Exploring Image Synthesis Using A Robust Classifier

Francesco Mastrocinque
Departments of Chemistry & ECE
Duke University
Durham, NC 27708
fam21@duke.edu

Owen Gibson
Department of ECE
Duke University
Durham, NC 27708
ogg4@duke.edu

Abstract

This work explores the impact of seed image distribution and classifier robustness on challenging tasks in image synthesis. Our approach utilizes a non-robust, ResNet50 model as a benchmark classifier and explores an adversarially robust ResNet50 model for image generation. Our findings demonstrate that the adversarially robust classifier greatly outperforms the benchmark classifier for image generation. The results described herein also explore other use cases of the robust classifier involving image inpainting, image super-resolution, and image-to-image translation.

1 Introduction

Advances within deep learning have allowed for extensions beyond classical image classification into new generative tasks like image synthesis.[1] Previous work has found that generative tasks can be performed by training a single adversarially robust model.[2] In general, generative assemblies have a wide variety of use-cases such as applications in digital art, facial editing, and stle transfer.[3, 4, 5] For our project we focused on exploring image generation, image inpainting, image super-resolution, and image-to-image translation. In this paper we will discuss related works, our methods that we used to develop these tasks, challenges we encountered during implementation, and finally our results from the several image synthesis tasks.

2 Related Work

Aleksander Madry's lab at MIT explored the image synthesis problem in their paper "Image Synthesis with a Single (Robust) Classifier." [1] In this paper, Santurkar et al. explored only using a basic classification framework as an approach to image synthesis rather than other state-of-the-art approaches that leverage Generative Adversarial Networks to accomplish this task. An important distinction that Santurkar et al. discovered was that training the classifier to be adversarially robust greatly improved performance for image synthesis tasks. After training the adversarially robust model, a projected gradient descent (PGD) attack is used to manipulate the image towards a target class that causes salient characteristics of the target class to be perceptible. Santurkar et al. showed that slight variations to the PGD attack allows for different image synthesis tasks. This paper was important for helping us understand the topic, providing an already trained model, and for providing suggested approaches that we were able to leverage for our implementation.

3 Methods

3.1 Dataset

The Imagenette dataset used in this study is a subset of the famous ImageNet dataset and was obtained from the following website: https://github.com/fastai/imagenette. The Imagenette dataset

consists of 10 classes (tench, English springer, cassette player, chain saw, church, French horn, garbage truck, gas pump, golf ball, and parachute) and contains a total of 13,394 images that are divided into training and validation sets using a 70/30 split. In this work, the 320 pixel resolution Imagenette dataset was used and images were resized to 224 x 224 pixels.

3.2 Seed Distributions for Image Generation

Generating diverse samples for a given class requires the use of a random seed to begin the class score maximization task. Formally, this involves sampling a random seed from a class-conditional seed distribution, and minimizing the loss of the target label using projected gradient descent (further details in Section 3.4). In order to probe the dependence of the random seed on image generation performance, three distributions were explored and sampled in order to produce the starting random seed. The first distribution that was explored was a random uniform distribution on the interval [0, 1) and was used as the benchmark distribution (Figure 1a). Sampling from such a distribution bears no dependence on the input image classes and their statistical distributions, and thus are seed images that can be completely cast as random noise.

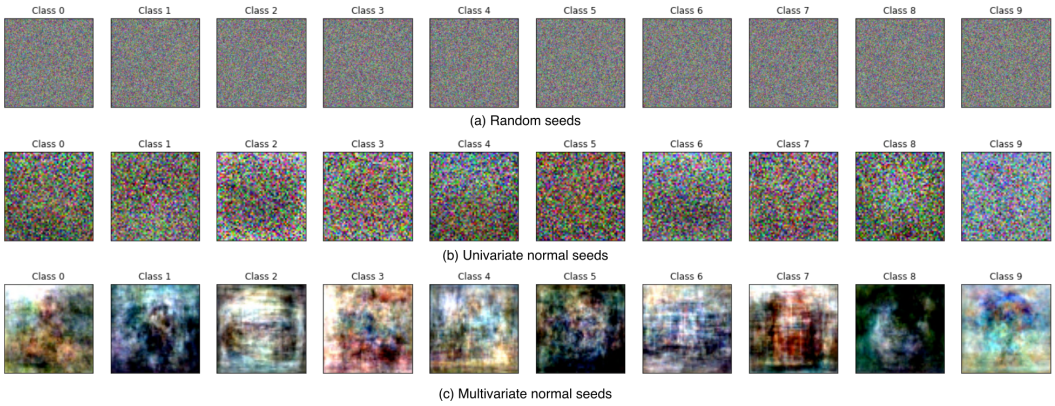

Figure 1: Representative random seed images generated for each class by sampling from a (a) uniform random distribution on the interval [0, 1), (b) univariate and (c) multivariate normal fit for each class.

On the contrary, both univariate and multivariate normal distribution fits to the empirical class-conditional distributions offer sample seed images that bear some statistical similarity to the data distribution as a function of the given input class (Figure 1b,c). For the univariate and multivariate normal fits, images were grouped together by class into sublists and were subsequently downsampled to 56 x 56 pixel images to reduce computational load. The univariate and multivariate normal distributions were calculated on the downsampled images for each class using PyTorch's built-in distribution functions. Random seed images for each class were then sampled from their respective class distribution fit. Before passing the images into the adversarial model, the random seed images generated for each class were upsampled to match the required 224 x 224 pixel ResNet50 input shape.

3.3 Models

The model architectures used as classifiers and explored in this work are based off a ResNet model architecture.[6] For the non-robust classifier, a pre-trained ResNet50 architecture was used with the default weights provided by PyTorch via the 'IMAGENET1K_V2' weights parameter. The robust classifier was provided by the Madry Lab as a ResNet50 classifier that was trained using a robust optimization objective on the full ImageNet dataset, where instead of minimizing the standard expected loss $\mathcal{L}$, minimizing the worst cass loss $\mathcal{L}$ over a specific perturbation set Δ was used:

$$\mathbb{E}_{(x,y)\sim D}\left[\max_{\delta \epsilon \Delta} \mathcal{L}(x+\delta,\, y)\right]. \tag{1}$$

This specific set Δ is able to capture imperceptable changes, such as small ℓ_2 perturbations, preventing the robust classifier from relying on imperceptable features of the input for classification tasks.

3.4 Adversarial Attacks

An iterative PGD adversarial attack was performed on the seed input images, G_y, for each task explored in this work, which utilized the robust ResNet50 model as a determinant classifier. At each iteration, the PGD attack manipulated the input image depending on the calculated loss between the robust model's classification of the generated input image and the true, desired target label. For image generation and image-to-image translation tasks, this consisted of taking the current iteration attacked image, x', and computing the cross entropy loss, $\mathcal{L}$, of the label, y, for the respective class using the equation:

$$x = \underset{\|x'-x_0\|_2 \leq \varepsilon}{\arg\min} \mathcal{L}(x', y), \qquad x_0 \sim G_y \tag{2}$$

Both image generation and image-to-image translation use this equation while keeping the difference between each iteration within the bounds $\| x' - x_0 \|_2 \leq \varepsilon$. The super-resolution task uses a similar equation, but instead limits large deviations using the equation:

$$x = \underset{\|x'-\uparrow(x_L)\|_2 < \varepsilon}{\arg\min} \mathcal{L}(x', y) \tag{3}$$

Where x_L is an unaltered down-sampled image used as the seed image (Figure 5a), and the $\uparrow$ denotes the up-sampling operation based on nearest neighbors. Image inpainting expands on the initial gradient equation to instead focus on the corrupted region of the seed image:

$$x = \arg\min_{x'} \mathcal{L}(x', y) + \lambda \, \| (x - x') \odot (1 - m) \|_2 \tag{4}$$

The seed image for inpainting is directly pulled from the dataset with a 60 x 60 pixel patch added to corrupt a random area of the image (Figure 4). In all tasks, the computed gradient was normalized and subtracted from the seed image with a step size of alpha. The updated image tensor was then renormalized to ensure the changes were not greater than epsilon and clamped to keep the tensor values between 0 and 1. The PGD attack hyperparameters used for each image synthesis task are outlined in Table 1.

Table 1: Hyperparameters used for adversarial attacks.

Task	ε	Iterations	α
Image Generation	40	60-100	1
Image Inpainting	21	720	0.1
Image Super-Resolution	8	40	1
Image-to-Image Translation	60	200	0.5

4 Results

4.1 Image Generation

4.1.1 Using Various Seed Distributions and a Robust Classifier

Full image generation outputs using random, univariate normal, and multivariate normal seed distributions and a robust model classifier are available in the Supplementary Information (Figures 1SI-5SI). Select image generation examples for each of the seed distributions used are entered in Figure 2. The crossentropy loss was monitored as a function of PGD iteration, and of note is the lack of loss convergence prior to 60 iterations for most classes when both a random uniform and univariate normal distribution seed image was used (Figure 6SI). Therefore, PGD attacks that utilized these two seed distributions were run with 100 iterations to reach convergence. It is interesting to note the perceivable semantic differences that were synthesized between the unconverged and converged attacks (Figures 1SI-4SI). Often times, the converged attacks presented a higher level of perceptable detail and decreased number of artifacts in the synthesized images. In addition, it is interesting to observe that starting with a uniform random seed often produced images with well-defined class-specific features and qualitatively outperformed attacks starting with a univariate normal distribution seed, especially with regards to background synthesis (Figure 2a-b). For example, the gas pump image synthesis in Figure 2 contains some synthesized grass in the background when

starting with a random seed, but no background is present in the image synthesized when starting with a univariate normal distribution seed. On the contrary, the PGD attacks that utilized a multivariate normal distribution seed image often converged well before 60 iterations (Figure 6SI) and tended to produce more realistic images with noticeably richer features than attacks that utilized the other two seed distributions, although sometimes with aliasing artifacts (Figure 2c).

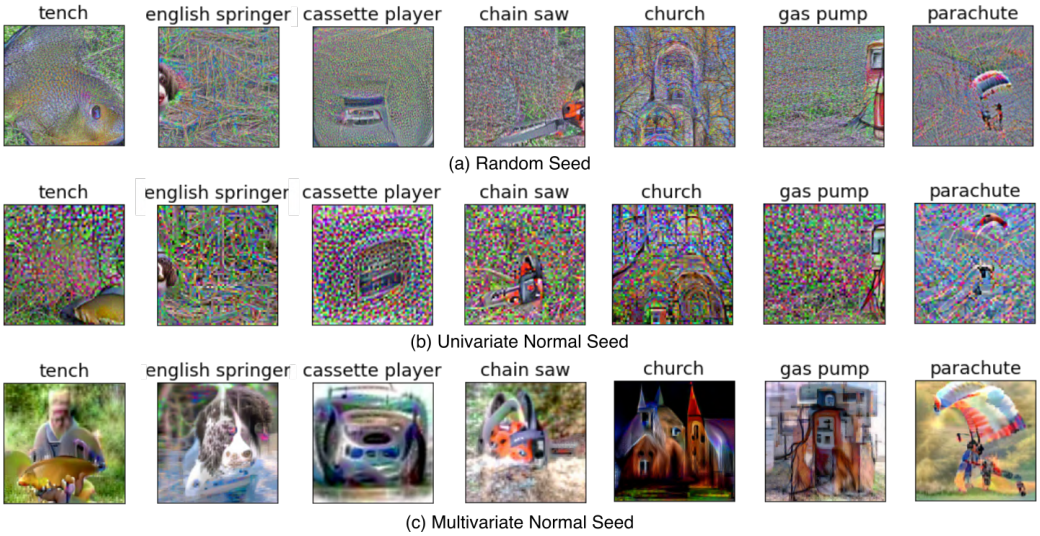

Figure 2: Image generation outputs using a robust classifier starting from (a) uniform random (100 iterations), (b) univariate (100 iterations), and (c) multivariate (60 iterations) normal seed distributions.

4.1.2 Using a Multivariate Normal Seed Distribution and a Non-Robust Classifier

Image generation using a non-robust classifier produced synthesized images that largely resembled the input multivariate normal distribution seeds (Figure 3, Figure 7SI). These images look almost identical to the input seeds and did not produce any outputs that contained features specific to the desired output class that can be perceived by a human. The loss as a function of PGD attack iterations was monitored to ensure convergence (Figure 8SI). This confirms that a robust optimization of a classifier will prevent the model from relying on imperceptible features of the input for classification and indicates that robust models exhibit more human-aligned gradients. Robust training ensures that changes in the model's predictions directly correspond to salient input changes. As a result, the results from image synthesis using a robust model are more perceivable by a human as compared to using a non-robust model. If the model is not trained to be adversarially robust, this non-robust classifier may predict the correct target class, but the changes may not be perceptible to a human observer. Because the multivariate seed distribution images generally performed poorly with the non-robust classifier, image synthesis using random or univariate seed distribution images were not performed.

Figure 3: Select image generation outputs using a non-robust classifier starting from a multivariate normal seed distribution image (60 iterations).

4.2 Image Inpainting

As discussed in Section 3.4, the seed image for the image inpainting task consisted of an image from the Imagenette dataset, but with a 60 x 60 pixel patch applied to hide features of the image.

Inpainting performance was dependent on both class and location of the patch (Figure 4). Often

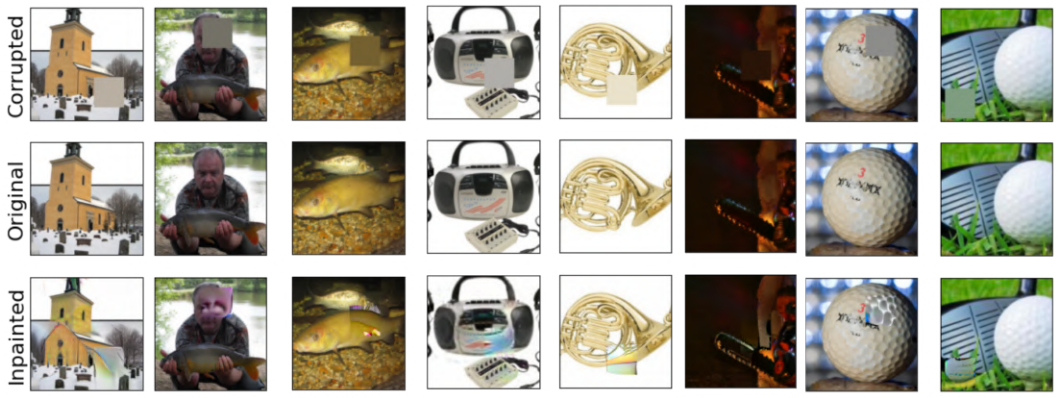

Figure 4: Select inpainting generated images using a 60 x 60 pixel patch.

the worst performing inpainting images occurred when the patch was not covering any of the target class. For example, in Figure 4 for the tench class, one of the patches covers the human's face and is nowhere near the tench (2nd column). As a result the inpainted image incorrectly manipulates the face. On the other hand, in Figure 4 for the tench class, one of the patches mostly covers the tench body (3rd column). The resulting inpainted image looks natural and would take a more detailed look to recognize where the patch was initially placed. Similar results were found across the different classes, with some inpainted images looking natural while others clearly the inpainted patch do not match the surrounding context.

4.3 Image Super-Resolution

The seed image for the image super-resolution task consisted of an image from the ImageNette dataset, initially downsampled to 64 x 64 pixels and upsampled back to 224 x 224 pixels. As a result the seed image lost a lot of details as shown in Figure 5. Super-resolution performance was again

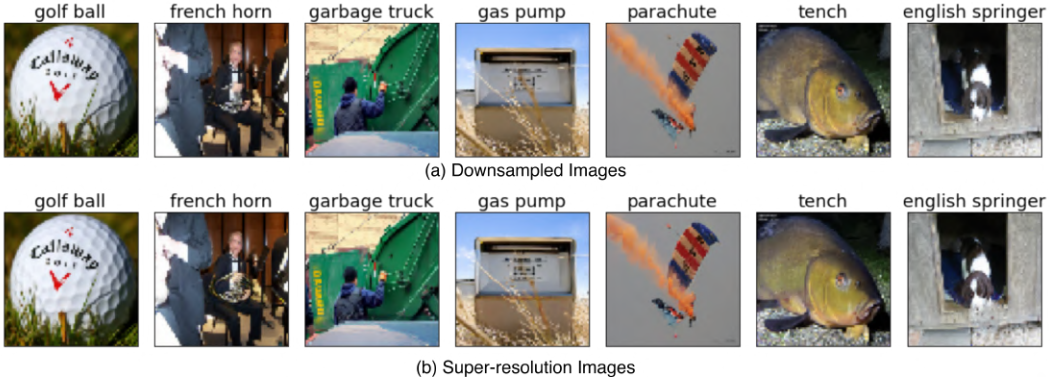

Figure 5: Select images of super-resolution using a robust classifier. Downsampled images were resampled to 224 x 224 pixels.

dependent on both class and the random image selected. For example, in Figure 5 for the golf ball class, the ball dimples were much more perceptible in the super-resolution image. Similarly, in Figure 5 for the french horn class, the super-resolution image accentuated the french horn to make it more perceptible within the surrounding context. In these instances the super-resolution results looked like a natural improvement over the low-resolution images. On the other hand, in Figure 5 for the garbage truck class, it was hard to notice the difference between the low-resolution and super-resolution class. Despite the lack of improvement on some classes the super-resolutions all looked perceivably natural with no large distortions introduced resulting from the super-resolution attacks.

4.4 Image-to-Image Translation Using a Generally Naive Robust Classifier

As has been described previously, robust models provide a well-defined mechanism for transforming inputs into a target class. Here, we demonstrate that robust classifiers provide a powerful mechanism for image-to-image translation, allowing an input image belonging to a given class to be transformed into an output belonging to another target domain in a semantic manner. As opposed to previous literature reports that deliberately train a unique robust classifier only on one unique pair of source and target domains (e.g., Horse $\rightarrow$ Zebra), we demonstrate that image-to-image translation can be achieved using a general (naive) robust classifier trained on an entire dataset. Such a technique allows for a much higher throughput post-training, as any image from a given class can be translated to any target domain and eliminates the need to train a new unique robust classifier for mapping images from each input class to each target class and allows for rapid unpaired image-to-image translation.

Figure 6: Image-to-image translation of several inputs to various target domains using a generally naive robust classifier with a ℓ_2 constrained PGD attack.

As can be seen in Figure 6, the ℓ_2 PGD attack on the naive robust classifier is capable of producing image outputs of a specific target domain using source images that are either semantically similar to the target domain (e.g., Tench $\rightarrow$ Goldfish) or semantically distinct (e.g., Parachute $\rightarrow$ Crane). It is interesting to note that input classes that have similar semantic features to the output class (e.g., Tench $\rightarrow$ Goldfish; Chainsaw $\rightarrow$ Cannon) are translated to capture detailed features (color, texture, patterns, etc) in the target class that are absent when the input class does not have similar semantic features to the output class (Parachute $\rightarrow$ Stingray; Parachute $\rightarrow$ Crane). For example, input images of a tench were transformed to have detailed and semantically distinguishable features of a goldfish when the target class was a goldfish. These details include the golfish's orange color and ripple pattern, producing output images that are remarkably similar to a goldfish. On the contrary, images where the source and target classes have distinct semantic features, like in the parachute images translated to be a stingray, the output image captured the structure of a stingray, but not much of the stingray texture and did not alter the color of the parachute to match the color of a stingray. In general, however, these data are extremely promising and illustrate a strong potential use-case for robust classifiers as a model for intricate and rapid image-to-image translation.

5 Conclusion

Our findings demonstrate that adversarially robust classifiers are able to perform a variety of image synthesis tasks. In particular, robsut classifiers are necessary to generate images with class-specific features that are perceivable to a human. Without robustness, the image synthesis tasks performed poorly as these classifiers encouraged image manipulation that are imperceptible. We also find that starting from a multivariate normal distribution seed outperformed the other seed distributions for image generation.

Acknowledgments and Disclosure of Funding

We would like to acknowledge Dr. Yiran Chen, Jingyang Zhang, and Matthew Inkawhich for their useful suggestions and guidance on this project. We would like to thank the Electrical and Computer Engineering Department at Duke University for their generous computational resources throughout the semester.

References

[1] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 8780–8794. Curran Associates, Inc., 2021.

[2] Shibani Santurkar, Andrew Ilyas, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Image synthesis with a single (robust) classifier. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[3] Yanhong Zeng, Huan Yang, Hongyang Chao, Jianbo Wang, and Jianlong Fu. Improving visual quality of image synthesis by a token-based generator with transformers. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 21125–21137. Curran Associates, Inc., 2021.

[4] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2287–2296, 2020.

[5] Eva Cetinic and James She. Understanding and creating art with ai: Review and outlook. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 18:1 – 22, 2021.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.