One Size Does Not Fit All: Transferring Adversarial Attacks Across Sizes

Jeremiah Duncan
The University of Tennessee, Knoxville
jdunca51@vols.utk.edu

Amir Sadovnik
The University of Tennessee, Knoxville
asadovnik@utk.edu

Abstract

As CNNs become more prevalent, it is important to increase their robustness before placing them in mission-critical applications. Recent work has exposed strong vulnerabilities in them. And while numerous attacks have been developed in white and black-box settings, they rely on the fact that the size of the image is known. In this work, we examine the effect of resizing methods and size ranges on adversarial example efficacy. Then, we present a novel way to resize adversarial noise which can target models that accept images of a different size. Finally, we present a multi-scale attack which can create one adversarial image that transfers to multiple different sizes at once. Our methods are agnostic to the attack used and the way resizing is done. We show that our methods work on image downscaling methods commonly used in deep learning libraries using a subset of ImageNet, called CINIC-10.

1. Introduction

Reliance on CNNs is increasing at an exponential rate. Their use has spread from image classification contests [18] to self-driving cars [3, 17], face recognition [13, 21], and x-ray analysis [15]. Despite their increased use in mission-critical roles, it is trivial to fool them with adversarial examples—images with added noise meant to cause misclassification [9, 12, 20].

In typical research settings, it is often assumed that there is minimal preprocessing applied to images before inference. Partly because datasets, like CIFAR-10 and ImageNet, can be downloaded in a standard size and file format. However, in real world settings, images come in all shapes, sizes, and formats, requiring advanced preprocessing. And while some attacks have attempted to persist adversariality through JPEG compression [19] or used resizing to transfer them between black-box models of the same input size [25], none have tackled transferring them between models that accept different input sizes to our knowledge.

In this paper we show how extreme resizing of an image effects adversariality and we propose a novel method for creating adversarial examples that can transfer through resizing transformations (See Fig. 1). Our method is able to create examples which can fool a classifier of any size regardless of the original image size, while keeping the original resolution.

Our paper is organized as such: First, we present an overview of attacks and defenses proposed in recent years, with a focus on ones that rely on image resizing. Second, we present our methods for creating adversarial attacks which can transfer through resizing. Third, we present the effect of different image resizing methods on adversariality and show that adversarial examples break during resizing. Finally, we evaluate the effectiveness of our method when used to extend existing attacks through resizing.

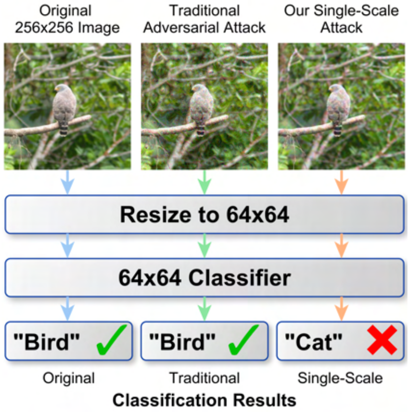


Figure 1: Traditional attacks are unable to transfer through a wide range of image resizing. When classifying images created using a traditional attack (green arrows), they are unable to trick classifiers that accept a different size than the original image. With our single-scale adversarial attack (orange arrows), we can create adversarial examples that transfer across sizes

2. Previous Work

2.1. Adversarial Examples

Adversarial examples are images that have a small added perturbation that causes them to be incorrectly classified by machine learning classifiers, while still being recognizable to humans. They were first discovered by Szegedy *et al*. [20], who implemented the first attack using L-BGFS optimization. Goodfellow *et al*. [9] expanded upon adversarial examples by introducing the Fast Gradient Sign Method (FGSM), a one-step method for creating them. Madry *et al*. [16] introduced Projected Gradient Descent (PGD), a stronger, iterative version of FGSM that bounds the perturbation in an ℓ_p ball around the original image. Out of all white-box attacks formulated in recent years, Carlini and Wagner's [5] attack is arguably the strongest. It has been shown to break many defenses that were previously successful in the past [1, 6, 4]. Although our method is agnostic to the type of attack used, in this work we use FGSM and PGD to generate adversarial examples. They are strong white box attacks that are still fast enough to test on a large dataset.

2.2. Downscaling Attacks

As our work is focused on the effects of resizing adversarial attacks, we mention three additional papers. Xiao [23] brought to light security flaws in many deep learning pipelines and showed that a nearest neighbor filter can easily be beaten. We extend gradient-based attacks so they can target classifiers of any size and keep adversariality through any resizing filter.

Athalye *et al*. [2] use Expectation over Transformation to create adversarial images robust to a presumed transformation distribution. While they consider scale, they limit it to 0.9-1.4× the original size. Xie *et al*. [25] used the resizing and padding methods described in [24] during each step of Iterative FGSM (I-FGSM) to create adversarial examples that transfer better in white and black-box settings. Although their work creates attacks that transfer across sizes, they also only vary the scale of the image slightly (up to 1.1×).

3. Our Method

3.1. Single-Scale Attack

Our single-scale attack allows an image of a certain size (s) to be able to attack a model which accepts images of a different size (t). We first resize the clean image X to match the classifier's input size t.

$$X_t = \text{resize}(X, t) \tag{1}$$

where the subscript refers to the current size of the image. We then use X_t as the starting point for a PGD attack on the target classifier.

$$X_{t_{adv}} = \text{ATK}(X_t, \epsilon) \tag{2}$$

where ATK is the attack algorithm (for example, PGD) used, and ϵ is the strength of the attack.

We upscale the previously downscaled clean and adversarial version of our image back up to the original size s.

$$X^t_{s_{adv}} = \text{resize}(X_{t_{adv}}, s) \tag{3}$$

$$X^t_s = \text{resize}(X_t, s) \tag{4}$$

Although X^t_s is the same size as X, they are different because X^t_s went through resizing twice and the superscript refers to the size of the image before being resized. We do this because taking the difference between them gives us a close approximation of what the adversarial noise would be if it could be perfectly upscaled back to the original size. We take the upscaled adversarial noise and add it to a clean copy of the image we started with, leaving us with an adversarial version of the original image, denoted X_{adv}.

$$X_{adv} = X + (X^t_{s_{adv}} - X^t_s) \tag{5}$$

When the high resolution adversarial example (X_{adv}) is sent to the target classifier, it will be resized to the target size t. Even though it has gone through resizing, this image will be able to "fool" the classifier as its noise targets that specific size while still looking like the original image X.

3.2. Our Multi-Scale Attack

We adapt our method to attack multiple sized classifiers at once for when the target classifier's input size is unknown or if there is an ensemble of classifiers that each accepting different sizes. Instead of attacking a single classifier, we assemble a set of classifiers keyed on different sizes and perform our single-scale attack on each individual classifier. This relies on the fact that although adversarial perturbations do not transfer through a wide range of scales, they transfer well over small scale changes. By selecting a representative set of classifiers to attack, we can create true "multi-scale" attacks.

This equates to changing Eq. 5 to

$$X_{adv} = X + \sum_{t \in T}(X^t_{s_{adv}} - X^t_s) \tag{6}$$

where T is the set of representative sizes we use to generate the attack on. In order to ensure that the resulting noise is not too large we change Eq. 2 to

$$X_{t_{adv}} = \text{ATK}(X_t, \frac{\epsilon}{\eta}) \tag{7}$$

where η is a suppressing factor on the noise. This defaults to our single-scale attack when T contains only one size and $\eta = 1$, while allowing flexibility in attacking multiple sizes at once. We introduce η because if we use the original ϵ when attacking multiple sizes at once, the resulting ℓ_∞ distance can be large. We find that setting $\eta = |T|$ provides a good balance between image quality and adversarial success using our multi-scale attack.

3.3. Model Architecture and Parameters

We use ResNet-20 [10] for our models, only changing the input size. Before training we use bilinear filtering to resize images, normalize them, and subtract the training mean. We also use data augmentation methods such as flipping, shifting, and rotation. We train each model for 200 epochs, with a learning rate of 0.1 that is reduced by a factor of 10 at 80, 120, and 160 epochs. We use a mini-batch of 32. Each model took less than a day to train on a single V100 GPU. Our attacks use ℓ_∞ distance to limit perturbations.

4. Experiments and Results

4.1. CINIC-10

In recent literature, MNIST [14], CIFAR-10 [11], Tiny ImageNet [22], and ImageNet [8] are four common datasets used to attack CNNs. For our purposes, MNIST, CIFAR-10, and Tiny ImageNet's images are too small to get meaningful results after further downscaling. And while ImageNet contains images that meet our downscaling requirements, using it to train models at multiple image sizes and with different resizing filters is computationally expensive. Instead we use a subset of ImageNet called CINIC-10 [7]. It is a mix between ImageNet and CIFAR-10 that collates many of ImageNet's synsets into CIFAR-10's classes. It includes 80k training images and 40k validation images.

4.2. Resizing Kills Adversariality

Although adversarial images are transferable between different models as long as they accept images of roughly the same size [9, 20], we wanted to test if this holds true when two model classify images of varying sizes. We conducted experiments with multiple resizing filters and sizes. In Fig. 3 we include a portion of these results that shows they do not transfer well through bilinear filter resizing and that the further away in size the target model is from the one used to generate the adversarial examples, the less effective they are. This can be seen in the curves representing the different ϵ values. As the attack was created on a 256×256 classifier, the accuracy on small images is almost equal to the clean accuracy. We obtain similar results for other resizing methods that we show in Appendix B, with one exception being the nearest neighbor filter which is vulnerable in other ways [23].

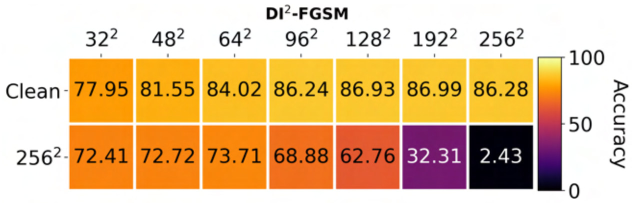


Figure 2: 256×256 classifier accuracy on 10,000 clean and adversarial images created using an adapted DI^2-FGSM with a transformation probability of 100%

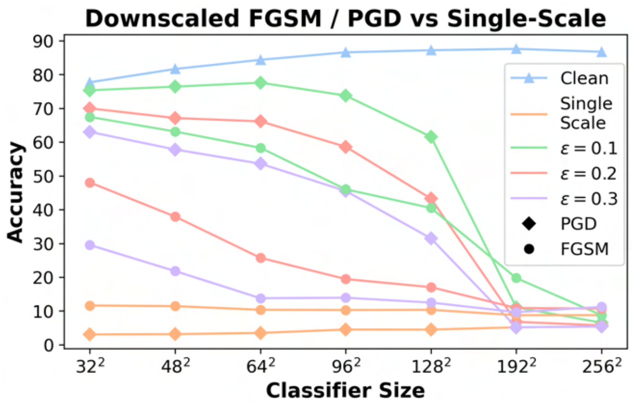


Figure 3: Accuracy on adversarial examples that target each sized classifier using our single-scale attack using FGSM ($\epsilon = 0.1$) and PGD ($\epsilon = 0.3$) versus examples created on a 256×256 classifier and downscaled

4.3. Baseline Comparison

Although there is no previous work that directly attacks models with differently sized images, we believe DI^2-FGSM [25] is the closest to our own. To fairly compare our attacks, we adapt it to our wide range of sizes. At every step of I-FGSM, they perform random resizing with a probability p, over a small range of sizes, $[299, 330)$. We perform the same algorithm, picking one of our seven sizes randomly ($[32, 48, 64, 96, 128, 192, 256]$) and then padding all sides before taking an I-FGSM step. In Fig. 2, we show that when using the attack settings suggested by [25], the adapted DI^2-FGSM does not perform better than simple downscaling (Fig. 3).

4.4. Single-Scale Attack

We use all 40k 256^2 validation images from CINIC-10 to create examples using our single scale attack whose target is one of the following sizes: $[32, 48, 64, 96, 128, 192, 256]$. This gives us 280k 256^2 single-scale adversarial images that we resize and send to the same classifier used to create them to see if their adversariality transferred through resizing. And we focus on downscaling because accuracy on clean images decreases when upscaled, shown in Appendix A.

In Fig. 3, we present the classifiers' accuracies when using FGSM and PGD. The results clearly show that examples generated using our attack mostly keep their adversariality after being downscaled while the original attacks do not. Although our method performs better than both attacks, it is clear that at higher ϵ values, FGSM attacks transfer between sizes better than PGD. However, the resulting FGSM images are often of very poor quality compared to ours.

Fig. 4 shows that although these attacks are able to fool classifiers that accept images of the same size as the one the attack was created on (the diagonals), they are not able to consistently fool classifiers that were trained on any other size (the off diagonals).

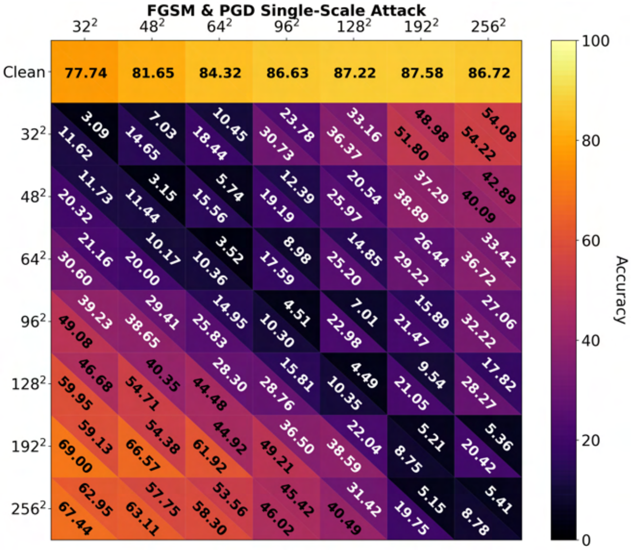


Figure 4: Accuracy on clean and adversarial images created using our single-scale attack ($\epsilon_{\text{FGSM}} = 0.1$, $\epsilon_{\text{PGD}} = 0.3$). The bottom half of each square is FGSM accuracy and the top is the PGD accuracy. Rows represent the size of the classifier the images were created on. Columns represent the size of the classifier the images were tested on. As the color gets darker, the better our attack worked at tricking the classifier

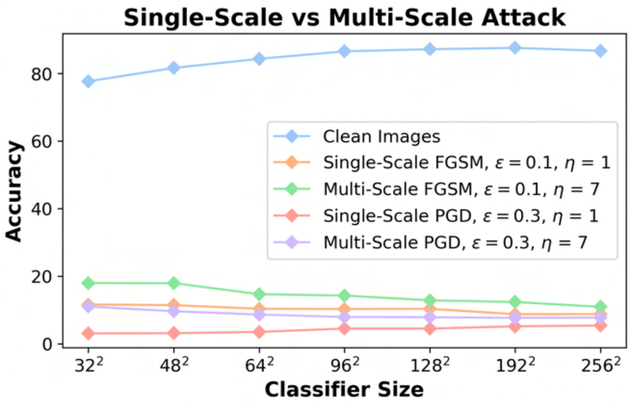


Figure 5: Accuracy on clean and adversarial images created with our single and multi-scale attack using FGSM and PGD. While the single-scale attacks are different for each size, our multi-scale attack is a single image able to attack all seven classifiers individually

4.5. Multi-Scale Attack

We created a representative set of classifiers where each model accepts images of a different size (e.g. one of $[32, 48, 64, 96, 128, 192, 256]$). Using our multi-scale attack, we generated 40k examples that we send to each classifier in the set, with FGSM and PGD results in Fig. 5. As opposed to the single-scale attack, each of these can attack all seven classifiers at once. This figure shows two things. The first is that our multi-scale attack, is able to sufficiently attack each classifier of a different size simultaneously, bringing all of their individual accuracies at or below 18% in the case of FGSM and 11% using PGD. The second is that each classifier's accuracy on the multi-scale images is not much higher than it is on the single-scale ones—indicating that we are able to keep each classifier's specific perturbations, even when adding all of their individual one's.

5. Discussion and Conclusion

In this paper we have shown a way to transfer adversarial attacks through resizing methods. While traditional FGSM and PGD attacks are unable to transfer well across sizes, our attacks can create adversarial images that transfer to a specific size or range of sizes.

There are many directions in which we can extend this work in the future. For example, looking for a better way to automatically determine the size or range of sizes to attack. In addition, we plan on examining how well our attacks transfer to sizes outside of the seven sizes we used, how other types of defenses effect our results, and how we can adapt our method to persist in these cases.

References

[1] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, July 2018. 2

[2] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 284–293, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. 2

[3] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to end learning for self-driving cars, 2016. 1

[4] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, AISec '17, page 3–14, New York, NY, USA, 2017. Association for Computing Machinery. 2

[5] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2017. 2

[6] Nicholas Carlini and David A. Wagner. Defensive distillation is not robust to adversarial examples. *CoRR*, abs/1607.04311, 2016. 2

[7] Luke N. Darlow, Elliot J. Crowley, Antreas Antoniou, and Amos J. Storkey. Cinic-10 is not imagenet or cifar-10, 2018. 3

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. 3

[9] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. 1, 2, 3

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016. 3

[11] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. 3

[12] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017. 1

[13] S. Lawrence, C. L. Giles, Ah Chung Tsoi, and A. D. Back. Face recognition: a convolutional neural-network approach. *IEEE Transactions on Neural Networks*, 8(1):98–113, Jan 1997. 1

[14] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 3

[15] C. Liu, Y. Cao, M. Alcantara, B. Liu, M. Brunette, J. Peinado, and W. Curioso. Tx-cnn: Detecting tuberculosis in chest x-ray images using convolutional neural network. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 2314–2318, Sep. 2017. 1

[16] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. 2

[17] B. T. Nugraha, S. Su, and Fahmizal. Towards self-driving car using convolutional neural network and road lane detector. In *2017 2nd International Conference on Automation, Cognitive Science, Optics, Micro Electro-Mechanical System, and Information Technology (ICACOMIT)*, pages 65–69, Oct 2017. 1

[18] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander Berg, and Fei Fei Li. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115, 09 2014. 1

[19] Richard Shin and Dawn Song. Jpeg-resistant adversarial images. 2017. 1

[20] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. 1, 2, 3

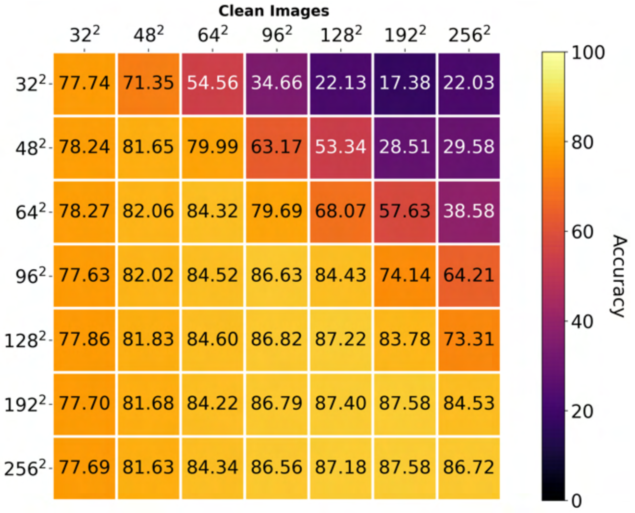
[21] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 1

[22] Jiayu Wu, Qixiang Zhang, and Guoxi Xu. Tiny imagenet challenge, 2017. 3

[23] Qixue Xiao, Kang Li, Deyue Zhang, and Yier Jin. Wolf in sheep's clothing - the downscaling attack against deep learning applications. *CoRR*, abs/1712.07805, 2017. 2, 3

[24] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. In *International Conference on Learning Representations*, 2018. 2

[25] C. Xie, Z. Zhang, Y. Zhou, S. Bai, J. Wang, Z. Ren, and A. L. Yuille. Improving transferability of adversarial examples with input diversity. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2725–2734, 2019. 1, 2, 3



	32^2	48^2	64^2	96^2	128^2	192^2	256^2
32^2	77.74	71.35	54.56	34.66	22.13	17.38	22.03
48^2	78.24	81.65	79.99	63.17	53.34	28.51	29.58
64^2	78.27	82.06	84.32	79.69	68.07	57.63	38.58
96^2	77.63	82.02	84.52	86.63	84.43	74.14	64.21
128^2	77.86	81.83	84.60	86.82	87.22	83.78	73.31
192^2	77.70	81.68	84.22	86.79	87.40	87.58	84.53
256^2	77.69	81.63	84.34	86.56	87.18	87.58	86.72

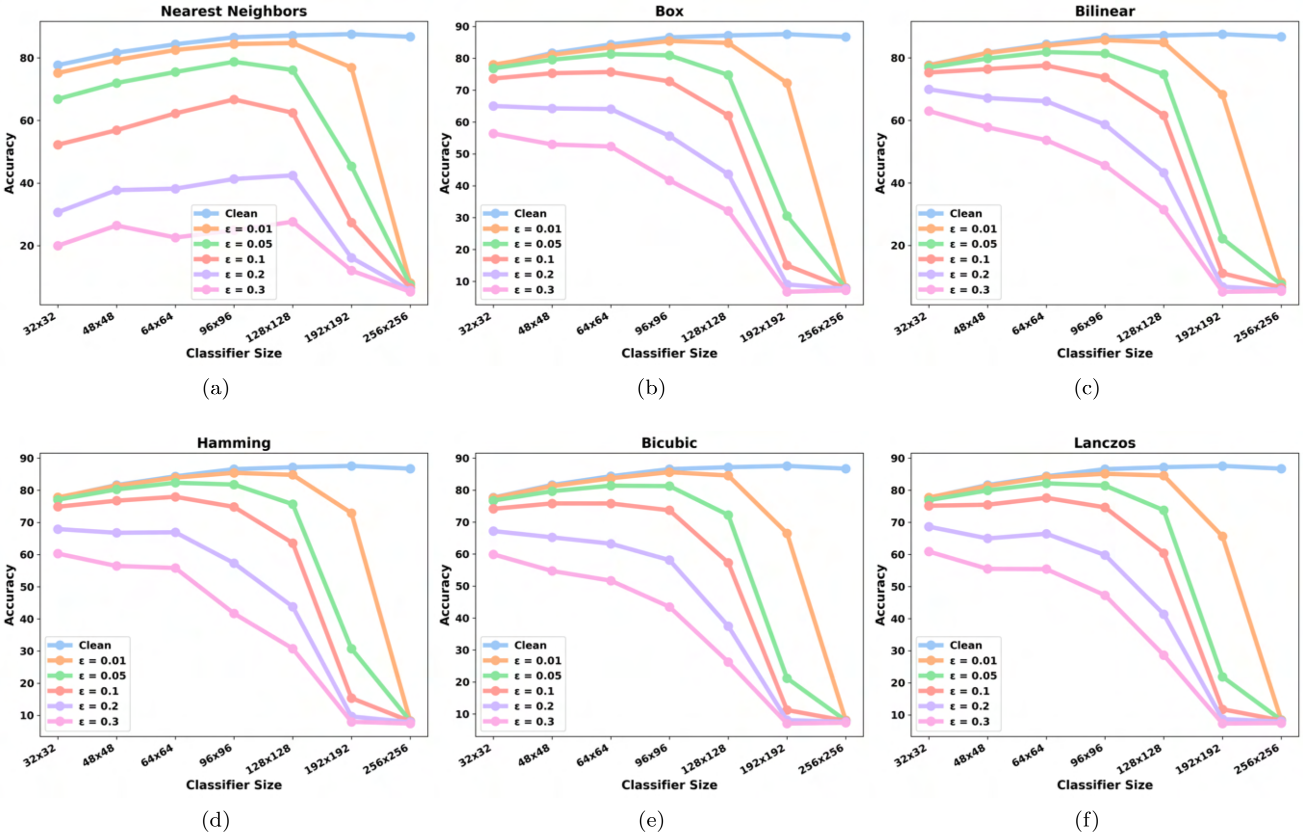
Figure 6: Accuracy of each classifier tested on clean images that were resized. Rows represent the size of the original images we started with while columns represent the size we resized the images to and the size of the classifier used. Sizes along and to the left of the diagonal (downscaled) mostly keep their accuracy after resizing, while sizes to the right of the diagonal (upscaled) start to lose accuracy immediately.

A. Clean Image Resizing

To help illustrate why we focus on downscaling images instead of upscaling them, we show the accuracy of upscaled images in Fig. 6. When you take small images and upscale them, you lose the quality and definition of them the further you go. And since classifiers are often trained on images that were downscaled or not resized at all, they start to become more inaccurate. For example, if we take our 32×32 images and scale them up to 256×256, we are only able to correctly classify 22.03%.

B. Resizing Kills Adversariality

We show in Fig. 7, all filters (except for nearest neighbor) achieve similar accuracies on clean and attacked images, regardless of ϵ or size. Most examples fool the network they were created on, regardless of the ϵ used. As the resized images get further in size from the original, they tend to stop working (*e.g.* 256×256 adversarial images work better on the 128×128 classifier than they do on the 64×64 one). It is clear that when the scale change is extreme, the accuracy for the clean images and adversarial ones is comparable.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 7: Classifier accuracy on clean and adversarial 256^2 images that have been downscaled using six image resizing methods. All adversarial images were created using PGD with varying ϵ values (as seen in each plot's legend) and tested on classifiers that were trained on clean 256^2 images resized using the filter in each plot's title.