Fool Me Once: The Viability of Adversarial Images Over Time

Avi Gulati

May 4, 2024

1 Introduction

Neural networks have shown immense capabilities in image classification tasks over the last decade. The ImageNet database, which features over 1 million images for training and over 50,000 for validating, ([KSH12]), challenged researchers around the world. In 2012, a convolutional neural network called AlexNet succeeded significantly with an error of only 15%, 10 points lower than the runner up ([Kos18]). Model performance like that of AlexNet or Resnet has demonstrated that neural networks are highly effective at image classification.

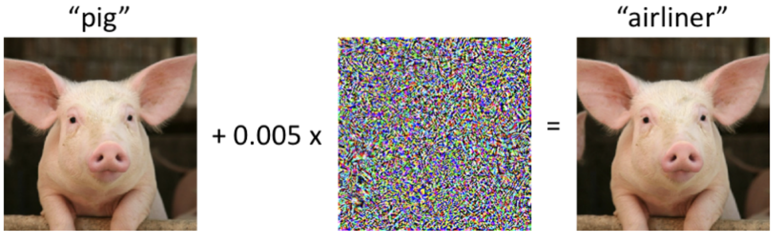

Figure 1: Adversarial Image Example: Classic Case of Pig as Airline. Courtesy of [EIS+19] lab

This efficacy is still limited by many factors, one of which is exposure to adversarial images. In the image above, provided courtesy of the Madry lab at MIT–a preeminent robustness research center in the country, a neural network correctly classifies an image as a pig. However, when that image is exposed to even the slightest of noise, where each pixel in the image is on the interval [0,1] and the noise is less than 0.005 for each pixel, the network misclassifies the image as an airliner. This is true even when there appears to be no difference between the perturbed image and the original image to the human eye; however, a clear difference exists for the classifier. It should be noted that adversarial images have to be intentionally generated. The addition of noise in the image above appears to be random noise, but that is not the case. Neural networks actually appear to do well even when images are perturbed randomly. They usually misclassify “only for specifically crafted perturbations” ([MS18]).

These specifically crafted perturbations are done accordingly. To train a classifier, we observe the following fundamental equation, where the goal is to change θ, namely the model's weights, such that the sum of the loss for each image in the data is minimized.

$$\min_{\theta} \sum_{x} \text{loss}(x, \theta). \tag{1}$$

Generating an adversarial image is close to the opposite of the model training goal. Instead of minimizing the loss, we choose to maximize the loss function by holding the model's weights at given and instead changing the image.

$$\max_{\delta} \text{loss}(x + \delta, \theta). \tag{2}$$

We observe the direction of steepest ascent with respect to loss by changing the image's pixel values, and then step by the δ value in that direction. To generate an adversarial image, why not make all the pixels black and then the model will surely misclassify the image? That would defeat the purpose of adversarial training to robustify a model in the first place, so perturbations are kept within a budget constraint, or eps for ϵ. In the scope of this research, eps is a maximum change to an input image in the L2 norm space. Given this maximum change, to generate the adversarial images, we change the image by small amounts (called δ or the attack step size) at each iteration to maximize the "adversarialness" of the image.

These adversarial images are then fed into the network during training to robustify the network. Current research in robustness is focused on novel methods for image generation like curriculum or regularization based approaches and the classic "tradeoff between robustness and generalization [as an] intrinsic limitation of adversarial training" ([BL21]). In this research project, I choose to explore the longevity of adversarial examples over time, an underexplored theme in the literature. Given the adversarial images that are generated at each training loop, I want to monitor how long these images continue to deceive a model across subsequent training epochs. Exploring the viability of adversarial examples over time helps us understand how training can influence a model's vulnerabilities and further robustify networks.

2 Methods

My code is located in this colab file. My code for changing the robustness package's train.py file is located in this repository.

Figure 2 shows a standard pipeline for adversarial training. Given a dataset, N images are sampled as they are and M images are sampled adversarially. These images are then combined into a single batch and the model is trained on them before this process is repeated.

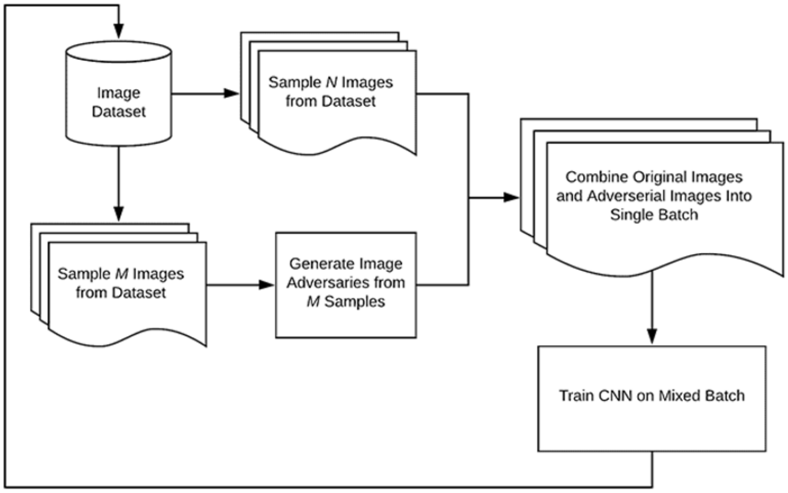
Image Dataset
Sample *N* Images from Dataset
Combine Original Images and Adverserial Images Into Single Batch
Sample *M* Images from Dataset
Generate Image Adversaries from *M* Samples
Train CNN on Mixed Batch

Figure 2: Robustness Training Pipeline, Source: Adrian Rosebock

To track the life cycle of an adversarial image, after the M images are sampled adversarially, each image is retained. At the end of each epoch, the adversarial images are tested to see if the model is able to correctly classify the images. In the scope of this research, adversarial images are only retained after the first epoch and then tested after each subsequent epoch.

This research makes significant use of the Madry lab's open source robustness library that facilitates adversarial training. We change the code in the training loop of the robustness package to retain data about an adversarial image. The dataset used is the CIFAR-10 dataset which contains 32 by 32 colors images across 10 classes which are airplane, automobile, bird, cat, dog, deer, frog, horse, ship truck. I sample 10% of the 60,000 image dataset because of limited compute and time resources. For adversarial image storage, I retain 200 images of the 5000 image training set Even with 10% of the dataset, adversarial training with an A100 GPU takes over 2 hours and 40 compute units in Google Collab (100 units cost $10). This research was unfortunately not compatible with a class-provided

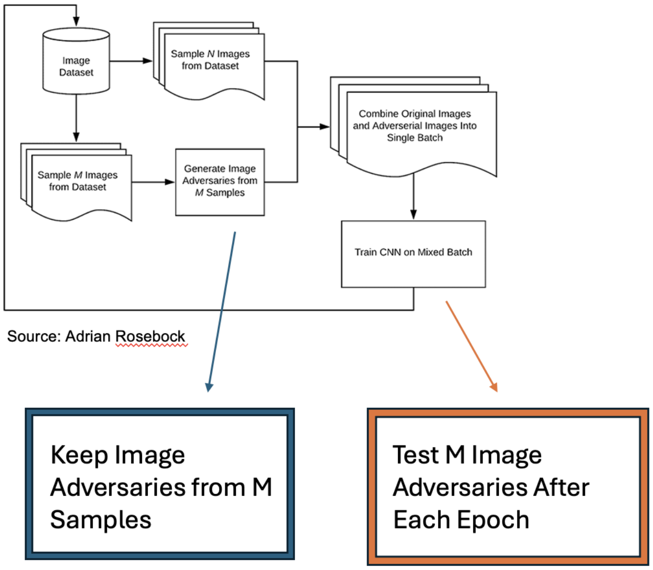
Image Dataset
Sample *N* Images from Dataset
Combine Original Images and Adverserial Images Into Single Batch
Sample *M* Images from Dataset
Generate Image Adversaries from *M* Samples
Train CNN on Mixed Batch
Keep Image Adversaries from M Samples
Test M Image Adversaries After Each Epoch

Source: Adrian Rosebock

Figure 3: Robustness Training Pipeline Adjusted for Finding Lifetime of Each Image

JupyterHub compute environment because of underlying package dependencies that were inaccessible like CUDNN and Pytorch incompatibilities that could not be resolved.

The model used for adversarial training is a pretrained, non-robust Resnet 50 on the CIFAR 10 dataset. A few of the important parameters to note for adversarial training are the EPS of 0.25, 7 attack steps (ie: 0.25/7), a learning rate of 0.1 for adversarial image generation, and non-targeted images. Targeted images are when the loss is maximized for the correct label and minimized toward an incorrect class label. Non-targeted adversarial training implies maximization of loss regardless of other classes.

When adversarial images are retained at the end of the first epoch, the following is stored in a set: the original image, the adversarial image, the original image's label, and the epoch distance of the image. The epoch distance is incremented at the end of each epoch should the model continue to classify the adversarial image not according to the original image's label. As a brief point of reflection, it would have behooved us in our methodology to have included a list of the misclassified labels of the model to determine whether or not the model continues misclassifying the image as the incorrect label repeatedly or whether these misclassifications appear to vary between classes over epochs.

3 Results

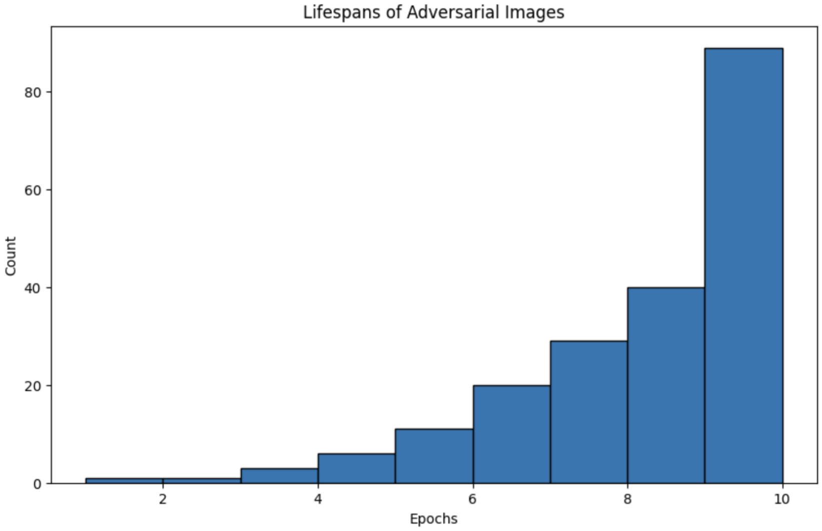

Figure 4: Distribution of 200 Adversarial Images Lifespans Over 10 Epochs

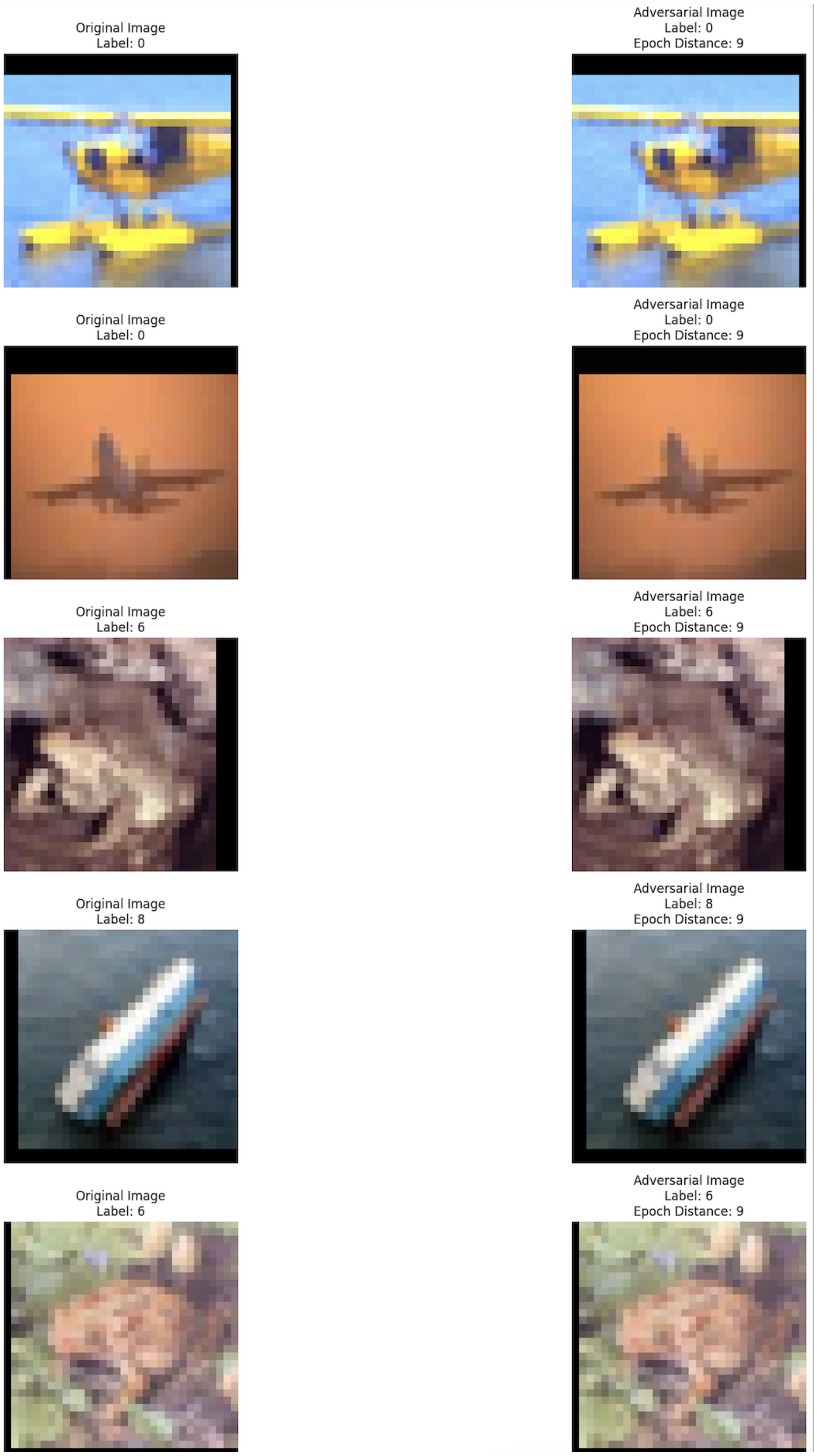

Figure 5: 5 CIFAR-10 Adversarial Images with Long Lifespans

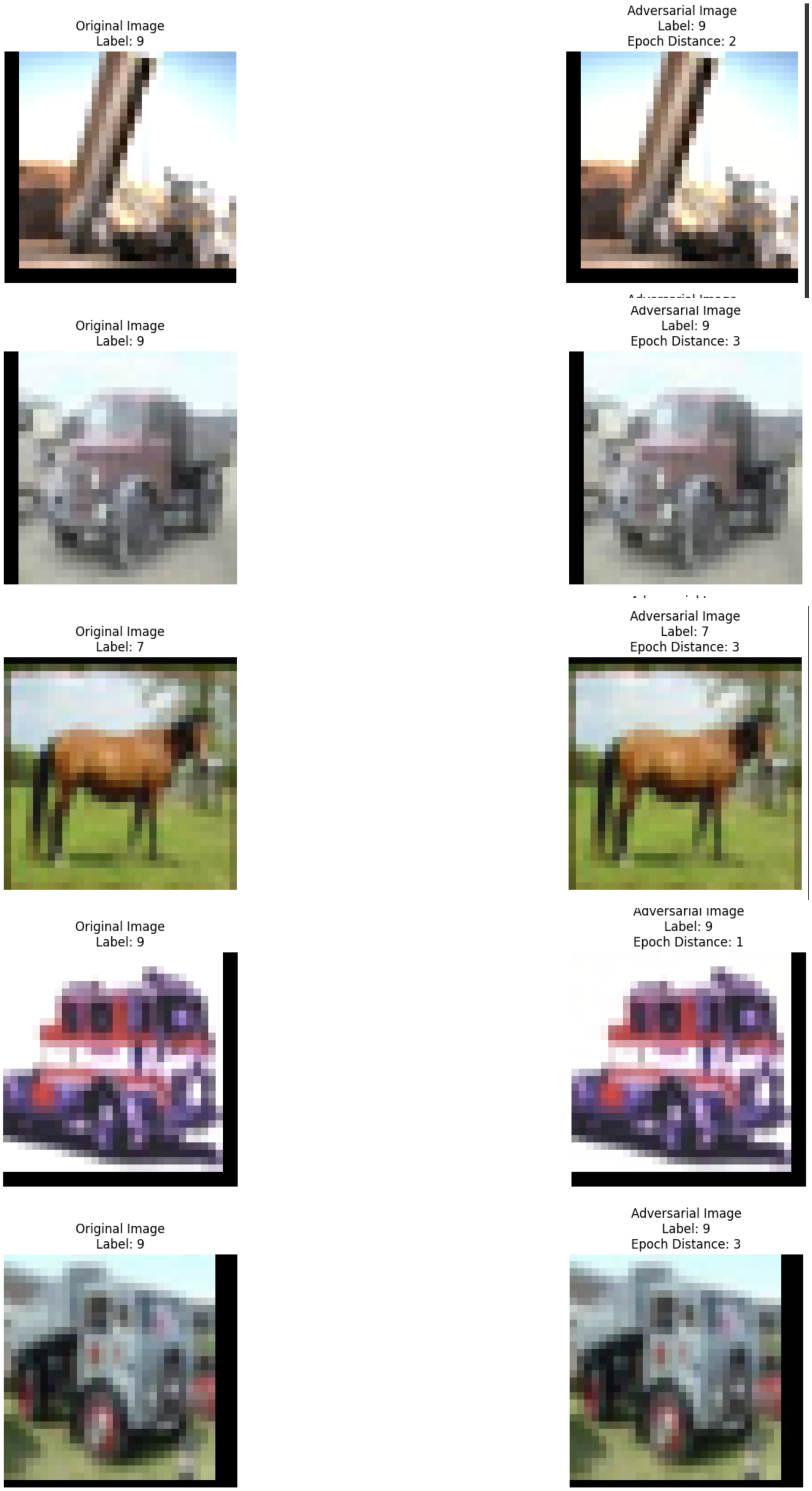

Table 1: 5 CIFAR-10 Adversarial Images with Short Lifespans

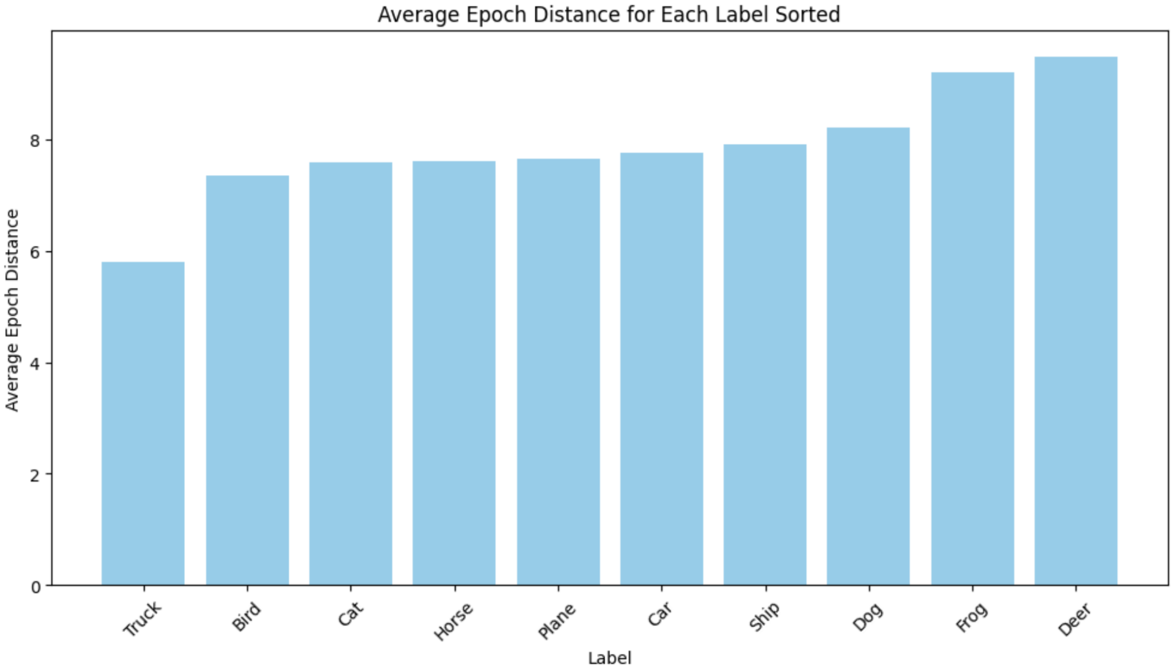

Figure 6: Relationship between Lifespan and Label

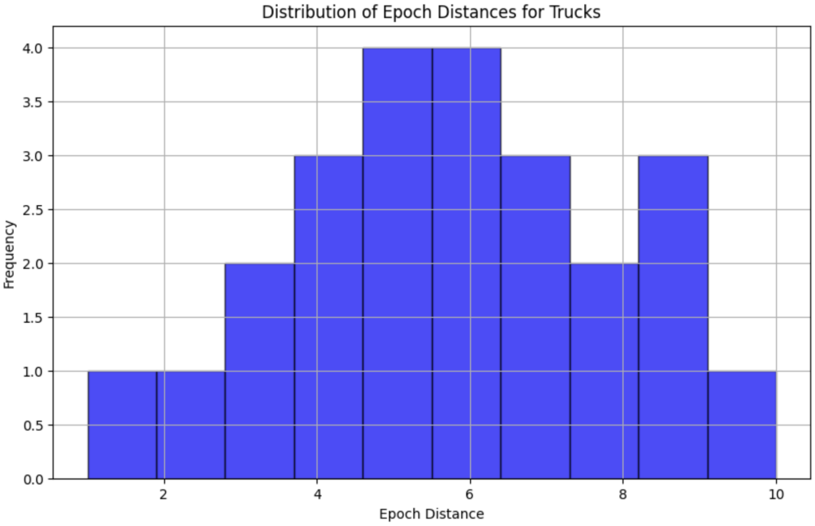

Figure 7: Distribution of Lifespan for Truck Images

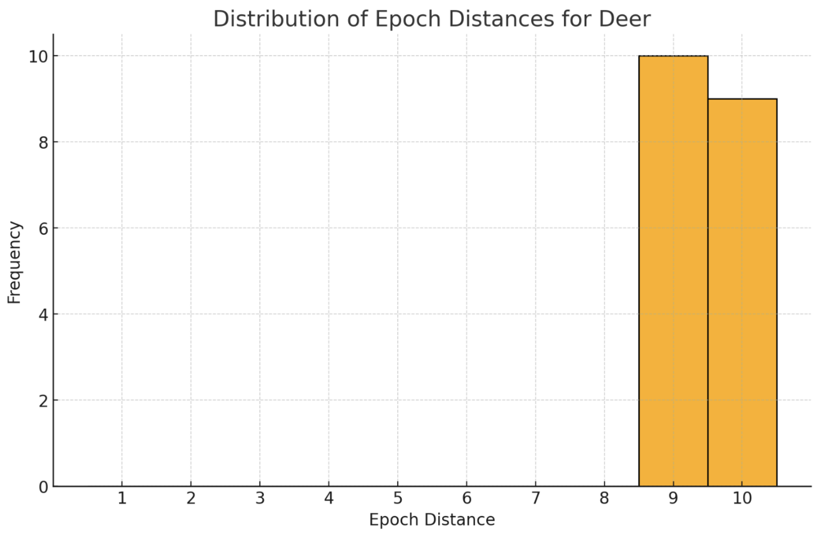

Figure 8: Distribution of Lifespan for Deer Images

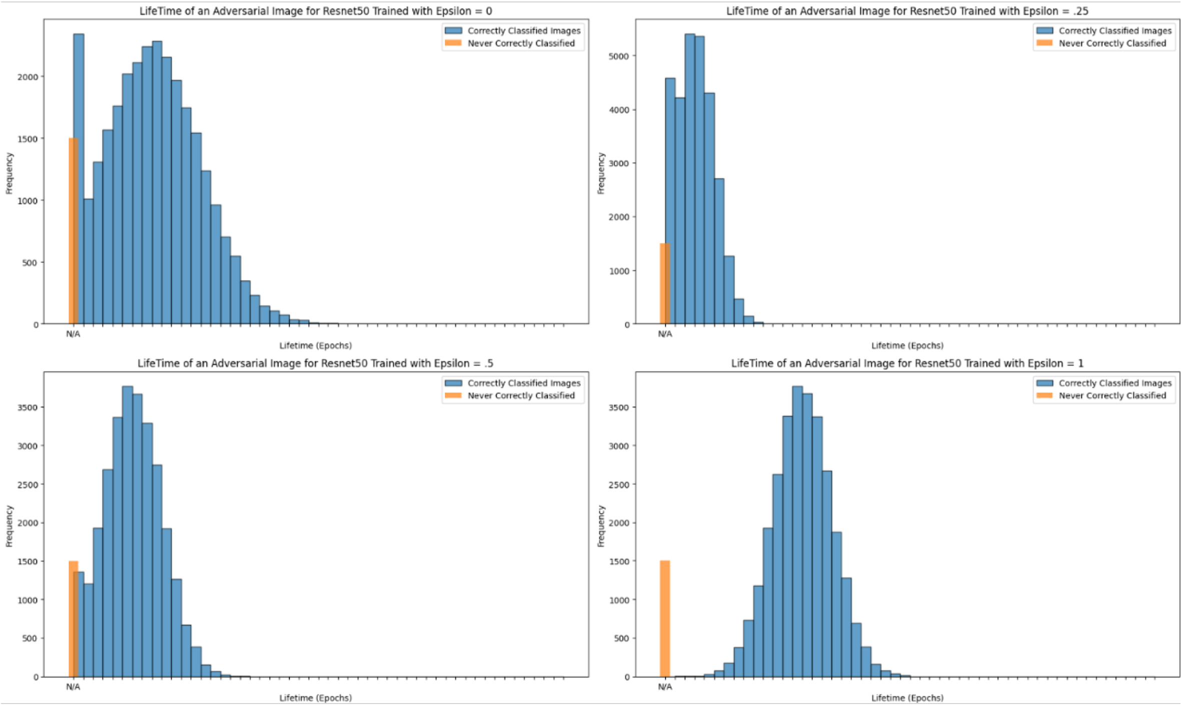

Figure 9: Hypothetical Attack-EPS: 0.25 on Different Pre-Trained Models Provided in Robustness Package

4 Discussion

Figure 4 shows the aggregated lifespans for 200 adversarial images generated in the first epoch. These images were generated during every 25th image in the 5000-image training data. We observe, in this left skewed distribution, a peak at 10, which suggests that around half of the adversarial images had long lifespans and were not correctly classified within this model's training loop. The other approximately 100 images are correctly classified within the 10 epochs. Of course, it makes sense that no adversarial image would have an epoch distance of 0; otherwise, it wouldn't be an adversarial image in the first place since images are generated in the epoch with the purpose of being misclassified.

The result with a peak at 10 is not surprising. It takes about 50 epochs to get a Resnet to classify Cifar-10 with accuracy. I would estimate that 10 epochs, as is evidenced by this peak at 10, was too short for adversarial training. If I had more resources and funding, I would have trained this for further epochs.

The next interesting question that emerges is then why some adversarial images weren't correctly classified by the end of the training loop and why some adversarial images only had a lifespan of 1, 2, or 3 epochs. Figure 5 and Table 1 begin the answer to this question. In Figure 5, I sampled 5 images from the 80 images that had long lifespans. From top to bottom, these five images are plane, plane, frog, ship, and frog. In Table 1, the 5 images with the shortest lifespans are truck, truck, horse, truck, and truck. Interested by the frequency of truck among images with shortest lifespans, I created Figure 6, a bar chart that demonstrates the relationship between lifespan and label. Indeed, we find that the truck has the shortest average epoch distance, or lifetime whereas the deer has the longest. I surmise that the truck is the most distinctive category among the 10, where with 32 pixels by 32 pixels, the rectilinear lines that characterize a truck differ greatly from any of the other labels that can blend into each other (like horse into deer or dog into cat). The cab of a truck usually features a clear triangular top and the body of the truck is rectangular. To the human eye, the truck appears to be far different from the car or the plane compared to the amount fo difference between a deer and a horse. This is of course a conjecture, and I would be curious to examine this in further detail with explainability methods like Gradcam or targeted adversarial training. Needless to say, there is clearly something that the model is able to learn fast about trucks compared to deer.

We show the distribution of lifespans for truck images in Figure 7 and for deer images in Figure 8. The latter is centered around 5.5, bearing a normal distribution whereas the deer lifespans are either 9 or 10. This means that half of the deer images were correctly classified by the end of training. If I had the opportunity and resources to run more epochs, I would be curious about which deer images would have lifespans of 10, 11, 12, or more and which ones would never be classified correctly.

To answer this question there are many variables that affect the lifespan.

- Perturbation Complexity
- Epoch Number
- Targeted vs Non-Targeted

For the first, perturbations can either be superficial or take advantage of underlying problems with the model. For the second, my hypothesis is that adversarial images introduced in earlier epochs are more likely to have shorter lifespans than ones introduced in later epochs since learning probably happens the most at the beginning. For the third one, it makes sense that non-targeted images have shorter lifespans. It's easier to correctly classify a generally-perturbed dog image than a dog image that has been perturbed to look like a wolf.

On further topics of discussion, adversarial images are believed to be generalizable. Adversarial images to a Resnet work against a VGG. An important question that emerges is whether or not the lifespans of adversarial images are also consistent across model architectures. Additionally, are these lifespan lengths model-dependent or dataset-dependent?

Finally, in Figure 9, I explore distributions of lifetimes across different pretrained models, though these histograms were created according to theory. Such verification would require at least 96 hours of compute with a Collab A100 GPU. With an Attack-EPS of 0.25 during adversarial training, we would observe that the shortest lifespans occur in the pretrained Resnet50 model robust to 0.25 and 0.5, whereas the longest lifespans would be for the pretrained non-robust Resnet50 and the pretrained Resnet with EPS 1. The orange bar in each histogram represents the images that would never be correctly classified.

5 Other Projects and Learnings

Before I chose adversarial image lifespan as my final project, I bounced back and forth between related projects. As this is my first applied and independent machine learning project, I was unprepared for the sheer complexity of even the simplest tasks. Months ago, when we submitted our project proposals, I wanted to explore the explainability of skin pathologies according to neural network classifiers. The

HAM10000 dataset provided images of melanoma, keratosis, vascular lesions and more. My goal was to use GRADCAM to identify the features the network was focusing on but also use targeted adversarial image generation techniques to see how the model was altering images to match a certain pathology–and then compare these two explainability methods. However, using the robustness package proved plenty difficult with an external dataset like HAM10000 that isn't supported natively in the package (like CIFAR-10 or ImageNet is). Despite my adapting the dataset to the necessary ImageFolder package, I still ran into unexpected dimensionality errors and dataset-fitting problems.

Given these errors, I transitioned my project to classify gender based on retinal fundus images. This proved fruitless because with a limited dataset of 5000 images, even a Resnet-50 could not make predictions. 11 epochs of training hovered around a 50% test and training set accuracy each time. These two projects required significant time and energy for tasks as simple as data cleaning and preparing before time intensive training. For a brief moment after these two projects did not work, I sought to predict sexuality based on facial morphology, as Kosinski and Wang found in 2018 ([Kos18]). Finding a dataset itself is a tall task, and then controlling for make-up and cosmetic touch-ups to a face is another large hurdle.

The good thing about exploring these three ideas was that I developed a newfound appreciation for even the simplest of machine learning tasks. Conversations with my mentor Morgan Talbot led me to the topic of adversarial image lifespan. I am grateful for his time and wisdom during this project.

References

[BL21] Tao Bai and Jinqi Luo. Recent Advances in Adversarial Training for Adversarial Robustness. *International Joint Conference on Artificial Intelligence*, April 2021.

[EIS+19] Logan Engstrom, Andrew Ilyas, Hadi Salman, Shibani Santurkar, and Dimitris Tsipras. Robustness (python library), 2019.

[Kos18] Michal Kosinski. Deep Neural Networks Are More Accurate Than Humans at Detecting Sexual Orientation From Facial Images. *Journal of Personality and Social Psychology*, 114(2), February 2018.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Neural Information Processing Systems*, 25, January 2012.

[MS18] Aleksander Mandry and Ludwig Schmidt. A brief introduction to adversarial examples. July 2018.