Adversarial Attack and Training on Deep Learning-based Gaze estimation

Clio Feng [1]

Supervisor(s): Dr. G. (Guohao) Lan, Lingyu Du

[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 25, 2023

Name of the student: Clio Feng
Final project course: CSE3000 Research Project
Thesis committee: Dr. Guohao Lan, Lingyu Du, Dr. Xucong Zhang

Abstract

Recently, while gaze estimation has gained a substantial improvement by using deep learning models, research had shown that neural networks are weak against adversarial attacks. Despite researchers has been done numerous on adversarial training, there are little to no studies on adversarial training in gaze estimation. Therefore, the objective of this project is to investigate how these adversarial samples affect the gaze estimation's performance and how the adversarial training elevates the effect of these adversarial attacks. For projected gradient descent adversarial attack, the result shows that the bound of the final noise, the step size and the number of steps toward the gradient, and the randomized noise initiation are all able to worsen the baseline performance to varying degrees. Further, the performance reveals that while projected gradient descent adversarial training can defend against certain adversarial attacks, its performance is not converging to the baseline. In general, the performance of adversarial training on gaze estimation could be influenced by data augmentation, loss function, model capacity, and the type of adversarial training.

1 Introduction

In recent days, gaze estimation has achieved a significant improvement by using deep learning. However, considerable deep learning-based methods suffer from vulnerability properties. Recent research had revealed that neural networks are not very robust in terms of dealing with slightly different distributions [9]. The reality that such a slight perturbation can force the existing neural networks to falter prevents the models from being used in security-critical areas [16]. As deep neural networks are susceptible to adversarial examples, the attackers can exploit their weakness to confuse the models by perturbing the raw image using noise. For gaze estimation, it means that the altered image might visually look similar to the original image, but the deep learning models will output the incorrect gaze direction [21].

1.1 Related Work

As some of the most recent results suggest that the presence of adversarial attacks may be an intrinsic deficiency of deep learning models, the work of [7] introduces the Fast Gradient Sign Method (FGSM) a single-step method that relied on linearizing the loss around the data points to maximize the loss of an image. However, as it is relatively weak, the Iterative Fast Gradient Sign Method was presented in the work of [12], as a multiple-step attack method that is iterative to find the local maximum loss point. Despite them, researchers had done numerous works in unveiling the various ways the attackers can attack the models, as the works from [2, 16, 18].

To defend against adversarial attacks, researchers have experimented with various adversarial training methods. In the work from [7], although they have produced favorable outcomes from adversarial training with FGSM, the models are only robust against FGSM adversarial attacks, and are vulnerable to slightly more complicated adversaries, for example, multiple-step attacks. Among the numerous work that has been done [12, 13, 20], the most promising result had been shown with Projected Gradient Descent (PGD) Attack from the work of [15]. It not only achieves low angular error but also proves to be able against various types of attacks. Nonetheless, the performance is still dependable for each dataset.

1.2 My contribution

However, regardless of the progress made in image classification tasks, there are little to no studies that have been done regarding adversarial training in gaze estimation let alone prove effective. Therefore, this project will focus on adversarial attacks and training on deep neural networks for gaze estimation. Specifically, this project will focus on PGD attack, as it is the most effective method for adversarial training on the classifier yet.

1.3 Research Questions

To better understand the ways the attackers can make these adversarial inputs that are nearly imperceptible from raw data and yet lead to false classification by the network [15], how they affect the performance of gaze estimation, and the effectiveness of adversarial training against such inputs, the paper will investigate the following subquestions:

- What are the different effects of PGD attacks with different experimental settings in the gaze estimation model?
- How adversarial training elevates the adversarial attack on gaze estimation?

2 Methodology

2.1 Overview

After modifying the original test samples by the PGD attacks with different experimental settings, I will input the adversarial samples into the baseline model and compare the angular errors against the original samples to assess the effectiveness and advantages of the PGD attacks. For adversarial training, the objective is to have both the angular errors of the original test sample and the adversarial sample decrease similar to the baseline model.

2.2 Projected Gradient Descent Attack

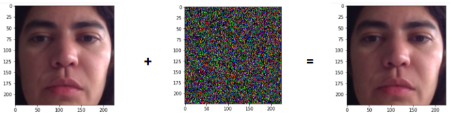

Figure 1: **An PGD Attack Example**

The PGD attack is a white-box attack which means the attacker has access to the model gradients [11]. The PGD attack is a multiple-step method [15]. PGD attack is to frame finding an adversarial example as a constrained optimization problem. The constraint is usually expressed as the L^2 norm of the perturbation. The perturbation will be added to the original image so the content of the adversarial example looks the same as the original image, which can see in Figure 1. Therefore, its goal is to find the noise that maximizes the loss of a model on a particular input while keeping the size of the perturbation smaller than a specified amount referred to as epsilon ϵ.

To find the maximum loss, PDG first start to pick a random perturbation within the range of negative to positive epsilons $[-\epsilon, \epsilon]$, which can be viewed as the bounded circle around the original sample. Second, PGD takes a gradient step toward the direction of maximizing loss with a step size α. If the perturbation exceeds the epsilon ϵ bound, PGD will project it back onto the bound. Therefore, the constraints for the amount of perturbation are satisfied. Finally, PGD repeats taking the gradient steps until convergence, which finds the local or global optimum.

Here the x is a data sample with label y, and N is the size of the dataset. An adversarial data sample of x is $x^{'}$. $J(\theta, x_t, y)$ represent the optimization loss (adversarial loss), where θ is the model gradient. The projected gradient descent can be represented in the following equation 1:

$$x^{'}_{t+1} = Proj\{x^{'}_t + \alpha \cdot sign[\nabla_x J(\theta, x_t, y)]\} \tag{1}$$

Therefore, after the images are modified by the PGD attacks with different experimental settings, I can input the perturbed sample into the model and compare the angular errors against the original sample for the untargeted attacks.

2.2.1 Adjustment for Gaze Estimation

Since PGD is usually for a classification problem, I change it to L1Loss for gaze estimation. In addition, the default parameter is also changed accordingly based on the experiment.

2.3 Projected Gradient Descent Adversarial Training

Common defense consists of introducing adversarial images to train a more robust network, which is generated using the target model [6]. For PGD adversarial training, I replace every training sample with its PGD-perturbed counterpart. The objective is to acquire a small adversarial loss after the replacement. If I can achieve a very small final loss against adversarial samples, it will mean the model is robust to adversarial inputs and no allowed attack can fool the network [15].

To improve adversarial training, I will first explore the effect of data augmentation on performance by experimenting with the varying numbers of adversarial samples in the training set. Second, I will experiment with the effect of different loss functions.

In addition to data augmentation and loss function, increasing model capacity might also enhance the model robustness against the perturbation [15]. The universal approximator theorem (Hornik et al., 1989) assures that a neural network with at least one hidden layer can represent any function to an arbitrary degree of accuracy so long as its hidden layer is permitted to have enough units [15]. Therefore, I will also explore the effect of the model capacity for gaze estimation in the following experiments by increasing the capacity of the network or using a stronger method for the inner optimization problem.

Furthermore, according to [15], a phenomenon they observed is that for image classification if I train a network to be robust against PGD adversaries, it becomes robust against a wide range of other attacks as well. Therefore, if the result is promising against the PGD attack, I will explore the robustness of the adversarial training by testing the trained model with other adversarial attacks.

2.3.1 Adjustment for Gaze Estimation

As the Projected Gradient Descent Adversarial Training method in [15] is for classification tasks originally, I modified the method for gaze estimation. Therefore, I change the cross-entropy loss for classification tasks to L1 loss.

2.4 Other Adversarial Training methods on Gaze Estimation

Hoping to improve the adversarial attack through the adversarial algorithm, in addition to the PGD adversarial training, I also explore other adversarial attack and training methods that also exploit the gradients of a neural network to build an adversarial image, which all of the other adversarial attack and training methods are either a build on or top of the PGD adversarial attack or a similar modification: GN, FGSM, BIM, FFGSM, PGD, PGD2, EOTPGD, MIFPGD, NIFPGD, SINIFGSM, VMIFGSM, VNIFGSM. A brief description of all the following adversarial attacks experimented with is detailed in the appendix A.

For each of them, I will use the following three steps to find whether the adversarial attack and training that are both effective in attack and defense. First, implement its attack to the baseline model to assess the effectiveness of each adversarial attack. Second, implement its own adversarial training, meaning replacing all of the original images in the training set with the adversarial counterparts that produce by it, against itself. For example, for FFGSM adversarial training, all of the images in the training set will be replaced with their FFGSM adversarial counterpart and the testing set will be both the original samples and its FFGSM adversarial counterpart. Third, to test the generalism of the adversarial training that proved effective, test it with PGD adversarial attack to ensure whether it is also able to defend against another type of attack as well not only its own, meaning the testing set will be including PGD adversarial samples.

3 Experiments

3.1 Experimental Setup

3.1.1 Dataset

The dataset is using the MPIIFaceGaze (normalized) [22], where it contains 15 different subjects' face images, and each subject has 3000 images with the gaze direction labeled (pitch and yaw). Each image is an RGB image with a height of 448 and a width of 448.

3.1.2 Model Arichitecture

For the baseline model, I experiment with three different models and compare their average angular error, and the average time is taken for a single epoch within a certain step size to determine my baseline model.

AlexNet The first 14 subjects are my training set, and the 15th subject is my test set. Contains 6 convolutional layers, each follows by batch Normalization.

LetNet The first 14 subjects are my training set, and the 15th subject is my test set. Compare to the first model the layer, only contains two convolutional layers, each follows by batch Normalization.

ResNet The first 14 subjects are my pre-training set, the 100 images of the 15th subject are used for fine-tuning, and the rest of the 15th subject is my test set. The RestNet contains four convolutional layers and the Residual block contains two convolutional layers, each follows by batch Normalization. For the calibration, I freeze all the batch normalization layers.

3.1.3 Training Details

I have resized the image to 224 × 224. For the angular error, I convert the pitch and yaw into 3-Dimensional vectors, and compare the angular difference between two direction vectors. 20 epochs are sufficient, as the angular error decrease by 0.01 degree when increasing epochs to 40. therefore, the default experimental setting is 20 epochs, a 0.0001 learning rate, and an Adam optimizer.

3.1.4 Baseline Models Performance

The more complex the neural network, the better the performance. I can see from the result that ResNet has an angular error of around 2, while AlexNet is around 6.8 degrees. For our experiments, a baseline of around 8 degrees is sufficient to test the effectiveness of the adversarial attack and training. Therefore, our baseline model is LetNet, where its angular error for training is 2.3 degrees and the angular error for testing is 8.2 degrees.

3.2 Aversarial Attack Visibility Experiment

Despite current architectures of models usually leaning on visual features that humans can see but ignore [4], the impact of the adversarial manipulation on choices made by human participants is still statistically consequential from the experiments of [5] and [8], and humans are sensitive to the exact type of non-robust features that lead to adversarial attacks.

Therefore, before experimenting on the experimental setting of adversarial attacks, I will address the issue of attack visibility. Because of the disparity in acuity of human and machine vision, humans might find some pictures entirely uninterpreTable. Nevertheless, the dissimilarity between human and machine perception of adversarial images depends on distinct types of attacks. Since while some types of adversarial attacks assemble images that appear completely undecipherable to humans, others might not depend on subtle visual features that are below the human perceptual threshold. Consequently, the human perceptual threshold could be a deciding aspect in aligning the experimental settings with all the adversarial attacks, since different experimental settings can have different levels of perturbation, as an example in Figure 7 with PGD attack.

One way to represent the human perceptual threshold could be from the difference of images, and it is calculated with the mean squared error (MSE) in Equation 2, where x_1, x_2 represent the two comparing images, h, w are height and weight of the image, $(x_1 - x_2)$ is the sum of pixels different.

$$MSE(x_1, x_2) = \frac{(x_1 - x_2)^2}{h \cdot w} \tag{2}$$

From Figure 2, it can observe that the perturbation becomes more visible when the image difference increased. When above 22, even though the changes in the images are slight to human eyes, there are still small perturbations if one looks closely, yet when below 20 there is no visible trace of the attack can be found. Therefore, I set the image difference to 22 as the human perceptual threshold.

3.2.1 Experiments and Result

One way to set the human perceptual threshold could be by assessing the difference between the original image and the modified image. Since when the image difference increase, the perturbation is more visible and the human perceptual level also increases.

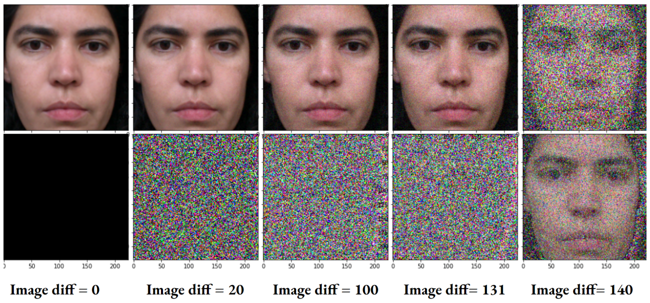

Figure 2: **Image Difference for Human Perceptual Threshold:** The top images are the modified images (original image + the bottom noise image), and the bottom images are the noises added to the original images corresponding to each of the image differences.

3.3 Ablation Study of Project Gradient Descent Attack on Gaze Estimation models

3.3.1 Default Settings

To not exceed the human perceptual threshold, while maximizing the loss, I set the default experimental setting as the following: *Initiated with Random Start, Epsilon ϵ = 5, Alpha α = 0.6, Number of Steps = 10.*

3.3.2 Influence of Amount of perturbation

Epsilon ϵ is the maximum amount of perturbation allowed on the modified image. It can be understood as the size of the ball or the bound of the size of the perturbation. Therefore, it is the only experimental setting that affects the visibility of the perturb on the images. Therefore, If ϵ is too small, the perturbation is too small from visible to regular human eyes, while loss is also small. However, If ϵ is too large, while the loss is increased, it also increases the risk of being exposed. From Figure 3, the result shows that as ϵ increases, the image becomes less discerning.

Here, the possible range of ϵ is from 0 to 255, representing the different amounts of noise allowed on the images. From

Figure 4, it can observe that while ϵ increase, the angular error also increase rapidly at first. As the noise bound increases, PGD has more attacks to explore. However, after it reached the angular error of 81.6 degrees, the noise bound continues increasing, as there is not much attack left to explore so it goes down gradually to converge around 30 degrees. Therefore, the angular error is maximum when the ϵ is 30.6, with an angular error of 81.6 degrees and a loss of 1.4. ϵ should be selected within the range of 30.6. As it is difficult to determine the visibility when the noise is too small, I compare the difference between the original and altered images to find an ϵ that maximizes the loss while keeping the stay in the human perceptual threshold within 22. Therefore, a safe range of perturbation that can be added to the original image is between 0 to 5.

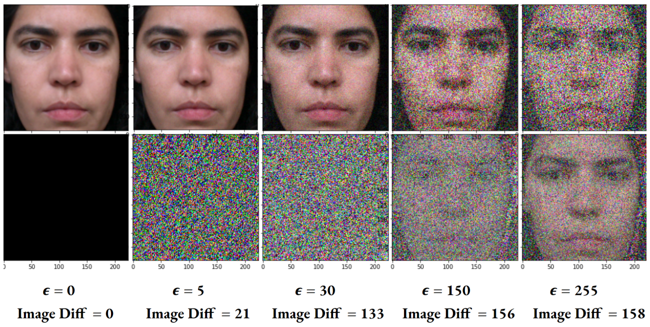

Figure 3: **Images for Different ϵ:** Default parameters: α = 0.6, Steps = 10, and random start.

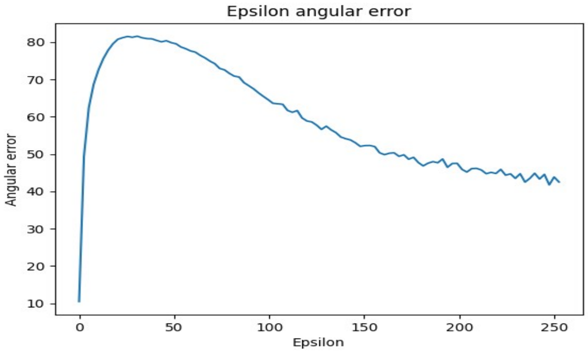

Figure 4: **Angular Error of Different ϵ:** Default parameters: Steps = 10, α = 0.6, and random start.

3.3.3 Influence of Different Stop Criteria

Stop criteria decide when to stop. Therefore, its experimental setting is important to determine whether PDG converges to the global optimum, which is the point of perturbation with the greatest loss. For this PGD attack, I set the stop criteria as a fixed number of steps. In practice, the Maximum Number of Steps is equal to 1000 or greater. Meaning if the number of steps is too little, PGD might never reach the local optimum, which leads to undesired performance. When increasing the number of steps, it is possible to lead to better performance. However, it will always have a tradeoff in computational power.

From Figure 5, it can observe that ϵ keeps the image from being too perturbs and visible to the human eyes. Even visually it is hard to detect the difference between these images, but one can still see the difference in the image difference. As the steps increased, the difference between the original image and the altered image is increasing as well, which explained the increasing loss in Figure 6. From Figure 6, it can observe that the bound set by ϵ, as when the number of steps increased, the angular error is converging. As increase the number of steps, PGD can explore more attacks within the range. As the images are hard to see difference 5, to find a number of steps that maximizes the loss while keeping the stay in the human perceptual threshold within 22, the steps that satisfied both is 10.

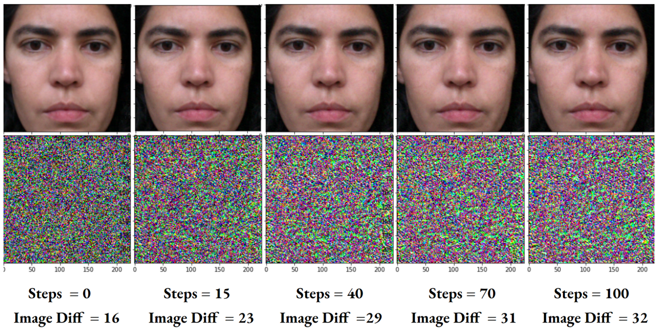

Figure 5: **Image Examples for Different Steps:** Default parameters: ϵ = 5, α = 0.6, and random start.

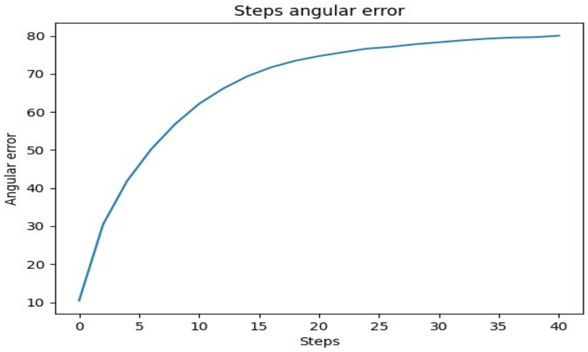

Figure 6: **Angular Error of Different Steps:** Default parameters: ϵ = 5, α = 0.6, and random start.

3.3.4 Influence of Step size

The step size α determines the length of the step, which can also consider as the learning rate of the loss function. It is important to determine whether PGD finds the maximum loss point or not. If PGD takes a large step, it will be good if the optimum is far away as PGD can explore more areas. If PGD takes a small step, it could also be beneficial if the optimum is close and can converge. However, in the worse case, if α is too large, it can diverge, and it can be slow if α is too small. As α also consider to be the learning rate, I set the range of α between 0 and 1.

From Figure 7, it can observe that when α is 0, there is still random noise generated by randomized initiation. Even though ϵ keeps the image from being too perturbs and visible to the human eyes, by seeing the difference in the image difference, it can observe that as the α increased, the difference between the original image and the altered image is increasing as well, which explained the increasing loss in Figure 8. From Figure 8, the result also revealed the bound set by ϵ, as when the α increased, the angular error is also converging. As increase the size of the step toward the gradient direction, PGD can explore more attacks within the range. Therefore, the number of steps and α are both important in aiding each other in finding the maximum loss within a certain range. As the images are also hard to see difference 7, to stay within the human perceptual threshold within 22, α is safe from 0 to 0.6. As the exponential relationship between angular error and α, the default α is 0.6.

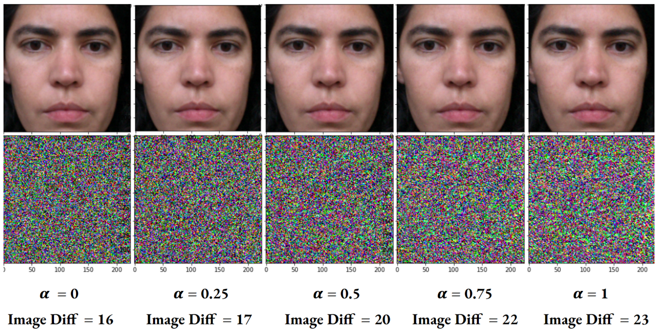

Figure 7: **Image Examples for Different α:** Default parameters: ϵ = 5, Steps = 10, and random start.

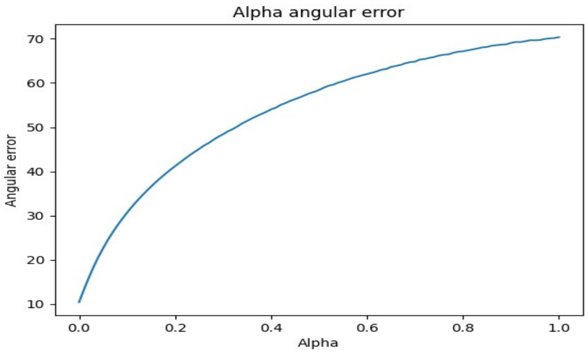

Figure 8: **Angular Error of Different α:** Default parameters: ϵ = 5, Steps = 10, and random start.

3.3.5 Influence of Random start

With a random start, I use random initialization of noise, starting at a uniformly random point in the range of ϵ bound. If there is no random start, the initial noise will be 0, and PGD will be at the center of the ϵ bound. The random start point is important since it can decide when and does PGD find the point with the highest loss or trap in a local maximum point, or whether PGD finds the global maximum or local maximum point. To visualize the change in noise with a random start, it can observe that there is more difference between the original and altered with a random start than no randomized start in Figure 9. When α = 0, there are no steps, only the initiation, so it is easier to visualize the image difference. For Table 1, the result shows that in both of the cases of already taken multiple steps in exploration and no step has been taken, the angular error is higher in the case of randomized start. Therefore, the random start at a different point each time which could lead to a higher angular error while not exceeding the ϵ bound.

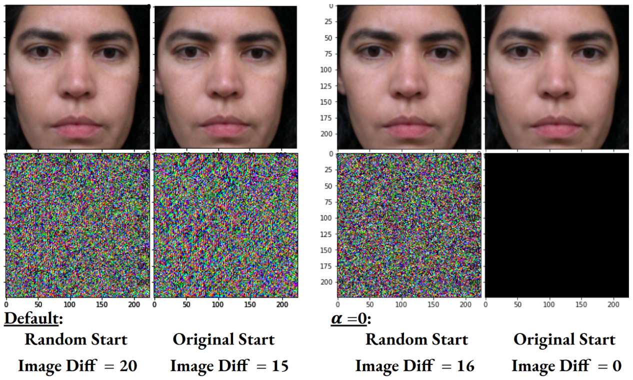

Figure 9: **Image Examples for Randomize start:** For Default, ϵ= 5, α = 0.6, Number of steps = 10. For α = 0, ϵ= 5, Number of steps = 10.

Angular Error for Random start	
Random	Average Angular error
Default:TRUE	78.709
Default:FALSE	73.051
α=0:TRUE	10.445962
α=0:FALSE	9.782

Table 1: **Angular Error for Random Start:** For Default, ϵ= 5, α = 0.6, Number of steps = 10. For α = 0, ϵ= 5, Number of steps = 10.

3.4 Projected Gradient Descent Adversarial Training on Gaze Estimation

3.4.1 PGD Adversarial Training with Classifier

From Figure 10, it can observe that the effect of PGD adversarial training on the baseline model against the PGD attack for the classification task, in which both losses of the original sample and adversarial sample decreased and approached the training loss closely, even the original sample go below the training loss.

3.4.2 PGD Adversarial Training on Gaze Estimation

For the experimental setting, I test the trained baseline model against the PGD attack with the same parameter as in training. The result of the experiments can be seen in Figure 11 and Table 2 with *20 epochs: Full AdvTrain*. From Figure 11, it can

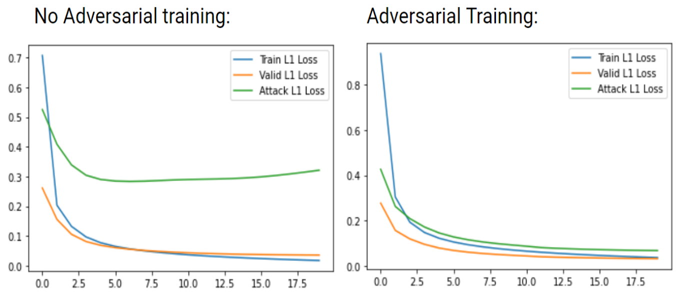

Figure 10: **PGD Adversarial Training on MINST dataset for Image Classification:** *Valid* means the original sample. *Attack* means the modified sample. The y-axis is the angular error. The x-axis is the number of epochs.

observe that the angular errors for the adversarial samples and original samples are decremental as the number of epochs is increasing. As the result from Table 2, the angular error for the adversarial samples with no PGD adversarial training is sufficiently decreased when applying PGD adversarial training with a 67.973 angular difference, from the original 78.709 to 10.736 angular error. Therefore, PGD adversarial training has some defense against the same PGD adversarial attack. However, it is not as effective as the classification task, as the angular error of both the adversarial samples and original samples did not go below or equal to the baseline.

From increasing the epochs to 60 in Table 2 with *60 epochs: Full AdvTrain*, the neural network is not converging to the baseline compared to Figure 10. One possibility might be because of the training time is not enough, since for gaze estimation and adversarial training, it usually takes more epochs to train. In addition to epochs, another reason could be because of modeling. Therefore, in the following sections, I will also experiment with different data argumentation, model capacity, and loss functions to learn their consequence on PGD's adversarial training in the hopes to improve the performance even further.

Angular Error for LetNet			
	Train	Test: Original	Test: Altered
No Adversarial Train	2.098	8.838	78.709
20 epochs: Full AdvTrain	9.118	8.750	10.826
20 epochs: Half AdvTrain	7.314	9.263	12.372
30 epochs: Full AdvTrain	8.670	9.583	10.989
60 epochs: Full AdvTrain	8.164	9.163	11.162

Table 2: **Angular Error for LetNet:** *Train* means the angular error of the sample being trained. *Test: Original* represents the angular error of the original samples in testing. *Test: Altered* represents the angular error of the adversarial (attacked) samples in testing. *Full AdvTrain* represents the standard adversarial training with implementation detailed in Section 2.3. The experiment and implementation on *Half AdvTrain* are explained in Section 3.4.3.

3.4.3 Impact of Data Argumentation

To explore the impact of the number of adversarial samples in the training set on the performance of adversarial training, I will compare the experiments of replacing all the 42000 images on the training set with its adversarial counterpart by only replacing half of the training set. For these two methods, I abbreviate them as full adversarial training and half adversarial training. For the half adversarial training, half of the training set is original samples, while the other half is adversarial samples. The experiment result can be seen in Table 2 as *20 epochs: Full AdvTrain* and *20 epochs: Half AdvTrain*. From decreasing the adversarial samples and increasing the number of original samples in the training set, the result shows that the performance worsens as the angular error increases for both the original samples and the adversarial sample in the testing set.

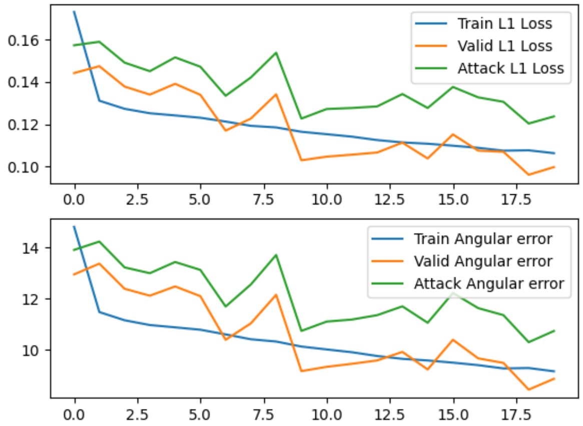

Figure 11: **PGD Adversarial Training With PGD Attack:** *Valid* means the original samples. *Attack* means the adversarial samples. The y-axis is the angular error. The x-axis is the number of epochs.

In conclusion, First, including the original samples in the training set does not improve the angular error for original samples in the testing set and is not effective in the training set. Second, the number of adversarial samples could affect the PGD adversarial training performance. As the number of adversarial samples increases, the angular error for both the adversarial and original samples decreased.

3.4.4 Impact of Model Capacity

To explore the impact of the model capacity in the training set on the performance of adversarial training, I will perform the PGD adversarial training with AlexNet, which is a more complex model than LetNet and have more layers. Comparing the performance of the LetNet in Table 2 and the performance of the AlexNet in Table 3, the result shows that the model has more defense against the adversarial attack when without the adversarial training is AlexNet, while the adversarial attack is more successful with LetNet, as the *Test: Altered* in *No Adversarial Train* is larger for LetNet. However, the PGD adversarial training seems more successful with LetNet than AlexNet. For PGD adversarial training on AlexNet, not only does the improvement of the average angular error from no adversarial training to adversarial training is smaller than LetNet, and the average angular error with adversarial training is higher than LetNet, but also the angular error for the original samples seem to perform worse with PGD adversarial training.

In conclusion, model capacity has an impact on robustness. As the capacity increase, it becomes more resistant to the adversarial attack. However, the performance of the adversarial training is not necessarily becoming more effective as the model capacity increase.

Angular Error for Other Experiment			
	No AdvTrain: Altered	AdvTrain: Original	AdvTrain: Altered
AlexNet	34.949	13.907	13.525
L2Loss	79.759	9.586	11.625

Table 3: **Angular Error for Other Experiment:** *No AdvTrain: Altered* Angular error of the adversarial samples when no adversarial training. *AdvTrain: Original* Angular error of the non-altered samples with adversarial training. *AdvTrain: Altered* Angular error of the adversarial samples with adversarial training

3.4.5 Impact of Loss Function

To improve the PGD adversarial training on Gaze estimation, I explore two different loss functions, L1: $Loss(x, y) = |x - y|$, and L2: $Loss(x,y) = (x - y)^2$. L1 Loss Function as it is not affected by the outliers than L2 Loss. From Table 3, the result shows that the adversarial attack is more successful with L2 Loss by a little. For adversarial training, the improved degree is roughly the same with both loss functions. Thus, the table reveals that L2Loss is more susceptible to adversarial samples than L1Loss as L2 is more sensitive to outliners. In addition, PGD adversarial training has the same effect with both loss functions. However, since L1Loss is more robust against the adversarial samples and leads to better results, L1Loss will be our default loss function.

3.4.6 Against Other Adversarial Attack

To explore the performance of PGD adversarial training against other attacks, I have selected a few attacks that have similarities to PGD mentioned in Section 2.4 and Appendix A. From them, some are simpler than PGD (GN, FGSM, BIM, FFGSM), while some are more complicated (EOTPGD, MIFPGD, NIFPGD, SINIFGSM, VMIFGSM, VNIFGSM), to evaluate whether PGD adversarial training is able to against other attacks.

From Table 4, I notice that PGD adversarial training is an even better defense against certain other attacks than PGD itself (VMIFPSM, VNIFPSM), and it is most effective against VNIFPSM attack. Comparing the effect of the attack without adversarial training in Figure 5, the result shows that PGD adversarial training is not able to against the GN attack among the attacks, while is least effective on EOTPGD attack. For most attacks, even if they did not reach the baseline, PGD adversarial training still improved their average angular error significantly. In conclusion, PGD adversarial training is able to defend against other adversarial attacks as well and possesses a certain level of attack generalism.

Angular Error for PGD Attack Generalism			
	AdvTrain: Original	AdvTrain: Altered	Improve ?
GN	13.513	13.513	No
FGSM	8.773	10.688	Yes
EOTPGD	10.439	12.234	Yes
MIFGSM	10.055	11.5818	Yes
NIFGSM	9.207	10.520	Yes
VMIFGSM	8.97	9.142	Yes
VNIFGSM	8.303	9.960	Yes

Table 4: **Performance of PGD Adversarial Training with Other Attacks:** *Improve ?* Yes when angular error lower than *NoAdvTrain:Altered* in Table 5.

3.5 Other Adversarial Training Methods on Gaze Estimation

To explore other adversarial training methods possibility that could lead to better performance than PGD adversarial training, I will explore the following methods mentioned in Section 2.4 and Appendix A that also exploit the gradients of a neural network, in the ranking from simple to complex: GN, FGSM, BIM, FFGSM, PGD, PGD2, EOTPGD, MIFPGD, NIFPGD, SINIFGSM, VMIFGSM, VNIFGSM.

3.5.1 Implementation Details

As different experimental parameters of each attack have a different level of effect on the performance, in order to construct a comparable experimental environment, I have also imposed the human perceptual threshold for each adversarial attack and training, meaning their adversarial sample cannot exceed the human perceptual threshold set in Section 3.2. Therefore, for each of the adversarial attacks, their experimental settings can be seen in Table 7 in the appendix.

3.5.2 Performance result

Angular Error for Adversarial Attack and Training					
	No Adv Train: Altered	Attack ?	Adv Train: Original	Adv Train: Altered	Improve ?
GN	7.012	No	7.721	7.722	No
FGSM	39.654	Yes	9.447	11.522	Yes
BIM	10.763	Yes	9.527	9.527	Yes
FFGSM	9.818	Yes	10.924	10.963	No
PGD2	73.378	Yes	12.736	13.416	Yes
EOTPGD	81.324	Yes	7.331	9.440	Yes
MIFPGD	58.136	Yes	7.575	9.397	Yes
NIFPGD	71.876	Yes	10.444	12.559	Yes
SINIFGSM	52.818	Yes	7.970	9.867	Yes
VMIFGSM	46.357	Yes	9.010	10.904	Yes
VNIFGSM	61.688	Yes	8.433	10.417	Yes

Table 5: **Performance for Adversarial Attack and Training:** *Attack ?* Yes when the angular error is above the baseline model. *Improved ?* Yes when the angular error below the *No AdvTrain: Altered.*

First, for the adversarial attack, the results show that merely adding random noise to the image is not enough to change the output of the model and even might perform better than before, which can see from GN. For the adversarial attacks that used the gradient of the model, the performance is varied, regardless of the complexity of the algorithm or the iteration of steps taken. From Table 5, the result reveals that EOTPGD is the most effective attack, while FFGSM is the least effective.

Second, for adversarial training, the result reveals that merely randomizing the noise does not affect attacking the model or defense against any attacks, as GN even performs worse than before. For the adversarial training that used the gradient of the model, the performance is also varied, regardless of the complexity of the algorithm or the iteration of steps taken, in which the improvement is varied from roughly 1.2 degrees to 78 degrees, and FFGSM even performs worse than being attacked. Therefore, the results show that FFGSM is not as effective in defense and attack, as it just merely increases 1 angular error degree when attacking. From Table 5, the outcome unveils that for the adversarial attack that is not that effective in attack, their adversarial training result is also less than ideal, which can observe that BIM improved the least. Among all the experimented attacks, EOTPGD improved the most even more than PGD, which also has the least average angular error.

3.5.3 Attacks Generalism against PGD Performance result

Angular Error for Attacks Generalism			
	AdvTrain: Original	AdvTrain: Altered	Improve ?
GN	8.996	50.685	No
FGSM	10.432	12.805	Yes
PGD2	13.306	13.417	Yes
EOTPGD	10.093	11.652	Yes
MIFPGD	9.469	11.557	Yes
NIFPGD	12.539	9.663	Yes
SINIFGSM	7.549	10.475	Yes
VMIFGSM	9.123	11.308	Yes
VNIFGSM	8.840	11.008	Yes

Table 6: **Performance of Adversarial Training with PGD Attacks:** *AdvTrain: Original* Angular error of the non-altered samples with adversarial training. *AdvTrain: Altered* Angular error of the adversarial samples with adversarial training

Third, for testing the attack generalism of each adversarial training, performing adversarial training against other adversarial attacks, the result shows that depending on the different adversarial attacks, the performance of the adversarial training changes in the Table 6, which SINIFGSM has the least average angular error and PGD2 have the most when against the PGD attack. An interesting observation is that some of the simpler versions that PGD adversarial attacks build on (GN, BIM, FFGSM) seem to fail within the three experiments, in which FGSM is successful against FGSM but fails against the PGD attack. Currently, I hypothesize that simple adversarial training is not as able against more complex adversarial attacks occasionally. However, we do need to verify it by experimenting with other adversarial attacks in future experiments.

4 Responsible Research

4.1 Scientific Integrity

There are two potential ethical aspects related to this project. First, the MPIIFaceGaze dataset that we used with the extra human facial landmark annotation and the face regions accessible for 37,667 face images [22] might consider containing sensitive information which might violate the ”universalism” of Mertonian norms, which ”The evaluation of research results should be based entirely on impersonal criteria and be without any form of prejudice against nationality, gender, race, personal characteristics, etc.” [17]. However, for privacy, the dataset had been preprocessed and only released the face region and blocked the background in images [22], and there is no potential in releasing any sensitive information regarding the subjects. Therefore, the dataset contains no information that could violate one’s privacy and is only used for gaze estimation.

Considering specific applications of gaze estimation, currently, gaze estimation has been considered to diagnose brain trauma [1]. One of the consequences of adversarial attacks on these applications could be misdiagnosed in the case of false positives. Therefore, following the ”communism ” of Mertonian norms [17], meaning the research result is public property and is available to all, the study of adversarial training could be used as precautionary for stakeholders like hospitals.

4.2 Reproducibility

The experiment and result of this research project are totally reproducible by following the section 2 and 3 of the project.

To reproduce the baseline performances, the dataset, the neural network structures, and the default training details are all included in the section 3.1. The standard PGD attack that we used is from [10] and the modification that we made for this research is detailed in section 2.2. For the PGD adversarial training, we follow the method in [15] and follow with modification detailed in section 2.3. For other adversarial attacks and training, the detailed implementation and parameter setup for each attack are described in section 3.5.

5 Conclusions and Future Work

5.1 Conclusion

Since the lack of research has been done in adversarial attacks and training on gaze estimation, the goal of this project is to explore how these adversarial samples affect the gaze estimation’s performance and whether the adversarial training elevates the effect of these adversarial attacks. From experimenting with the different experimental settings of the PGD attack, the effect of ϵ bound can be observed on both the attack visibility and the increasing angular error. ϵ is the only setting that decides the human perceptual level, and as ϵ increases, the human perceptual level increase. However, as the ϵ bound increased, we have more attacks to explore. After PGD already reached a certain loss, there is not much attack left to explore so it goes down gradually to converge. While α and the number of steps do not impact the human perceptual level as much, they all converge within the bound set by ϵ when they increase. Lastly, we also proved that the random start usually gives out better results as mentioned in [15].

From experimenting with PGD adversarial training against other attacks, the result shows that even though PGD adversarial training can be against certain adversarial attacks and possesses a certain level of attack generalism, the performance is still not ideal as not converging to baseline. One possibility might be because of the training time is not enough, since for gaze estimation and adversarial training, it usually takes more epochs to train, which could be a future question to investigate. From experimenting with different data augmentation, model capacity, and activation functions, their performance reveals that increasing adversarial training samples and using the L1 Loss function could lead to better performance. However, more future experiments could be done with different model capacities like ResNet, more loss function, and increase the training dataset by injecting different types of adversarial samples instead of one.

From experimenting with other adversarial training, EOTPGD proves to be even more effective than PGD adversarial training for PGD attacks, while GN is proven to be not effective for attack and defense. However, depending on the different adversarial attacks, the performance of the adversarial training changes. In order to prove the full attack generalism of some of the adversarial trainings that prove effective and the intriguing observation that simple adversarial training seems weaker against stronger adversarial attacks, future experiments are needed by including more types of attacks, other than PGD. Lastly, another future work could be done to assess the current MSE method that determines the human perceptual threshold, by involving more testing involved with more participants.

A Other Adversarial Attacks

- **GN** Gaussian Noise. A one-step method, which adds random Gaussian noise.
- **FGSM** Fast Gradient Sign Method. A one-step method, which only takes one step toward the gradient direction [7].
- **BIM** Iterative-FGSM. A multiple-step method, which is a simple improvement to FGSM. They suggest applying the same step as FGSM multiple times with a small step size and clipping the pixel values of intermediate results after each step to ensure that they are in an ϵ-neighbourhood of the original image [12].
- **PGD2** A multiple-step method, which PGD without random start.
- **EOTPGD** Expectation Over Transformation PGD. A multiple multiple-step method, which builds on top of PGD, within each step, iterating with a number of models to estimate the mean gradient [23].
- **FFGSM** (Fast's FGSM)An one-step method, which is one step in PGD attack. Including random start, one step toward the gradient direction, and the noise is projected back to ϵ [20].
- **MIFGSM** Momentum Iterative FGSM. A multiple-step method, which is a momentum-based Iterative Fast Gradient Sign Method [3].
- **NIFGSM** Nesterov Iterative FGSM. A multiple-step method, which adapts Nesterov accelerated gradient into the Iterative Fast Gradient Sign Method [14].
- **SINIFGSM** Scale-Invariant attack Method. A multiple multiple-step method, which Scale-Invariant Iterative Fast Gradient Sign Method, calculates the sum of the gradients over the scale copies of the input image [14].
- **VMIFGSM** Variance Tuning MIFGSM. A multiple multiple-step method, which uses variance tuning with momentum-based Iterative Fast Gradient Sign Method. At each iteration for the gradient calculation, we consider the gradient variance of the previous iteration to tune the current gradient [19].
- **VNIFGSM** Variance Tuning NIFGSM. A multiple multiple-step method, which uses variance tuning with Iterative Fast Gradient Sign Method using Nesterov accelerated gradient [19].

Experimental Setting for Adversarial Attack							
	Alpha	Epsilon	Steps	Random	decay	beta	other
GN	n/a	3	n/a	n/a	n/a	n/a	n/a
FGSM	n/a	4	n/a	n/a	n/a	n/a	n/a
BIM	0.9	4	10	n/a	n/a	n/a	n/a
FFGSM	0.9	3	n/a	n/a	n/a	n/a	n/a
PGD2	0.9	8	10	Yes	n/a	n/a	n/a
EOTPGD	0.6	6	10	n/a	n/a	n/a	2
MIPGD	0.4	8	10	n/a	1	n/a	n/a
NIPGD	0.4	8	10	n/a	1	n/a	n/a
SINIFGSM	0.39	8	10	n/a	1	n/a	5
VMIFGSM	0.4	8	10	n/a	1	3/2	5
VNIFGSM	0.5	8	10	n/a	1	3/2	5

Table 7: **Experimental Setting for Adversarial Attack**

References

[1] Military's eye-tracking system can provide 'third hand' for trauma surgeons, 2012.

[2] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks, 2017.

[3] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Xiaolin Hu, and Jun Zhu. Discovering adversarial examples with momentum. CoRR, abs/1710.06081, 2017.

[4] Marin Dujmović, Gaurav Malhotra, and Jeffrey S Bowers. What do adversarial images tell us about human vision? eLife, 9:e55978, sep 2020.

[5] Gamaleldin Elsayed, Shreya Shankar, Brian Cheung, Nicolas Papernot, Alexey Kurakin, Ian Goodfellow, and Jascha Sohl-Dickstein. Adversarial examples that fool

both computer vision and time-limited humans. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 31. Curran Associates, Inc., 2018.

[6] Joao Gomes. Adversarial attacks and defences for convolutional neural networks, 2018.

[7] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015.

[8] Jan Philip Göpfert, André Artelt, Heiko Wersing, and Barbara Hammer. Adversarial attacks hidden in plain sight. In Lecture Notes in Computer Science, pages 235–247. Springer International Publishing, 2020.

[9] Saumya Jetley, Nicholas A. Lord, Namhoon Lee, and Philip H. S. Torr. Learn to pay attention, 2018.

[10] Hoki Kim. Torchattacks: A pytorch repository for adversarial attacks. arXiv preprint arXiv:2010.01950, 2020.

[11] Oscar Knagg. Know your enemy: How you can create and defend against adversarial attacks, 2018.

[12] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world, 2017.

[13] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. Defense against adversarial attacks using high-level representation guided denoiser, 2018.

[14] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E. Hopcroft. Nesterov accelerated gradient and scale invariance for improving transferability of adversarial examples. CoRR, abs/1908.06281, 2019.

[15] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2019.

[16] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks, 2016.

[17] J. B. Morrell. Sociology of science - the sociology of science. theoretical and empirical investigations. by robert k. merton. ed. by norman w. storer. chicago and london: University of chicago press, 1973. pp. xxxi 605. £6.25. The British Journal for the History of Science, 8(1):70–71, 1975.

[18] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses, 2020.

[19] Xiaosen Wang and Kun He. Enhancing the transferability of adversarial attacks through variance tuning, 2021.

[20] Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training, 2020.

[21] Mingjie Xu, Haofei Wang, Yunfei Liu, and Feng Lu. Vulnerability of appearance-based gaze estimation, 2021.

[22] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. It's written all over your face: Full-face appearance-based gaze estimation. CoRR, abs/1611.08860, 2016.

[23] Roland S. Zimmermann. Comment on ”adv-bnn: Improved adversarial defense through robust bayesian neural network”. CoRR, abs/1907.00895, 2019.