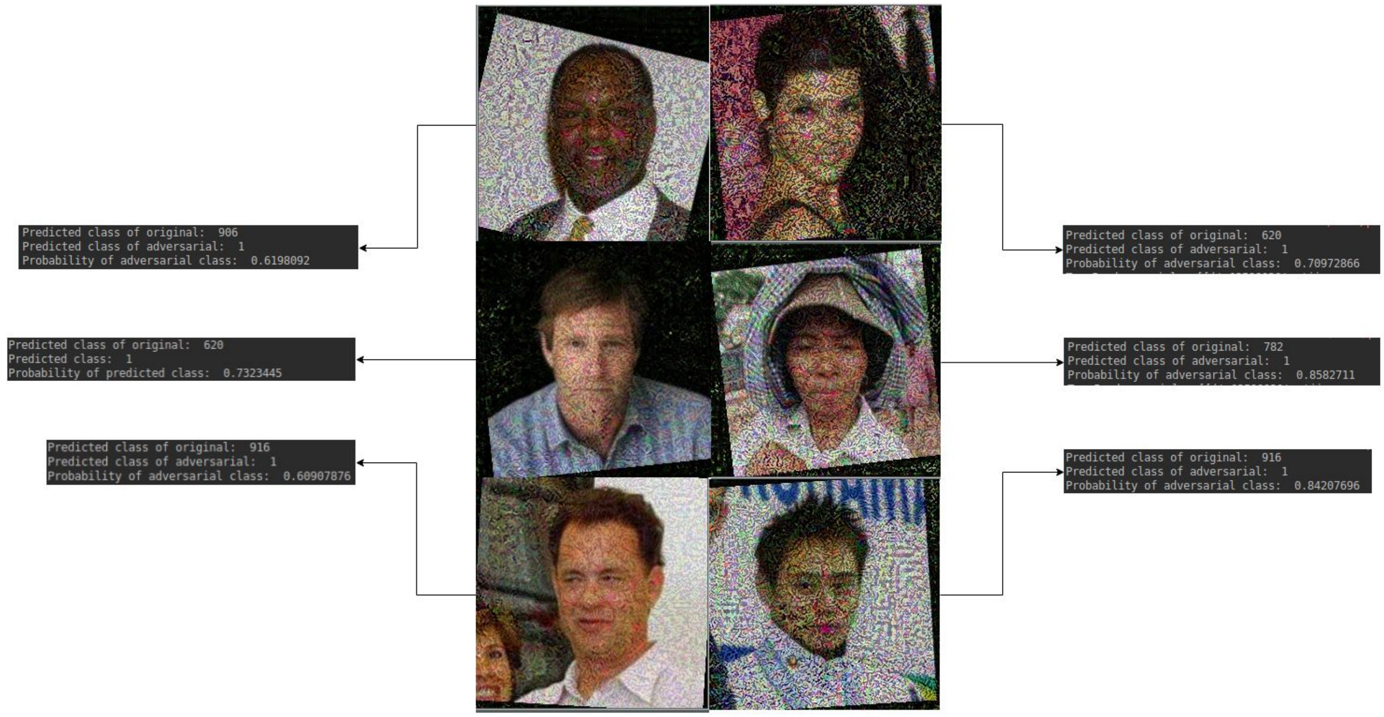
Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm [11]. This attack plus our face and the desired label we want it to fool, and we get an image back. This is our generated filter with our face.

For our discriminator, we are using a deep convolutional neural network trained on ImageNet. Specifically, we will be using ResNet50. The ResNet50 is a residual network model, which is a deep neural network, that has 50 input nodes [12][13]. We pass this to our foolbox as our model, and then use it to get predictions for the classifications of our generated face and filter. We then return this label and the confidence of it by the model.

Finally, we go into our training. We will first take a batch of out photos, specifically half of them, add random noise filters over them, and then pass them to the discriminator. We then get the results over both the fake images and real images, and train on the generator based around some loss function [14]. This is done for a defined very large number of sets (the cited link has the the epochs as 30000), so we need to install a save setting, saving our progress after a certain number of our steps above.

EXPERIMENTS

For testing of our system, we took various images, developed a filter for them, and then passed them to the discriminator to see how it classified it.



```
Predicted class of original:  906
Predicted class of adversarial:  1
Probability of adversarial class:  0.6198092
```

```
Predicted class of original:  620
Predicted class of adversarial:  1
Probability of adversarial class:  0.70972866
```

```
Predicted class of original:  620
Predicted class:  1
Probability of predicted class:  0.7323445
```

```
Predicted class of original:  782
Predicted class of adversarial:  1
Probability of adversarial class:  0.8582711
```

```
Predicted class of original:  916
Predicted class of adversarial:  1
Probability of adversarial class:  0.60907876
```

```
Predicted class of original:  916
Predicted class of adversarial:  1
Probability of adversarial class:  0.84207696
```

Figure 2: Examples and Prediction Outputs for a subset of our results. The faces are in order: Danny Glover, Marisa Tomei, Aaron Eckhart, Sim Yong, Tom Hanks, and Shinya Tamaguchi. Confidence of targeted class label in order is .619, .710, .732, .858, .610, and .842

Figure 2. shows some examples of our results. Looking at the 'predicted class of the original', the class the discriminator predicts for the image of the face without the filter, we can see that the discriminator was unable to properly recognize the faces, and instead predicted the images to be pictures of either a CRT screen (ImageNet class 782), laptop (ImageNet class

620), or of a website (ImageNet class 916). However, we were able to successfully steer all predictions towards our targeted class of the goldfish, with significantly high probabilities, the lowest of which was ~61 percent.

DISCUSSION AND RESULTS

In the end, we were able to have the generator create a filter that can be overlaid over the face image passed to the generator and create an image to fool the discriminator. Also, we can do a targeted attack and even create a filter to target a certain classification of our images, specifically the first category (a goldfish). However, the original class that the discriminator was wrong. Here we can see that the discriminator classified the face (without the filter) as ImageNet class 620, which is a laptop. This can be attributed to the results of the deep funneling, which skewed the image and makes it appear as an image of a face on a screen, rather than just a face. Furthermore, we can see that the resulting image is still recognizable as a face to the human eye, reflecting the minimized distance between the original and resulting image.

This result is indicative of an error in our set-up, as we weren't able to merge the two datasets (LFW and ImageNet) into one cohesive dataset. Instead, we were only able to used the LFW dataset as a bank for faces to put filters on, and ImageNet as a dataset to push the classifications towards. Merging the two datasets and then training a discriminator on the merged dataset is a project that could be pursued in a later research project, and would surely enhance the results.

Furthermore, we did not properly implement a GAN, and instead only focused on generating adversarial examples. We did this using a framework called foolbox to generate adversarial examples. In a future project, implementing the whole GAN would produce stronger results.

CONCLUSION

In conclusion, making a GAN is hard. We specifically set out to develop a GAN that could generate filters that can be overlaid over faces and cause facial recognition systems to classify the image as not a face. We are going to do this by performing a targeted attack on a chosen label by us. We were inspired to do this for two major reasons: the large relevance and importance of facial recognition in many fields, like social, technological, or medical, and because of the large success GANs have had in the fields of imaging. We used the LFW deep funneled dataset to act as our source of faces and use ImageNet to act as our source of data that we train our discriminator on. We were able to generate filters based off of desired images, choose the desired label that we want our discriminator to pick, have our discriminator pick that label, and still have the generated images look like humans face to the eye. We did, however, have problems getting to the training stages of the GAN, and we had some issues with the initial classification of our images, based off of the staging of the images in the LFW deep funneling set.

CITATIONS AND REFERENCES

[1] https://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf
[2] https://www.globalme.net/blog/facial-recognition-technology-explained
[3] https://arxiv.org/pdf/1604.04382.pdf
[4] https://arxiv.org/pdf/1612.05362.pdf
[5] https://arxiv.org/pdf/1703.05921.pdf
[6] https://academic.oup.com/mnrasl/article/467/1/L110/2931732
[7] https://link.springer.com/article/10.1007/s41781-017-0004-6

[8] http://vis-www.cs.umass.edu/lfw/#download

[9] http://www.image-net.org/

[10] https://ml.berkeley.edu/blog/2018/01/10/adversarial-examples/?fbclid=IwAR0wvcFro4VSCPMuKhaXc0zglJaSpgwgpLLOyebrscz7KqsPDSvmHjsxr1c

[11] https://github.com/bethgelab/foolbox/blob/master/foolbox/attacks/lbfgs.py

[12] https://www.quora.com/What-is-the-deep-neural-network-known-as-%E2%80%9CResNet-50%E2%80%9D

[13] https://keras.io/applications/#resnet50

[14] https://skymind.ai/wiki/generative-adversarial-network-gan

-http://www.evolvingai.org/fooling?fbclid=IwAR2Ernn3z7AgQPAZy7QU1-9gj4qjd4-A2Y1bxxTTbQLbm68_3AzQTRVTWGg

-https://towardsdatascience.com/implementing-a-generative-adversarial-network-gan-dcgan-to-draw-human-faces-8291616904a

-https://medium.com/@jonathan_hui/gan-some-cool-applications-of-gans-4c9ecca35900