Reproducibility Challenge: Analysis of robust classifiers and handling image synthesis tasks using representations learned from robust models

Qiang Xu (260686619)
Department of Electrical Engineering
McGill University
Montreal, Canada
qiang.xu3@mail.mcgill.ca

Kevin Chen (260658680)
Department of Electrical Engineering
McGill University
Montreal, Canada
shiuan-wen.chen@mail.mcgill.ca

Tian Bai (260773882)
Department of Mathematics and Statistics
McGill University
Montreal Canada
tian.bai2@mail.mcgill.ca

Abstract—**Neural Information Processing Systems (NeurIPS) holds a challenge to ensure that published articles are reliable and reproducible. The goal of this report is to study and reproduce experiment described in "*Image Synthesis with a Single (Robust) Classifier*" published by Shibani Santurkar in 2019, where a basic classification framework was used to tackle challenging tasks in image synthesis such as image generation, inpainting, super-resolution, etc. The CIFAR-10 dataset is chosen for this experiment to compare the results with the original paper on the image generation task. We also discovered a set of parameters which the results might be more plausible [1]–[3].**

Index Terms—**Adversarial Examples, Image Generation, Robust Classifiers**

I. Introduction

The development and prosperity of machine learning algorithm leads to the trend of image synthesis in the technology industry for the past few years. Before the trend of machine learning, researchers used traditional image synthesis methods which is time-consuming and complicated. For example, to merge two pictures, one has to find the features in both images applying feature detection and description methods such as scale-invariant feature transform (SIFT) or Speeded up robust features (SURF), match the image with algorithms like Random sample consensus (RANSAC), and finally blend the image using multiband-blending [4]–[7]. Merging images could be that hard, not to mention image synthesis. Nevertheless, the field of image processing changed revolutionary in 2014. In 2014, Ian Goodfellow published the first generative adversarial network (GAN) algorithm and opened the gate for image synthesis [8]. In 2017, an astonishing research paper "*Progressive Growing of GANs for Improved Quality, Stability, and Variation*" published by Tero Karras rose attention to the image synthesis with GAN [9]. The image synthesized by Karras' team is so real that human can hardly distinguish the real ones and those synthesized by Karras' team. Now, GAN is used for image translation, super-resolution, image translation, image generation, etc. However, researchers are searching for simpler algorithms or methods compared GAN to gain better performance, and Shibani Santurkar proposed a possible solution in "*Image Synthesis with a Single (Robust) Classifier*" [10].

The purpose of this experiment is to reproduce the result from "*Image Synthesis with a Single (Robust) Classifier*" by Shibani Santurkar [10]. We focus on training a robust classifier from scratch and reproduce the image generation task based on the model. Furthermore, we extend the range of datasets that can be used for the image inpainting task.

In "*Image Synthesis with a Single (Robust) Classifier*", Shibani proposed a simpler toolkit than GAN for solving this task. As mentioned in their report, they "*employ a generic classification setup (ResNet-50 with default hyperparameters) without any additional optimizations (e.g., domain-specific priors or regularizers)*" [10].

Shibani's team perform input manipulation to maximize the prediction scores with gradient descent. They implemented *adversarially robust* classifier trained by *robust optimization* objective, and, instead of minimizing expected loss $\mathcal{L}$ with Equation 1,

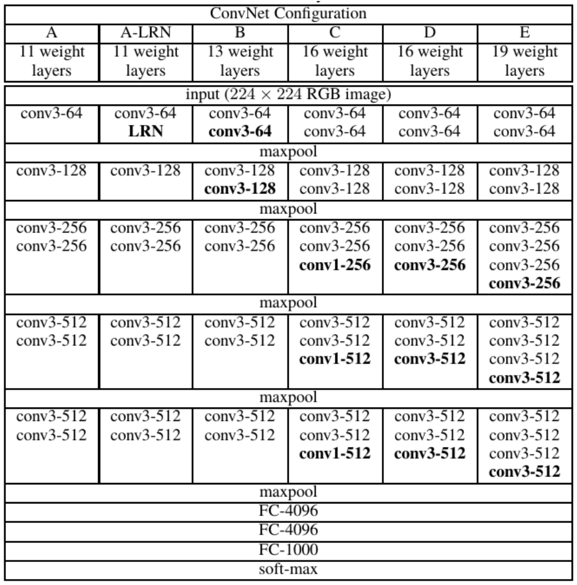
$$\mathbb{E}_{(x,y)\sim\mathbb{D}}[\mathcal{L}(x,y)] \tag{1}$$

the worst case loss over a specific perturbation set Δ is minimized with Equation 3.[10]

$$\mathbb{E}_{(x,y)\sim\mathbb{D}}[\max_{\delta\in\Delta}\mathcal{L}(x+\delta,y)] \tag{2}$$

Shibani's team applied Projected Gradient Descent (PGD) as adversarial attacks, in order to efficiently solve this min-max problem. Details will be provided in the later section.

In order to achieve the similar results, a robust classifier was trained with *VGG-13*[11] architecture as shown in Figure 1, a simple CNN structure from today's view. We successfully trained a classifier which can handle perturbation on CIFAR-10 dataset. The performance of the classifier is surprisingly similar , even better than the one used by the paper.



ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Fig. 1. VGG-13 architecture [12]

II. Dataset

The dataset used is the same as one of the datasets used in ”*Image Synthesis with a Single (Robust) Classifier*” to reduce variance of the experiment. We trained the model with CIFAR-10 as applied in the paper.

CIFAR-10 contains 60,000 32x32 3-channels images in 10 classes and is designed to have 50,000 training images and 10,000 testing images. Each class has 6,000 images to eliminate bias from the dataset as shown in Figure 2 [3].

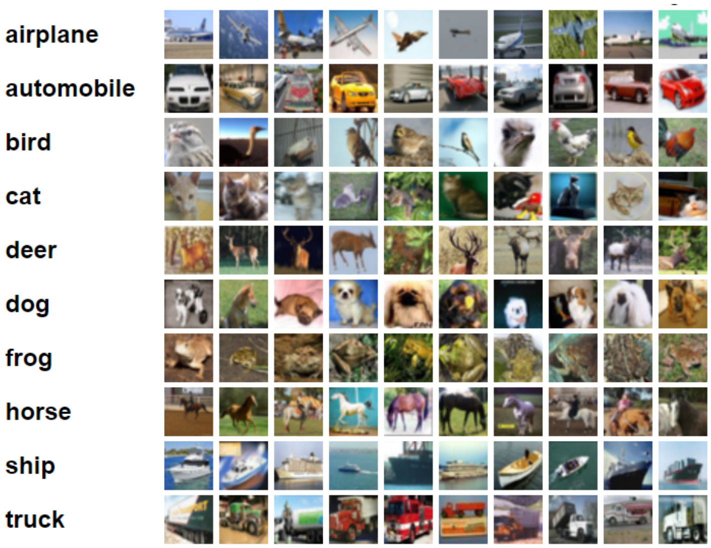


Fig. 2. CIFAR-10 dataset [3]

III. Implementation

A. Robust Classifier

The core of this task is to leverage robust models for image synthesis. Robust models learn representations that are more visible and plausible from the perception of humans.

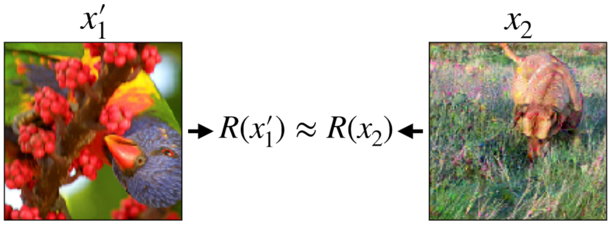


Fig. 3. Similar features [13]

For a standard classifier, despite these two images look completely different to humans, they share very similar representations, while adversarial examples will be possibly misclassified with such standard models.

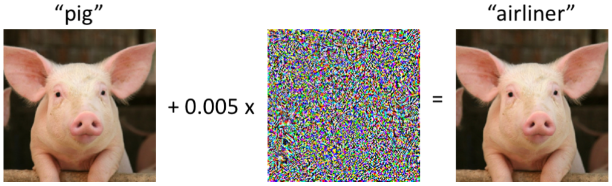


Fig. 4. Perturbation [14]

The image on the left is correctly classified by a state-of-the-art convolutional neural network. After perturbing the image slightly, the classifier regards it as an “airliner” with high confidence. Such phenomenon indicates that the features learnt from a standard classifier are not human-meaningful. Images with perturbation, rotation or translation are not supposed to be misclassified.

1) Intuition behind robust training: To avoid misclassification, one of the simplest strategies is to construct and incorporate such adversarial examples into the training purpose. Since the standard classifiers are susceptible to perturbation, we can train some adversarial examples. However, what kind of perturbation should be applied on images for training purpose becomes one of the question we countered. To solve this question, we went back to the loss function we mentioned at the beginning, which is minimization of such a maximization function.

$$\mathbb{E}_{(x,y)\sim\mathbb{D}}[\max_{\delta\in\Delta}\mathcal{L}(x+\delta,y)] \tag{3}$$

If we want to optimize θ by stochastic gradient descent, this involves computing repeatedly the gradient with respect to θ for the loss function on some batch, which is repetitively update Equation 4.

$$\theta := \theta - \alpha\frac{1}{|B|}\sum_{x,y\in B}\nabla_\theta \max_{\|\delta\|\in\Delta}\mathcal{L}(h_\theta(x+\delta),y). \tag{4}$$

2) Danskin’s Theorem and Adversarial Attacks: The inner part is computed by using Danskin’s theorem [15]. The theorem states that in order to compute the gradient of a function containing a max term, we need to find the maximum, and compute the normal gradient evaluated at that point. In other words, the steps is

$$*argmax\nabla_\theta \max_{\|\delta\|\in\Delta} \ell(h_\theta(x{+}\delta), y) = \nabla_\theta \mathcal{L}(h_\theta(x{+}\delta^\star(x)), y) \tag{5}$$

where

$$\delta^\star(x) = argmax\mathcal{L}(h_\theta(x+\delta), y). \tag{6}$$

However, finding the max is not an easy task. The better we solve the inner part, the closer the Danskin's theorem will hold. Consequently, in order to perform well on solving the maximization problem, it is essential to apply strong adversarial attacks into the inner maximization procedure. Projected gradient descent (PGD) approach,a canonical method for solving constrained optimization problems, is the strongest attack founded so far [15]. PGD is an approach that repeatedly takes a step in the direction of the gradient of the loss function, and then projects the result point back to the constraint set:

$$\Pi_C(x + \eta\nabla L(x, y)) \ . \tag{7}$$

Here, Π_C refers to projecting a point onto the set C. For a given point x, computing $\Pi_C(x')$corresponds to finding the point in C that is closest to x. And, η denotes the step size [15].

B. Illustration and Model for synthesis tasks

For the illustration purpose, we constructed a sample convolutional neural network, and trained both normally with SGD and robustly using projected gradient descent (PGD). The dataset relied on was the standard MNIST database. We used the default hyperparameters and a 0.3 for PGD [16]. The standard trained model was indeed susceptible to perturbation as shown in Table I. This training process did well on classifying original images. However, when perturbation was introduced, the error was almost 100%. The PGD training amazingly overcame the perturbation, original images and images with perturbation were both correctly classified. The result was based on 20 epochs on either training procedure.

Training	Train Error	Test Error	Adv Error
SGD	3.5%	3.4%	99.9%
PGD	7.2%	2.2%	6.9%

TABLE I
ERROR OF TWO MODELS

It took approximately 20 seconds to go through every epoch of PGD training. This efficiency is under the support of NVIDIA Tesla P4. On the other hand, as we attempted to train from scratch on the CIFAR-10 database, each PGD epoch took 9 minutes and converged fairly slowly. This part of the reason that we decided to illustrate our scratching based on MNIST database.

For the sake of time, we leveraged the set-up from package *robustness*[17]. We built up a VGG-13 classifier trained adversarially on the CIFAR-10 dataset, and set the adversarial perturbation budget to be a small value, 1.0. The trade-off is that once a really large value is set for the perturbation budget, it is extremely difficult for the model to learn. The model we constructed with 100 epochs works quite well even when the perturbation budget increases to 1.5 on the test set. It reaches 89.6%, and 95.7% accuracy on the test set with 0.8 budget. This model will be leveraged for the image generation task.

IV. Analysis and Discussion

A. Realistic Image Generation

Image generation or synthesis is the task of producing new images from a given dataset. The first step of generating an image is to fit a distribution (in this case, we kept using multivariate normal distribution as mentioned in the paper) to each class to sample seeds. To generate a sample of class y, we sampled a seed randomly and minimize the loss $\mathcal{L}$ of label y using projected gradient descent (PGD) according to

$$x = \underset{\|x'-x_0\|_2 \leqslant \varepsilon}{\operatorname{argmin}} \ \mathcal{L}(x', y), \quad x_0 \sim G_y, \tag{8}$$

where Gy is the class-conditional seed distribution.

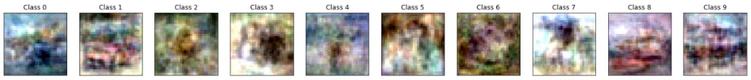


Fig. 5. Seeds from multivariate Gaussian

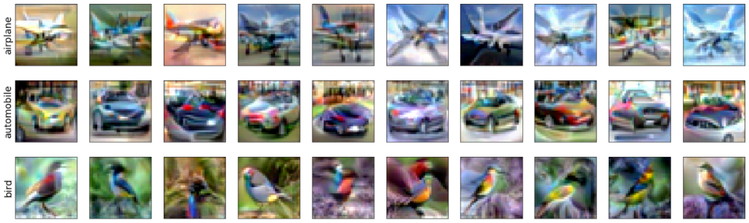


Fig. 6. Conditional image synthesis using Resnet (model from the original paper)

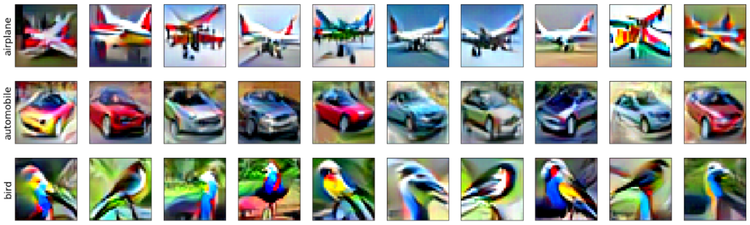


Fig. 7. Conditional image synthesis using our VGG-13 classifier

Dataset	Resnet	VGG-13
CIFAR-10	4.244±0.1	6.682±0.1

TABLE II
INCEPTION SCORES (IS) FOR SAMPLES GENERATED USING ROBUSTLY TRAINED CLASSIFIERS

Inception score is a metric for estimating the visual quality of generated images based on the variety in the set of generated images and the degree to which the images are realistic, i.e. each image looks like something. It is calculated by computing the KL divergence between the response produced by the image and the marginal distribution using an Inception network trained on ImageNet. We generated realistic images using both provided Resnet and our own VGG-13 model. The images generated using Resnet look blurry while the images generated using VGG-13 are sharper and have higher IS.

When tuning the hyperparameters, we increased the number of iterations from its default value of 60 to 100 and this slightly increased the inception scores of both classifiers. Increasing the step size would make the generated images sharper as a large step size leads to numerical instability. The difference in the generated images between Resnet and VGG could be explained through their structures. Since Resnet uses residual blocks to prevent the problem of vanishing gradient on deep neural networks, we expect Resnet to give more stable outputs. Our results show that, using the same hyperparameters, VGG's outputs have more extrema, which makes the images look sharper.

B. Image inpainting

Image inpainting is a process of recovering images with corrupted regions. The goal of inpainting is filling the missing pixels in a manner that is plausible in human's sense. *Given an image x, corrupted in a region corresponding to a binary mask m* $\in \{0,1\}^d$. Shibani's work suggests that a robustly trained feed-forward model suffices to tackle such a reconstruction task.

To go through the process, images are optimized to maximize the score of true class. Given a robust model trained on complete images, and an image X labelled y with missing pixels. Solving the following equation will arise the fixed image.

$$Xi = *\arg\min_{X'} \; \mathcal{L}(X', y) + \lambda||(X - X')\bigodot(1-m)||_2 \tag{9}$$

where $\mathcal{L}$ stands for the entropy loss, $\bigodot$ denotes the element-wise multiplication, and λ is a constant. PGD is used to optimizing the equation as mentioned in Shibani's work.

This paper provides examples from ImageNet dataset. Since an image with more pixels contains more information, an 224 x 224 image is easier to be recovered effectively compared to those with lower resolution, and leads to more visibility.

From the point of reproducibility, we also attempted to recover images with smaller size. In this case, we were handling this inpainting task with CIFAR-10 dataset.

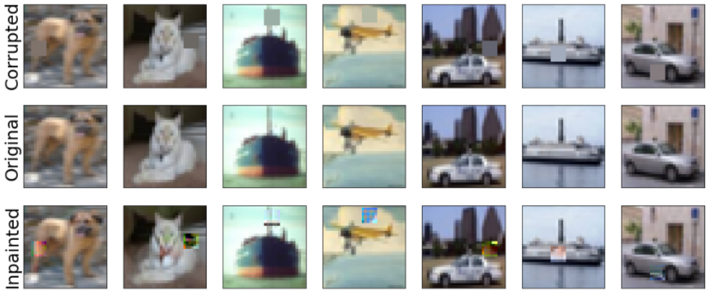


Fig. 8. Randomly chosen images for inpainting task (Clear images see in notebooks)

From the result shown above, the reconstructed images look fairly similar to the original ones. The key for reconstructing smaller images is to apply fewer iterations and smaller perturbation budget while optimization using PGD compared to the setting from ImageNet examples. Large budget will ruin the consistency of images because models trained on small images are susceptible to larger perturbation, which leads to presence of obvious color difference. Decreasing the number of iterations will also overcome this issue due to the lack of information.

Back to the 224*224 images, intuitively we can enable a larger perturbation budget and a larger number of iterations for more details of images. We kept the budget parameters the same as in the original codes, and increased the number of iterations by 300.

The results split into two extreme cases as shown in Figure 9. Reconstruction on the body of objects is to a high-level human-meaningful result. Meanwhile, the procedure is not able to handle recovering the background of images at all. To reason about the split-up, the model we leveraged only classified the objects but not the backgrounds. In another word, the Equation 9 optimization will fail while maximizing the score of true class.

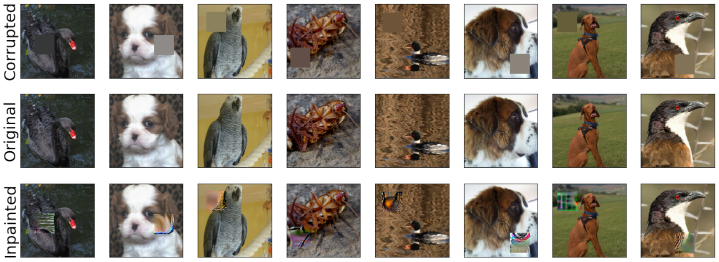


Fig. 9. Randomly chosen images from Restricted ImageNet. Those two birds and the insect images are almost perfectly constructed, but the first and fifth one are not really plausible to human

V. Conclusion and Discussion

In this project, we constructed an adversarially robust classifier which is not susceptible to perturbation on images, and leveraged it to the realistic image generation task mentioned in "*Image Synthesis with a Single (Robust) Classifier*".

Our results show that robust classifiers can be used for synthesizing realistic images. The difference between the provided model and our own VGG model is that Resnet gives outputs that are more stable as Resnet solves the degradation problem of deep networks. As a result, using the same hyperparameters, the images generated by VGG are sharper as there are more pixels that have values close to extrema in the images.

Moreover, we created a baseline for the image inpainting task. Images with smaller size are able to be reconstructed with smaller parameters values during PGD optimization. Slight adjustment on the PGD parameters led to a better result (at least from the perception of human) , for the ImageNet dataset.

However, behind the simplicity of leveraging simple robust classifier for complicated image synthesis tasks, the unignorable limitation remains unsolved to us. This procedure is not able to handle the background. The generated background of images is not negligible to human eyes, while the missing pixels on the background is almost unrecoverable due to the reason that a single robust classifier is hardly learning from the background.

In addition, we were not able to re-verify the image translation task since we had no access to the dataset, and we did not train a robust model based on ImageNet due to the lack of GPU supports. For curiosity and learning purpose, we aimed to improve our model efficiency to reduce the time complexity. As mentioned early in this report, the model from scratch took 9 minutes for each epoch, while the robustness package only took 3 minutes.

Overall, we are able to claim that experiments on this paper are reproducible. Our simpler robust model generated similar results as the ResNet-50 used by the author's team. Moreover, we expanded the availability of the toolkit on the inpainting task. We also reconstructed images which seemed more plausible by slightly adjusting the PGD parameters.

VI. Contribution on this project

Tian Bai: Construct the robust model and study on the inpainting task.

Qiang Xu: Focused on image generation task.

Kevin Chen: Organizing Write Up

References

[1] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015. DOI: 10.1007/s11263-015-0816-y.

[2] A. Krizhevsky, "Learning multiple layers of features from tiny images," *University of Toronto*, May 2012.

[3] *Cifar-10 and cifar-100 datasets*. [Online]. Available: https://www.cs.toronto.edu/~kriz/cifar.html, (accessed: 12.13.2019).

[4] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004, ISSN: 0920-5691. DOI: 10.1023/B:VISI.0000029664.99615.94. [Online]. Available: https://doi.org/10.1023/B:VISI.0000029664.99615.94.

[5] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, Jun. 2008, ISSN: 1077-3142. DOI: 10.1016/j.cviu.2007.09.014. [Online]. Available: http://dx.doi.org/10.1016/j.cviu.2007.09.014.

[6] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981, ISSN: 0001-0782. DOI: 10.1145/358669.358692. [Online]. Available: http://doi.acm.org/10.1145/358669.358692.

[7] H. Yong, J. Huang, W. Xiang, X. Hua, and L. Zhang, "Panoramic background image generation for ptz cameras," *IEEE Transactions on Image Processing*, vol. 28, no. 7, pp. 3162–3176, Jul. 2019, ISSN: 1941-0042. DOI: 10.1109/TIP.2019.2894940.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., Curran Associates, Inc., 2014, pp. 2672–2680. [Online]. Available: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

[9] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=Hk99zCeAb.

[10] S. Santurkar, A. Ilyas, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Image synthesis with a single (robust) classifier," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., Curran Associates, Inc., 2019, pp. 1260–1271. [Online]. Available: http://papers.nips.cc/paper/8409-image-synthesis-with-a-single-robust-classifier.pdf.

[11] A. Z. Karen Simonyan, *Very deep convolutional networks for large-scale visual recognition*. [Online]. Available: https://www.robots.ox.ac.uk/~vgg/research/very_deep/, (accessed: 12.10.2019).

[12] *Vgg-13—kaggle*. [Online]. Available: https://www.kaggle.com/pytorch/vgg13, (accessed: 12.11.2019).

[13] L. Engstrom, A. Ilyas, A. Madry, S. Santurkar, B. Tran, and D. Tsipras, *Robustness beyond security: Representation learning*. [Online]. Available: http://gradientscience.org/robust_reps/, (accessed: 12.11.2019).

[14] L. S. Aleksander Madry, *A brief introduction to adversarial examples*. [Online]. Available: http://gradientscience.org/intro_adversarial/, (accessed: 12.11.2019).

[15] D. T. Aleksander Madry Ludwig Schmidt, *Training robust classifiers*. [Online]. Available: http://gradientscience.org/robust_opt_pt1/, (accessed: 12.08.2019).

[16] C. J. B. Yann LeCun Corinna Cortes, *The mnist database of handwritten digits*. [Online]. Available: http://yann.lecun.com/exdb/mnist/, (accessed: 12.09.2019).

[17] L. Engstrom, A. Ilyas, S. Santurkar, and D. Tsipras, *Robustness (python library)*, 2019. [Online]. Available: https://github.com/MadryLab/robustness.