Image Synthesis Using a Single Robustness Classifier

Shining Yang, Abhijith Tammangari, Minjung Lee
shining.yang@duke.edu, at396@duke.edu, minjung.lee@duke.edu
Duke University

Abstract

In this project, we performed several image synthesis tasks (image generation, image inpainting, super-resolution, image-to-image translation, and sketch-to-image translation) using a pre-trained adversarially robust classifier released by Santurkar et al (2019). Due to the limitation in computation resources, we used the ImageNette dataset, which is a subset of 10 easily classified classes from Imagenet. In line with the paper, we found that the features learned by a basic classifier are sufficient for all these tasks, provided this classifier is adversarially robust. The idea for these augmentations is to take the original adversarial model and use L2 FGSM attacks in various means to achieve the synthesis tasks.

1 Introduction

The idea of utilizing adversarial attacks of L2 FGSM on an adversarially trained robust classifier to perform image synthesis tasks was initially proposed by Santurkar et al (2019). The main goal behind this paper is to utilize L2 FGSM with various loss functions and minimize the worst-case loss of the set of perturbed images in order to achieve tasks such as image generation, super-resolution, image in-painting, and image-to-image translation.

Based on their findings a simple FGSM attack with a varying number of iterations, loss functions, and alpha values while minimizing the loss for the batch of samples will easily allow us to perform the various synthesis tasks at hand.

On top of proposing this idea, Santurkar and his team also provided a GitHub repository containing links in their paper which contain the suggested L2 FGSM attack attributes as well as model weights for the adversarially trained res-net 50 model. Our goal is to initialize a RES-NET50 model and utilize their adversarially trained RESNET50 model weights to load the pre-trained model. Then we will use our implemented iterations of L2 FGSM attacks with the respective loss functions for each task with the goal of minimizing that total loss. Doing so should enable us to achieve the goal of performing all of our image synthesis tasks.

Furthermore, in our use-case we utilized the Imagenette dataset which is a subset of the Imagenet dataset. The original dataset that Santurkar's team utilized to train their adversarial model was the Imagenet data set, so the Imagenette dataset will still work for our goals. The Imagenette dataset has a total of 10 classes which were relabeled to match the original Imagenet data classes.

2 Image Generation

Realistic image generation has long been a challenge in the computer vision field. According to Santurkar et al (2019), an adversarially robust classifier suffices for image synthesis tasks including realistic image generation. The generation process is based on maximizing the class score of the desired class using the robust model. The goal of this maximization is to add relevant and significantly meaningful features of the desired class to the input image. Here in terms of the input image, we

Submitted to Duke University ECE 661 Fall 2022. Do not distribute.

utilize a random seed at the beginning in order to generate a diverse set of input samples. In line with Santurkar et al (2019), we choose a simple but sufficient multivariate normal distribution to fit the class-conditional distribution. We visualized the conditional seeds for each class in Figure 2.

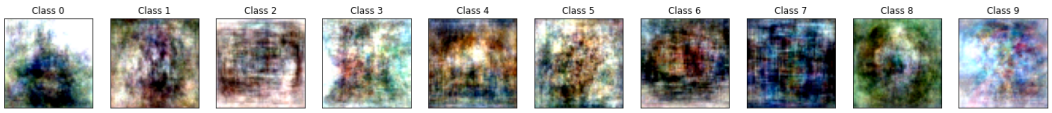

Figure 1: Seeds Visualization

To maximize the class score for our desired classes, we minimize the loss $\mathcal{L}$ of target label y:

$$x = \underset{\|x'-x_0\|_2 \leq \varepsilon}{\arg\min} \mathcal{L}\left(x', y\right), \quad x_0 \sim \mathcal{G}_y \tag{1}$$

where $\mathcal{G}_y$ refers to the class-conditional seed distribution. We adopted the pre-trained model weights and hyperparameters released by Santurkar et al (2019) and utilized Projected Gradient Descend (PGD) with an L2 norm constraint as the strategy to minimize loss. This approach enables us to achieve image generation from conditional-sampled seeds given any desired class. Images generated by our method at a 224×224 resolution are shown in Figure 2 (see appendix for images on all 10 classes). The generated images are diverse, realistic, and show clear visual patterns toward the targeted class.

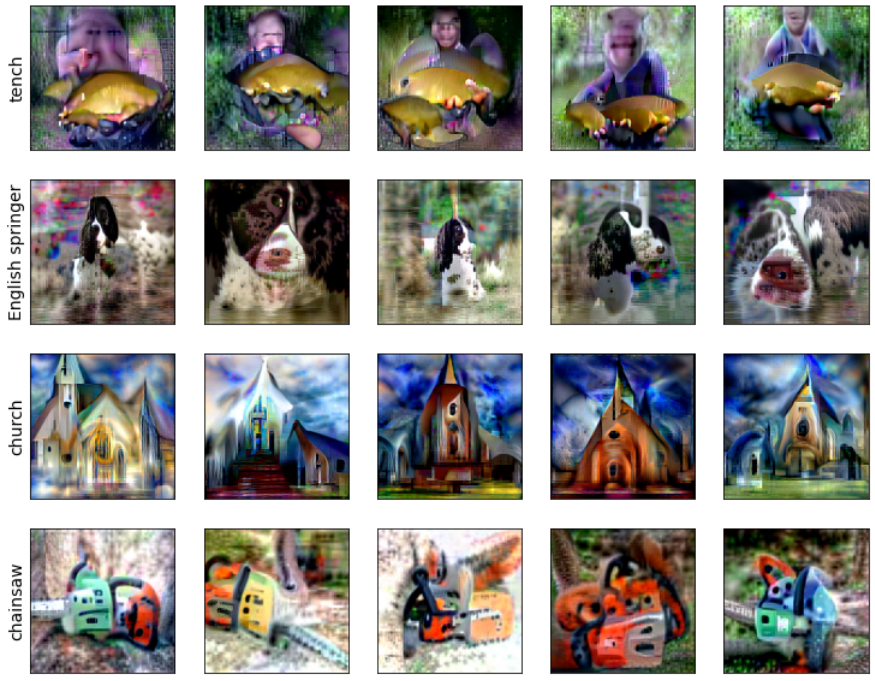

Figure 2: Random samples produced using an adversarially trained robust classifier

For comparison, we also implemented a standard non-robust classifier (in this case, a standard ResNet50) to perform the same tasks. Again, we use the same conditional sampled seeds as input, minimize the loss using PGD with L2 norm constraint, and visualize the results in Figure 3. The generated images using a non-robust classifier look vague and noisy without showing any plausible visual patterns toward the targeted class. The contrast results demonstrate the powerfulness and sufficiency of a single robust classifier in realistic image generation tasks.

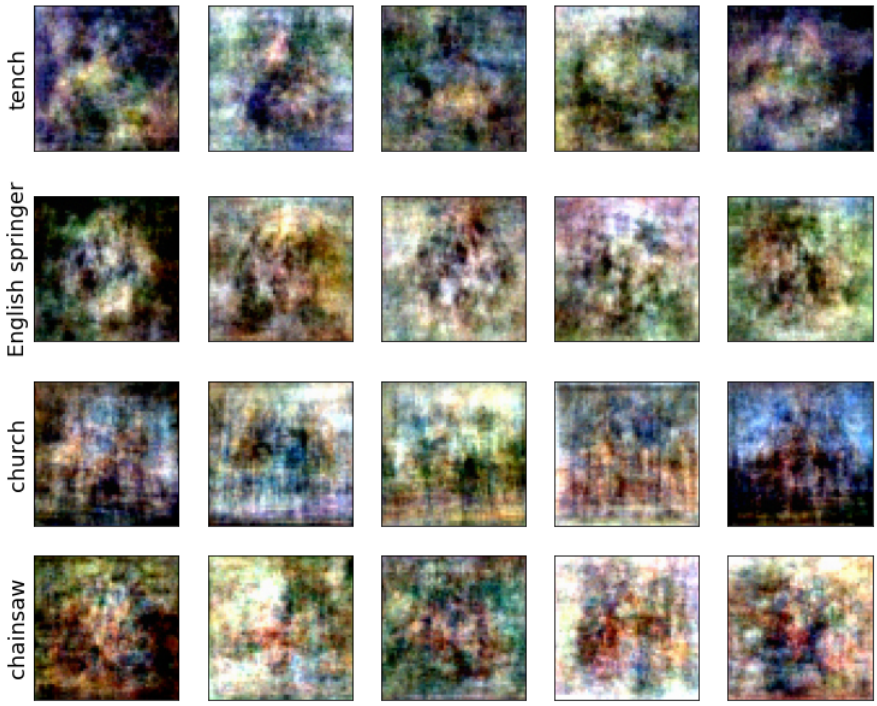

Figure 3: Random samples produced using a standard non-robust classifier

3 Image Inpainting

The goal of image inpainting is to recover the missing pixels using robust models in a manner that is perceptually plausible with respect to the rest of the image. Given a robust classifier pre-trained on uncorrupted data, and a corrupted image x with label y, we solve

$$x_I = \arg\min_{x'} \mathcal{L}(x', y) + \lambda \|(x - x') \odot (1 - m)\|_2 \tag{2}$$

where $\mathcal{L}$ is the cross-entropy loss, $\odot$ denotes element-wise multiplication, m is a binary mask, and λ is a constant (which is 10 here).

We used the L2 FGSM attack on the ImageNette dataset, and for an optimization variable x', we picked random seeds as 1) pure random noise and 2) sample from class-conditional seed distribution, the same as used in Image Generation.

Pure random noise At first, we used pure random noise like Random Noise Attack to get an optimization variable x', and the result is shown in figure 4. Since severe noise was generated, we decided not to use this method.

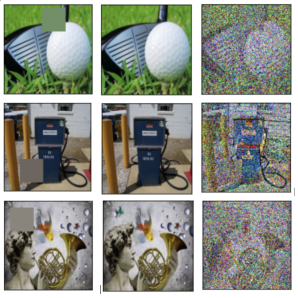

Figure 4: Image inpainting with random samples from pure random noise

Class-conditional seed distribution Instead, we generated random samples from class-conditional seed distribution $\mathcal{G}_y$ which was also used in Image Generation, and the result is shown in figure 5. Additional results can be found in Supplementary Images A.2.

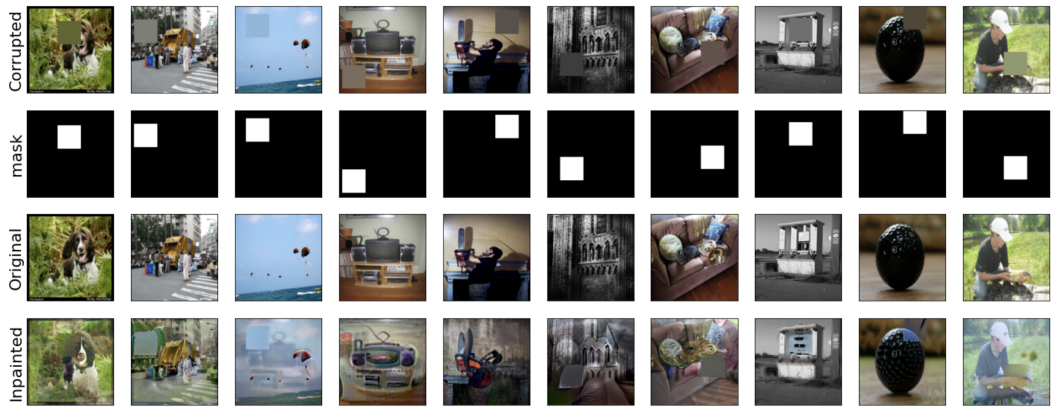

Figure 5: Image inpainting with random samples from class-conditional seed distribution

Although inpainted images do not perfectly match the original images, we can say that it's not too difficult to recognize the images, and most of them look perceptually plausible.

4 Super-Resolution

The goal of Super Resolution is to take a low-resolution image and increase its resolution. This is achieved by utilizing a standard L2 FGSM model where we minimize the loss function on an adversarially trained ImageNet model to increase the resolution of the images. In terms of execution and evaluation, we took some samples of each class from the Imagenette dataset and down-sampled the image to reduce its resolution, then we can up-sample this image to restore it to its original 224 x 244 pixel dimensions but still have its low resolution. These images are then passed into the L2 FGSM attack and the resulting output can be seen in the figure below:

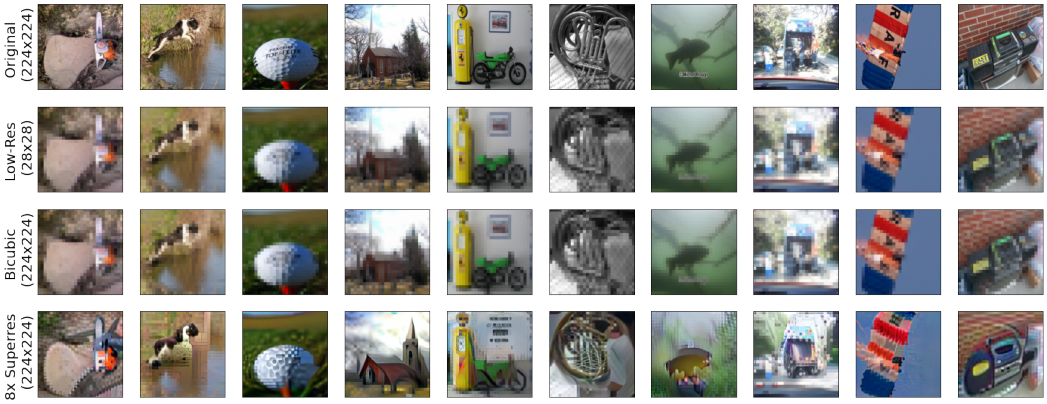

Figure 6: Super-Resolution of images from each class in Imagenette

Looking at the output from the L2 FGSM we can see that this methodology did indeed increase the overall resolution of the object that is classified but there are a few peculiarities to note. The first interesting point is that in the first column we see that the chainsaw's resolution has been increased and becomes clear but this method of attack also increases the resolution of the tree in the background but it is recognized as a saw by the model and it now gains a serrated edge akin to that of a saw-blade. Moving on to the 6th column which contains a french horn, we notice that our method causes the

original black-and-white image to become a colored image where the french horn is now seen as gold. Lastly in the 7th column with a tench fish, we see that the resolution of the fish itself is greatly increased but the background becomes distorted. This suggests that the original model utilized the fish to identify its class but the rest o the image is ignored and thus becomes this blurry and distorted background. The only scenario in this subset that did not have a clear super-resolution output was that of the parachute, this can be understood as this parachute seems to be on fire which is potentially an outlier compared to a standard parachute image.

5 Image-to-Image Translation

For image-to-image translation, our primary goal is to utilize the standard L2 FGSM where the loss function is minimized to convert Imagenette images from one class to another class. To put it simply our method of achieving this goal is to take the standard L2 FGSM model with the input images being the original class and the labels being the class we want to translate the original images into. Doing this results in the following situations where we translate cassettes to french horns, churches into gas stations, and garbage trucks into golf balls:

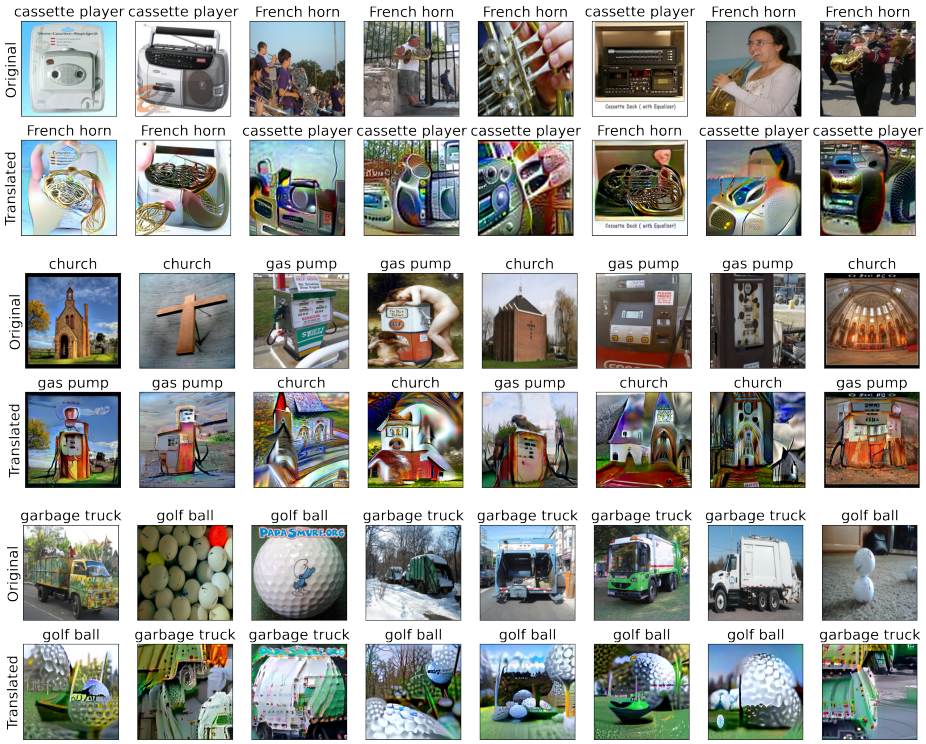

Figure 7: Super Resolution of images from each class in Imagenette

From the examples above we can see that this method takes what the model identifies as the original class and converts those pixels into what it thinks the target class looks like in this scenario. From the examples, in figure 7 we can see that there is an easily identifiable conversion in each of the examples. Although the target class is recognizable in the output images, the background of the image in most cases is either distorted or is replaced with more instances of the target class. This is easily apparent in the garbage truck to golf ball translation as a singular garbage truck is replaced with many instances of golf balls. Overall image to image translation in the Imagenette dataset seems to work but sometimes the outcome is bizarre and wild in some cases.

6 Conclusion

In this work, we leveraged the pre-trained single robustness classifier to perform four different image synthesis tasks based on the paper released by Santurkar et al (2019).

First, in Image Generation, we generated realistic images by adding relevant and significantly meaningful features of the desired class to the input images. Second, in Image Inpainting, we recovered the missing pixels using robust models in a manner that is perceptually plausible for the rest of the image. Third, in Super-Resolution, we took the low-resolution images and increase their resolutions. And the last, in Image-to-Image Translation, we translated images from a source to a target domain in a semantic manner.

We were able to obtain results almost similar to those of the paper, and we confirmed the simple feed-forward classifier, when robustly trained, can be a powerful tool for image synthesis.

References

[1] Santurkar, S., Tsipras, D., Tran, B., Ilyas, A., Engstrom, L., Madry, A. (2019, August 8). Image synthesis with a single (robust) classifier. arXiv.org. Retrieved December 15, 2022, from https://arxiv.org/abs/1906.09453

[2] Santurkar, S., Tsipras, D., Tran, B., Ilyas, A., Engstrom, L., Madry, A. (2019, August 8) from https://github.com/MadryLab/robustness_applications

[3] Imagenette dataset by Jeremy Howard from https://github.com/fastai/imagenette

A Supplementary Images

A.1 Image Generation

We generated images from randomly-sampled seeds for 10 classes: tench, English springer, cassette player, chain saw, church, French horn, garbage truck, gas pump, golf ball, and parachute. The full set of images is shown in Figure 8. In comparison, the full set of images generated using a standard non-robust classifier is shown in Figure 9.

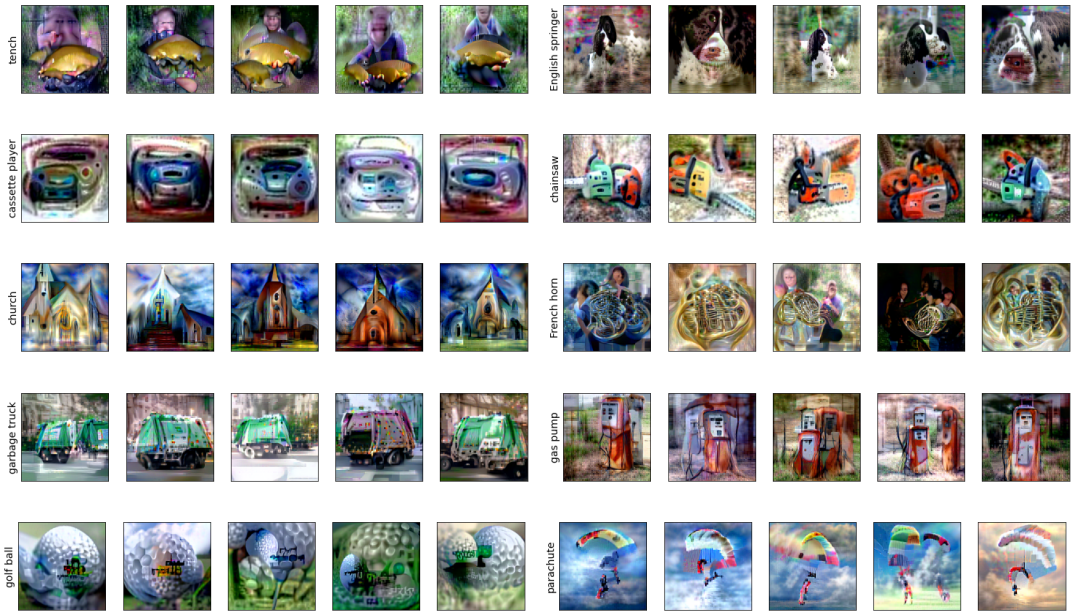

Figure 8: Random samples produced using an adversarially trained robust classifier

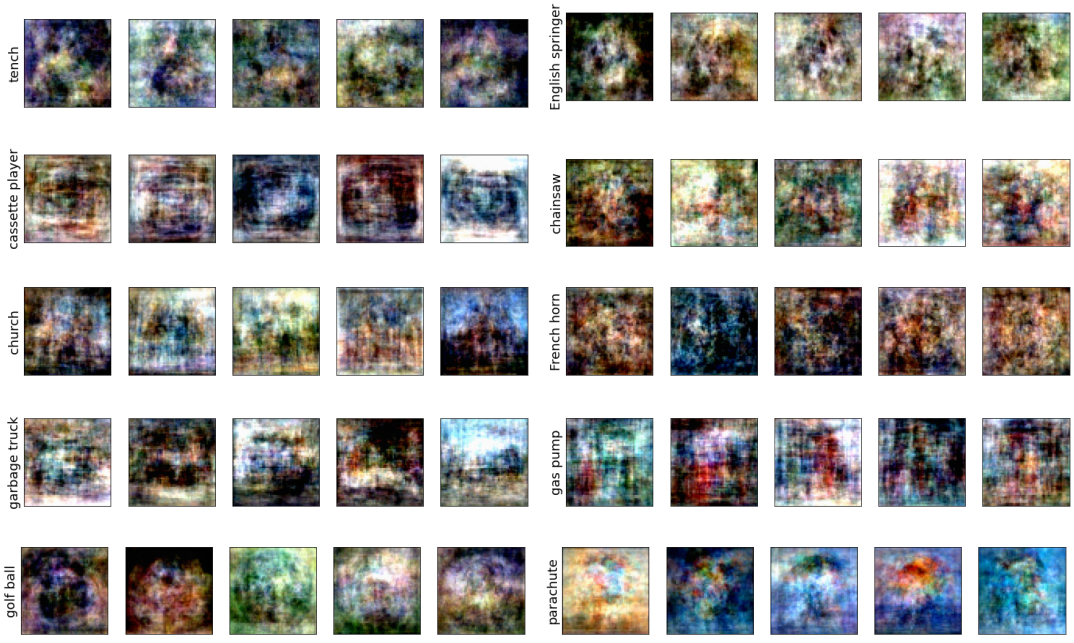

Figure 9: Random samples produced using an adversarially trained robust classifier

A.2 Image Inpainting

Figure 10: Additional image inpaintings with random samples from class-conditional seed distribution

B Individual Contributions

- Shining Yang: Shining is in charge of downloading data, performing the image generation task, making part of the poster, and writing up part of the report.
- Minjung Lee: Minjung is in charge of setting the project environment, performing the image inpainting, making part of the poster, and writing up part of the report.
- Abhijith Tammanagari: Abhijith worked Super resolution, image to image translation, making part of the poster, and writing up part of the report.

C Github Repo

Our codes could be found here: ECE661_final_project_f3