Learning Tattoos to Fool Facial Recognition Algorithms

Jenna Bates¹
Xiaoyu Zhang¹
Ajmal Mian²

¹ Center for Research in Computer Vision
University of Central Florida
Orlando, USA
² University of Western Australia
Crawley WA,
Perth, AUS

Abstract

In this paper, we introduce a means of utilizing adversarial attacks to fool facial recognition algorithms. The goal is to develop a facial tattoo that, when worn, leads a facial recognition model to misclassify the face. Currently, the attacks are non-targeted, meaning the goal is to simply misclassify the face, rather than misclassifying the face and classifying it as a chosen identity. The experiments began by implementing a projected gradient descent (PGD) attack on ImageNet images, then VGGFace2 images, and then finally localizing the adversarial attack to one portion of the VGGFace2 faces with the intention of printing them out and applying them to the face to fool facial recognition algorithms on the streets. Future plans for this project involve

1 Introduction

Adversarial attacks involve intentionally adding “invisible noise” into an image such that a human eye would not notice the change, yet a model will misclassify the image. As facial recognition and image classification becomes more popular in the world today, adversarial attacks present the question of their reliability. This project aims to utilize a projected gradient descent adversarial attack localized to the cheek to develop printable, wearable tattoos to fool facial recognition algorithms in the real world.

2 Datasets

We utilized two datasets for our adversarial attack experiments: VGGFace2 and ImageNet.

2.1 VGGFace2

The VGGFace2 dataset [3]. consists of 3.31 million images, all of which vary in ethnicity, profession, pose, lighting, and background. The images are of 9131 celebrities, approximately 59.3% male. Each identity has about 362.6 images on average and the dataset is split into training and evaluation groups. The training has 8631 classes and evaluation has 500 classes.
2.2 ImageNet

The ImageNet database contains over 14 million annotated images within over 20,000 categories. The annotations are crowd sourced and indicate a presence or absence of specific object classes along with bounding boxes around the object.

3 Adversarial Attacks

3.1 Fast Gradient Sign Method

The Fast Gradient Sign Method (FGSM) is considered the most efficient of adversarial attacks to run. It consists of "computing the sign of classifier’s cost function’s gradient and adding a certain error to each element in the data sample consistent with the direction of the sign" [2]

3.2 Projected Gradient Descent Attack

Projected Gradient Descent (PGD) attacks are iterative versions of the FGSM attack, calculating a new gradient with each iteration and adding a significantly smaller perturbation to each element in the sample. Unlike the FGSM attack which calculates only one gradient of the cost function and adds a larger bounded error. [3]

4 Experiments

There were three main experiments in the project: applying a PGD attack to non-face images, applying a PGD attack to face images, and finally localizing the PGD attack to a 50 pixel by 50 pixel square on the face image.

4.1 Applying A PGD Attack To Imagenet Images

In order to get a better understanding of PGD attacks and applying them to images, we started off by applying it to the Imagenet images. We tested various epsilon values and received near 100% fool rates (12.5% model accuracy) to the Resnet18 model at an epsilon value of .01 (the images are normalized on a scale of 0-1) and a perfect fool rate (0% model accuracy) to the model at an epsilon value of .03. The results for this experiment are shown in table 1. This experiment was implemented using tools provided by Foolbox. [4]

<table>
<thead>
<tr>
<th>Epsilon</th>
<th>Model Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (No Attack)</td>
<td>0.9375</td>
</tr>
<tr>
<td>.01</td>
<td>0.9375</td>
</tr>
<tr>
<td>.01</td>
<td>0.125</td>
</tr>
<tr>
<td>.03</td>
<td>0.0</td>
</tr>
<tr>
<td>.07</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Table 1: PGD Attack on Imagenet Images
4.2 Applying A PGD Attack to VGGFace2 Images

Prior to applying the PGD attack to the VGGFace2 dataset we tested the VGG model’s base accuracy. This resulted in a 70% base accuracy, likely due to the lack of facial landmarks and image cropping. The VGGFace2 database provides five facial landmarks, which are created by locating the face using the hyper-parameters in [2] and extending the bounding box by a factor of .3. Once the facial landmarks were applied, the images put into the model were much more cropped to the area just surrounding the face and the base accuracy for the model reached 90%. The differences in the images with and without the landmarks can be seen in figures 1 and 2.

Figure 1: VGGFace2 image before adding landmarks

Figure 2: VGGFace2 image after adding landmarks

The model is now ready to have the attack applied. The images are normalized on a scale of 0-255 so the epsilon values appear higher than in the previous experiments. The results for the experiment allowed for near perfect fool rate (.3% model accuracy) at an epsilon of 3 and a perfect fool rate (0% model accuracy) at an epsilon of 10. This experiment was implemented using tools provided by Foolbox. [4] The results of this experiment after applying the image landmarks are seen in table 2.

<table>
<thead>
<tr>
<th>Epsilon</th>
<th>Model Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (No Attack)</td>
<td>0.90</td>
</tr>
<tr>
<td>1</td>
<td>0.23</td>
</tr>
<tr>
<td>3</td>
<td>0.003</td>
</tr>
<tr>
<td>10</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Table 2: PGD Attack on VGGFace2 Images
4.3 Applying the Localization Mask

Our final experiment involved localizing the PGD attack to one area of the face. We chose the bottom left corner of the image to be as close as possible to the cheek. Originally, we planned on having the size of the attack to be a 30 pixel by 30 pixel square. However, because of the VGGFace2 dataset having variation in the poses of the subjects in the images [3], we felt that it was better for now to raise the size of the attack to be a 50 pixel by 50 pixel square in order to ensure the attack touched the face. The results of this attack only provided us with a 40% fool rate, likely due to the issue involving the attack not touching the face and not being large enough. Figure 4 shows an example of the square not being large enough to touch most of the subject’s face. We have future plans to create an active localization method to locate the eyes and nose and determine the location of the cheek, and therefore the location to place the adversarial tattoo. We believe that with this method we will be able to achieve much higher fool rates and be able to decrease the size of the tattoo to a realistic size. The epsilon for the tattoo in this experiment is 255 (the maximum) as we are no longer worried about making the attack look as if it it not there. The idea would be to make it seem like it is a facial tattoo or face paint so it is able to be colorful and unnatural. We aim to make the size of the tattoo equivalent to 30 pixels by 30 pixels or smaller.

Figure 3: Example of Localization Mask

Figure 4: Example of failed attempt to fool the model due to location and size

5 Future Plans

In the future, we hope to make the shape of the localization mask more towards a particular shape, such as a flag or simple line tattoo. This is with the intention of making the tattoo seem like something an average person would put onto their face either on a daily basis or at a special event such as a sports game or fair. The idea of developing a targeted attack to essentially ’transform’ a subject into a designated identity in the eyes of a facial recognition algorithm is also in the scope of our plans, particularly through the means of a black box.
attack. This would allow our attacks to be applied to not only one specific facial recognition model but most facial recognition models. The tattoo is to be printed out and applied to the face. The goal is for it to work from all visible angles and cause a near perfect misclassification. The location of the tattoo is to be determined by an active localization tool created using a face detection tool to determine the location of the eyes and nose and then the cheek. The placement of the tattoo is to be on the cheek.

References


